

# Desenvolvimento de um Sistema Supervisório Web para Automação Residencial

João Victor M. A. Marques  
Universidade Federal de Viçosa  
Departamento de Engenharia Elétrica  
Viçosa, Brasil  
joao.marques@ufv.br

André Gomes Torres  
Universidade Federal de Viçosa  
Departamento de Engenharia Elétrica  
Viçosa, Brasil  
angoto@ufv.br

**Resumo**—O trabalho apresenta um protótipo de sistema de automação residencial acessível por meio da *Internet*. Para a validação do sistema, dotou-se uma maquete de residência com sensores e atuadores, conectados a um microcontrolador. Desenvolveu-se um programa para leitura e escrita de dados através da conexão do microcontrolador com um *Broker MQTT*. O sistema supervisório da maquete foi desenvolvido em um *dashboard* hospedado na *web*, que pode ser acessado através de qualquer dispositivo com acesso à *Internet*. Avaliou-se o desempenho da aplicação e a usabilidade da interface desenvolvida. Por fim, analisou-se a possibilidade de utilização do sistema em um ambiente real e sugeriu-se melhorias para trabalhos posteriores.

**Palavras-chave**—Automação residencial; ESP32; Protocolo MQTT; *Internet das Coisas*; Node-RED; Experiência do usuário; Interface do usuário

## I. INTRODUÇÃO

Uma casa inteligente é uma residência equipada com sensores que observam o ambiente e atuadores que fornecem serviços proativos, visando melhorar a experiência dos moradores e visitantes [6]. A aplicação desse conceito é possível devido ao desenvolvimento de sistemas de automação residencial, em que um controlador monitora e controla dispositivos interligados [5]. O avanço das tecnologias sem fio permite o gerenciamento desses sistemas por diversos aparelhos eletrônicos, como *laptops*, *smartphones* e *tablets*, possibilitando o acesso remoto por meio da *Internet*, seguindo o conceito da *Internet das Coisas*.

Os sistemas de automação residencial, além de propiciar comodidade para todos os ocupantes, desempenham um papel especial em residências habitadas por idosos. Nessa faixa etária, é comum que as pessoas enfrentem dificuldades para realizar tarefas simples, como ligar ou desligar lâmpadas, fechar portas e janelas, e perceber alterações no ambiente, como um vazamento de gás ou um princípio de incêndio [19]. Dessa forma, uma residência inteligente promove a segurança de seus moradores, reduzindo os riscos de acidentes domésticos.

No entanto, muitas das soluções desenvolvidas baseiam-se em aplicativos móveis e em tecnologias sem fio de curto alcance, como o *Bluetooth* [2], [15], [23]. Isso resulta em alguns desafios: as atualizações de *software* podem não ser suportadas por dispositivos mais antigos, e a comunicação limitada por pequenas distâncias dificulta o acesso remoto para a supervisão da residência. Assim, este trabalho propõe o desenvolvimento

de um sistema supervisório *web* para automação residencial, que pode ser acessado por qualquer dispositivo conectado à *Internet*.

Na seção II será definida uma visão geral do sistema e nas seções III e IV, serão descritas as implementações de *hardware* e de *software*, respectivamente. Os resultados e discussões serão apresentados na seção V, seguido pelas conclusões na seção VI.

## II. VISÃO GERAL

O protótipo desenvolvido é composto por duas partes principais: uma física e outra de *software*. A parte física consiste em uma maquete de uma residência equipada com sensores e atuadores instalados em pontos estratégicos, todos conectados a um microcontrolador. Já a parte de *software* inclui um *script* responsável pela leitura dos dados dos sensores e pelo envio de comandos aos atuadores via microcontrolador. Além disso, o *software* também contempla um painel de controle que funciona como o supervisório da maquete.

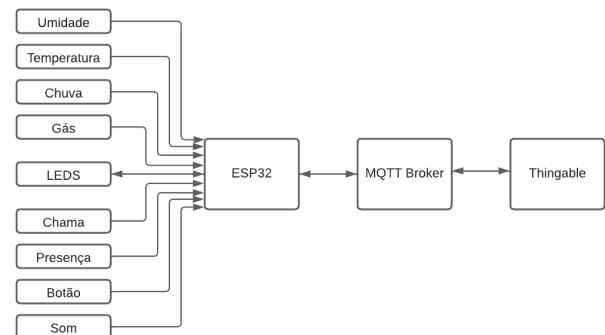


Fig. 1. Diagrama de blocos funcional do sistema contendo as partes de *hardware* e *software*.

A Figura 1 apresenta o diagrama de blocos funcional do sistema. Os sensores de temperatura, umidade e som estão conectados às entradas analógicas do microcontrolador, que possui conversores analógico-digitais integrados para a leitura precisa dessas grandezas. Os sensores de chuva, gás, chama, presença, por sua vez, estão conectados às entradas digitais, sendo acionados apenas quando a grandeza medida ultrapassa

o limite ajustado por meio de potenciômetros integrados aos módulos. O sensor de pressionamento de botão também está conectado à uma entrada digital. O sistema de iluminação da maquete, composto por LEDs, está conectado às saídas digitais do microcontrolador.

O microcontrolador utilizado no sistema está conectado a uma rede *Wi-Fi* com acesso à *Internet*. A comunicação entre a maquete e o supervisor é realizada por meio do protocolo *MQTT* (*Message Queuing Telemetry Transport*). Este protocolo, amplamente utilizado em projetos de *Internet das Coisas*, opera segundo o modelo *Publisher/Subscriber*. Nesse modelo, as mensagens são enviadas e recebidas através de tópicos, com o auxílio de um intermediário, o *Broker*. Diferentes tópicos podem ser utilizados simultaneamente, sem interferência entre as mensagens. Um cliente pode atuar tanto como *publisher*, publicando informações em um tópico, quanto como *subscriber*, recebendo informações de um tópico ao qual está inscrito. Esse protocolo tem sido amplamente difundido em projetos de *Internet das Coisas* devido ao seu baixo consumo energético e facilidade de implementação [17].

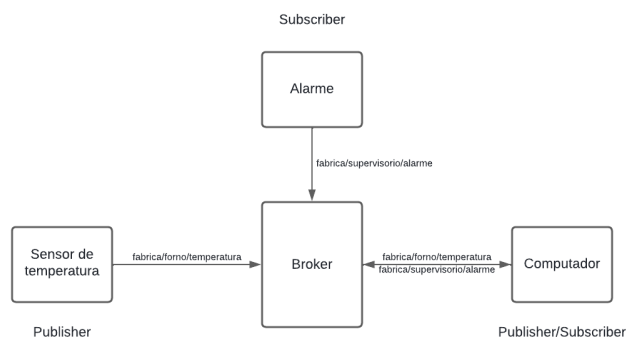


Fig. 2. Comunicação por meio do protocolo *MQTT* em um processo industrial

A Figura 2 exemplifica o uso do protocolo *MQTT* em um ambiente industrial. Suponha uma fábrica onde um sensor de temperatura, instalado em um forno, lê os valores de temperatura e os publica no tópico "fabrica/forno/temperatura". Um alarme, instalado próximo ao forno, é acionado caso a temperatura ultrapasse um valor limite. Nesse cenário, o sensor de temperatura funciona como um *publisher*, enquanto o alarme atua como um *subscriber*, recebendo informações do tópico "fabrica/supervisorio/alarme" após o processamento dos dados pelo computador, que nesse caso, está atuando tanto como *publisher* como *subscriber*.

Para a apresentação dos dados lidos pelos sensores e exibição dos comandos que podem ser realizados pelos atuadores, foi utilizada a plataforma da *Thingable*. A plataforma permite o desenvolvimento e entrega de aplicações de *Internet das Coisas* com facilidade. A *Thingable* permite a criação de aplicativos que monitoram e gerenciam dispositivos de *IoT* dispersamente localizados. As aplicações desenvolvidas podem ser executadas em qualquer navegador de *Internet* [28]. Assim, utilizou-se dessa ferramenta para criar um *dashboard* onde são exibidos

os dados em tempo real de todos os sensores instalados na maquete e foram centralizados todos os possíveis comandos que podem ser enviados aos atuadores.

A implementação do *dashboard* seguiu boas práticas relativas à experiência e à interface do usuário, visando a aceitação do usuário final [20]. Atentou-se para a responsividade do *design*, para que nos diferentes tamanhos de tela o uso do aplicativo se mantivesse consistente. A utilização de ícones reconhecíveis com *feedback* visual claro facilitou a identificação dos estados alterados na maquete. Ademais, optou-se por utilizar controles simples e diretos, como botões e controles deslizantes.

### III. IMPLEMENTAÇÃO DO HARDWARE

Esta seção abordará sobre os dispositivos de *hardware* utilizados na construção do sistema de automação que serão instalados na maquete para validar o trabalho desenvolvido.

#### A. ESP-WROOM-32

O ESP-WROOM-32, apresentado na Figura 3, é um sistema em um chip (SoC) que possui um microcontrolador integrado dual-core de 32 bits com *Wi-Fi* e *Bluetooth* incorporados. São disponíveis 25 portas programáveis que contêm conversores ADC e DAC, 16 canais PWM e 512 kB de memória SRAM [8].

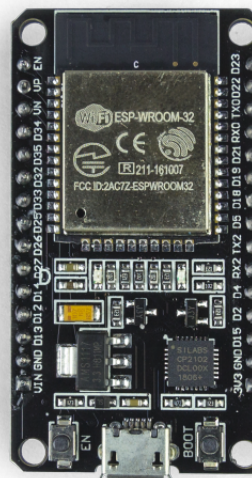


Fig. 3. Microcontrolador de modelo ESP-WROOM-32. Fonte: [21].

Foi escolhida uma placa da família da ESP-32 devido a seu baixo custo, facilidade de conexão *Wi-Fi* sem a necessidade de módulos adicionais, menor tensão de alimentação comparado com outras placas de desenvolvimento e baixo consumo de energia.

#### B. Sensor de temperatura e umidade DHT11

O DHT11, inserido ao módulo da Figura 4, é um sensor que possui um termistor e um sensor capacitivo integrados,

utilizados para medir a temperatura e a umidade do ambiente, respectivamente. Dessa forma, em um só módulo é possível obter a medida dessas duas grandezas físicas do ambiente.

Esse sensor pode realizar medidas de temperatura de 0 a 50°C com uma precisão de 2°C, e umidade entre 20% a 90%, com uma precisão de 5% [14].



Fig. 4. Módulo com sensor de temperatura e umidade DHT 11 integrado. Fonte: [27].

### C. Sensor de gases inflamáveis MQ-9

O sensor MQ-9, exibido em destaque na Figura 5, possui a capacidade de detectar concentrações de gases inflamáveis, como metano e propano, e monóxido de carbono. Os módulos da família MQ funcionam por meio de eletrodos que reagem com a presença do gás, diminuindo sua resistência a medida que a concentração do gás que se deseja detectar aumenta. Esse módulo pode detectar concentrações na faixa de 100 a 10000ppm desses gases [11].



Fig. 5. Módulo com sensor de gases inflamáveis MQ9 integrado. Fonte: [24].

### D. Sensor de chuva FC-37

O módulo sensor de chuva FC-37 é composto por duas partes: a placa coletora que coleta as gotas de água, mostrada na parte superior da Figura 6, e a placa eletrônica que envia o sinal para o microcontrolador, na parte inferior. A placa coletora possui uma resistência que varia de acordo com a presença da água em sua superfície: caso esteja molhada a

resistência aumenta, e sua tensão de saída diminui e caso esteja seca a resistência diminui, aumentando sua tensão de saída [29].

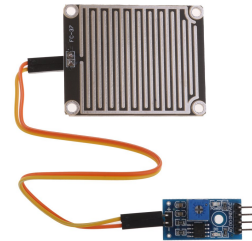


Fig. 6. Placa coletora de gotículas de água e placa eletrônica que compõem o sensor de chuva FC-37. Fonte: [10].

### E. Sensor de chama HW-491

O sensor de chama HW-491, apresentado na Figura 7, detecta a presença de chama ou outras fontes de calor com um sensor infravermelho que detecta luz com comprimento de onda entre 760 e 1100nm. A distância para detecção é de 20cm até 100cm [9].

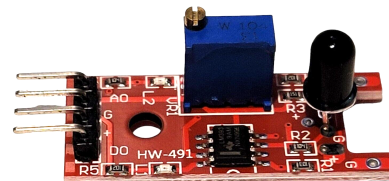


Fig. 7. Módulo contendo sensor infravermelho para detecção de chama. Fonte: [7].

### F. Sensor de presença HC-SR501

O sensor de presença HC-SR501 visto na Figura 8, é um sensor piroelétrico, ou seja, detecta movimento de acordo com a variação de radiação infravermelha que o corpo humano emite. A distância máxima de leitura pode ser variada de 3 a 7 metros e possui um ângulo de abertura de 110°. Outro possível ajuste é o tempo em que a saída do sensor ficará em tempo alto ao se detectar um movimento, que pode ser ajustado de 5 segundos a 2,5 minutos [1].

### G. Botão HW-483

O módulo HW-483 exibido na Figura 9, contém uma chave tátil que está configurada em uma topologia *pull-up*, ou seja, sua saída fica em nível alto e, ao ter o botão pressionado mecanicamente, vai para o nível baixo [12].



Fig. 8. Módulo com o sensor piroelétrico para identificação de presença. Fonte: [22].

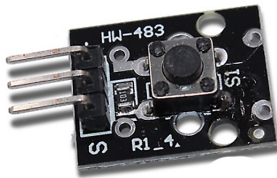


Fig. 9. Módulo contendo a chave táctil para pressionamento mecânico. Fonte: [26].

#### H. LED RGB HW-478

O módulo HW-478 apresentado na Figura 10, pode ser utilizado para produzir diversas cores, compostas das três cores primárias: verde, vermelho e azul. Isso é possível devido ao fato da placa possuir três diodos: vermelho, verde e azul e, ao variar a intensidade da luz destes três diodos, pode-se criar uma grande variação de cores [13].

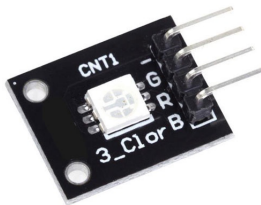


Fig. 10. Módulo com LED RGB. Fonte: [25].

#### I. Detector de som HW-496

O módulo HW-496, exibido na Figura 11, mede a intensidade do som em um ambiente a partir de um microfone de condensador elétrico. O módulo apresenta uma saída digital que, caso não detecte som, se mantém em nível baixo e quando o sensor detecta algum som dentro do limiar definido pelo ajuste por meio do potenciômetro localizado no módulo, a

saída altera-se para o nível alto. Há ainda uma saída analógica em que pode-se obter a variação do som no ambiente [18].

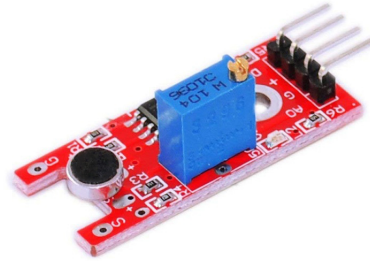


Fig. 11. Módulo detector de som. Fonte: [4].

## IV. IMPLEMENTAÇÃO DO SOFTWARE

### A. Programação da ESP32

Para o desenvolvimento do programa que será carregado no microcontrolador conectado aos dispositivos instalados na maquete, optou-se por utilizar o ambiente de desenvolvimento do Arduino. O Arduino IDE é uma plataforma *open-source* que facilita a escrita de código e o carregamento de programas na placa de desenvolvimento [3]. Apesar de ser desenvolvido para as placas da família do Arduino, com a instalação de um pacote adicional, é possível carregar programas em placas da família da ESP32.

O programa desenvolvido é estruturado em duas funções principais: *setup* e *loop*. Na *setup*, são realizadas as configurações iniciais necessárias para a execução do programa. Os pinos de controle são definidos como entradas ou saídas usando o comando *pinMode*. Em seguida, é estabelecida a conexão com a rede *Wi-Fi*. Caso a conexão seja bem-sucedida, o microcontrolador se conecta ao servidor *MQTT*. As credenciais de acesso à rede *Wi-Fi* e ao servidor foram definidas em um arquivo de configurações.

A função *loop* é executada em repetição contínua, como é ilustrado no fluxograma da Figura 12. Para cada iteração, verifica-se se o cliente está conectado ao servidor *MQTT*. Caso não esteja, há uma função auxiliar responsável por reconectar-se ao servidor e exibir o erro recebido em caso de falha na conexão. Em caso de conexão, são lidos os dados de todos os dispositivos conectados ao microcontrolador. Antes da publicação dos dados, é feita uma verificação do intervalo de tempo desde a última publicação, para que sejam publicados a cada 5 segundos. Para incluir a possibilidade do programa receber dados enquanto é executado, é definida a função *callback*. Essa função é responsável por processar as mensagens recebidas do servidor *MQTT*, identificando qual propriedade foi alterada pelo usuário e atualizando os valores correspondentes no microcontrolador para as iterações seguintes.

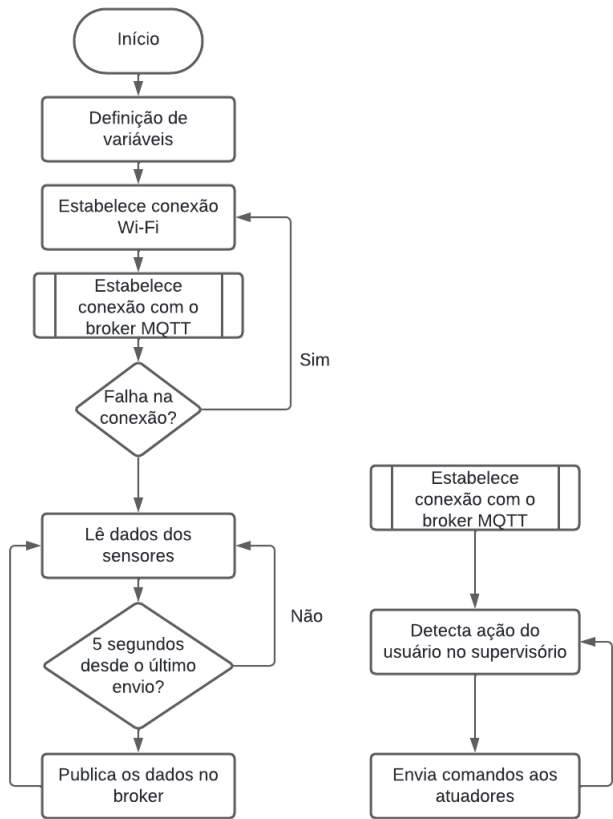


Fig. 12. Fluxograma do algoritmo desenvolvido para a leitura e escrita dos dados no sistema.

### B. Configuração da Thingable

A configuração de uma aplicação de *Internet das Coisas* na *Thingable* envolve a criação de duas entidades: "Tipos" e "Dispositivos". Os "Tipos" são agrupamentos de "Dispositivos" que possuem características em comum. Já os "Dispositivos" representam qualquer objeto que se deseja controlar e monitorar. No caso deste trabalho, o "Dispositivo" representa a maquete da residência com todos seus sensores e atuadores instalados.

Para a criação do "Tipo" é necessário definir as especificações dos dados que serão trafegados no sistema. Devem ser definidos se os dados são numéricos, booleanos ou um conjunto de caracteres. Caso sejam numéricos, é necessário definir se são números inteiros ou reais. Caso sejam números reais, é preciso informar o número de casas decimais utilizadas. A definição dos "Tipos" tem a função principal de customizar a base de dados para que as informações possam ser corretamente armazenadas.

A criação do "Dispositivo" envolve o preenchimento de informações básicas como a definição de um número de série, nome para identificação e a descrição do "Dispositivo". Além disso, pode-se definir a localização de onde está situado e configurar permissões de acesso. Após a correta criação do "Dispositivo", são fornecidas as credenciais que permitem o acesso ao servidor *MQTT* para o qual os dados serão enviados.

### C. Criação do dashboard

Uma das ferramentas disponíveis na *Thingable* é o *Logic Builder*, que incorpora o *Node-RED*, uma plataforma de programação visual que facilita o desenvolvimento de fluxos de informação de maneira simplificada. O *Node-RED* disponibiliza um editor onde podem ser inseridos blocos de código pré-compilados e prontos para serem utilizados, chamados de nós [16].

Os nós, de forma geral, podem ser classificados em nós de entrada, processamento e saída, que unidos formam um fluxo. Em suma, os nós de entrada e de saída utilizados nesse trabalho são, respectivamente o *mqtt in* e o *mqtt out*. O *mqtt in* assina as informações enquanto o *mqtt out* publica informações em um determinado tópico do servidor *MQTT* do dispositivo. Os demais nós de processamento são utilizados para padronizar as informações recebidas ou enviadas de acordo com as estruturas definidas. Na Figura 13, é ilustrado o caso de um nó *mqtt in* que recebe informação de um determinado tópico, um nó *function* que realiza o processamento da informação e a publica em outro tópico por meio do nó *mqtt out*.

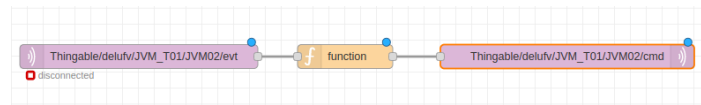


Fig. 13. Exemplificação de um fluxo no *Node-RED* com os nós *mqtt in*, *mqtt out* e *function*.

Adicionalmente, o *Node-RED* possui um módulo que oferece acesso a nós específicos para a construção de dashboards. Esses nós permitem a criação de elementos gráficos que exibem dados em tempo real ou apresentam séries históricas por meio de gráficos. Além disso, eles permitem a interação direta com o usuário através de botões, caixas de diálogo e outras interfaces de entrada. É possível também enviar notificações e ajustar dinamicamente a aparência dos componentes do *dashboard*, adaptando-o conforme a necessidade.

O processo de criação e personalização do *dashboard* foi guiado pela ênfase na experiência do usuário. Foram cuidadosamente escolhidos as cores, as fontes de texto, os modelos de gráficos e os elementos de entrada de dados, visando tornar a interface intuitiva e acessível. A disposição dos elementos na tela foi planejada para facilitar a compreensão dos estados da maquete e das ações disponíveis ao usuário, garantindo uma interação eficiente e agradável com o sistema supervisório.

## V. RESULTADOS E DISCUSSÕES

Todos os sensores e atuadores foram estrategicamente posicionados na maquete para cumprir suas respectivas funções. Os sensores de temperatura, umidade, chuva e presença foram colocados nas áreas externas da residência, enquanto os sensores de gases inflamáveis e de chama foram colocados na cozinha, onde o risco de incêndio e vazamento de gás é mais elevado. O sensor de pressionamento de botão foi localizado na porta de entrada para simular uma campainha, e o sensor de

som foi instalado em um quarto para detectar ruídos noturnos. Em cada cômodo, um LED foi inserido para compor o sistema de iluminação da casa.

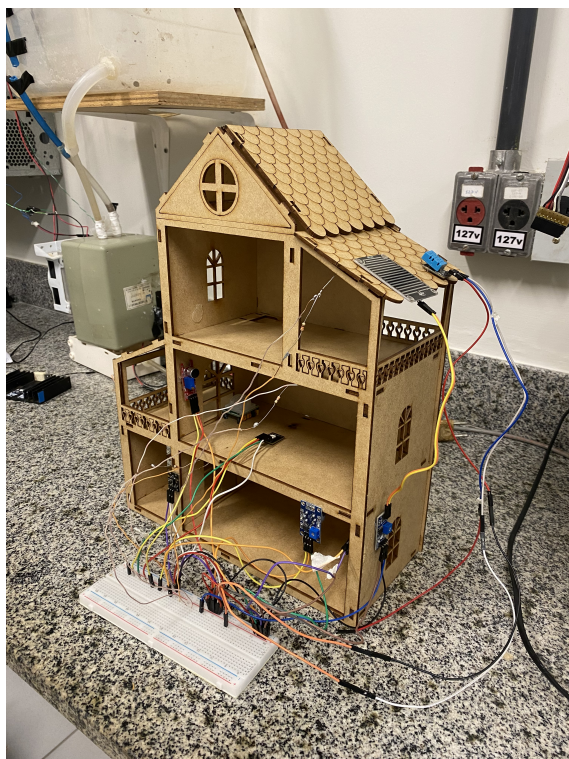


Fig. 14. Maquete da residência dotada de todos os sensores e atuadores utilizados.

Os componentes foram conectados ao microcontrolador, que inicialmente foi ligado a um computador que continha o programa de leitura e escrita de dados. Após o carregamento do programa na ESP32, a execução inicial confirmou o sucesso das conexões com a rede *Wi-Fi* local e com o servidor *MQTT*. Dessa forma, os dados captados já estavam sendo enviados para o dispositivo configurado no sistema *Thingable*.

Com a integração bem-sucedida entre o microcontrolador e o *Broker MQTT*, o funcionamento do *dashboard* pôde ser visualizado. A versão final do *dashboard* é composta por uma seção que exibe os dados de temperatura e umidade, mostrando as medições em tempo real e em uma série temporal; e uma seção que apresenta a leitura dos demais sensores. Nesta, os estados dos sensores digitais foram representados pela mudança na cor dos ícones que indicam as grandezas medidas. O sensor analógico de som foi ilustrado por meio de um controle deslizante. Por fim, a seção de controle da iluminação permitiu que o usuário interagisse com os LEDs instalados na maquete, podendo ligá-los ou desligá-los com um clique no ícone correspondente. Em um dos LEDs, foi implementado um controle de intensidade de luminosidade, ajustável através de um slider no *dashboard*, e em outro, era possível alterar a cor da luz utilizando uma paleta de cores.

Para melhorar a experiência do usuário, foram adicionadas funcionalidades extras na interface do *dashboard*. Cada botão ou ícone agora exibe uma legenda (*tooltip*) quando o usuário

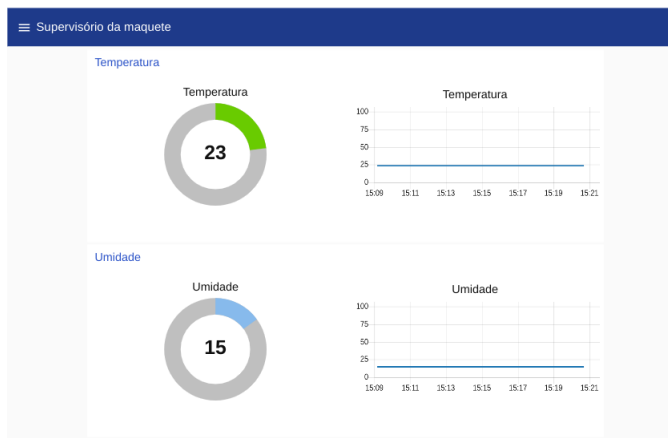


Fig. 15. Seções de apresentação dos dados de temperatura e umidade no *dashboard*.

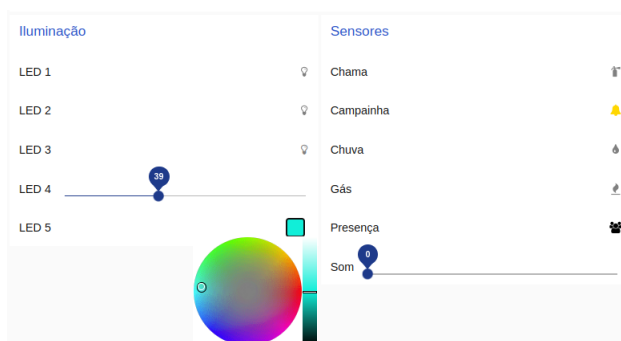


Fig. 16. Seções de comando e de leitura dos sensores no *dashboard*.

passa o cursor sobre ele. As legendas dos botões explicam o que acontecerá ao clicá-los, enquanto as dos ícones indicam o evento monitorado pelo sensor. Essas legendas ajudam novos usuários a entender melhor as funções disponíveis e reduzem a chance de erros ao usar o sistema. Além disso, qualquer mudança detectada pelos sensores gera uma notificação na tela enquanto o usuário estiver conectado. Caso o sistema não esteja sendo acessado, alertas críticos, como a detecção de fogo ou vazamento de gás, são enviados por *e-mail*.

Por fim, avaliou-se a responsividade das telas do aplicativo, demonstrando que ele se manteve funcional em diferentes tamanhos de tela, tanto em *smartphones* quanto em *laptops*, preservando a identidade visual e as funcionalidades adicionais desenvolvidas. O desempenho do sistema também foi analisado, revelando que o tempo de resposta das ações só era afetado em casos de conexões instáveis ou de baixa velocidade com a *Internet*, não gerando latência perceptível para o usuário em condições normais. Quanto aos sensores, após o ajuste de sensibilidade para o ambiente reduzido da maquete, as leituras se mostraram coerentes com as condições observadas no ambiente.

## VI. CONCLUSÃO

O desenvolvimento do protótipo confirmou a viabilidade de utilizar um único microcontrolador para automatizar uma

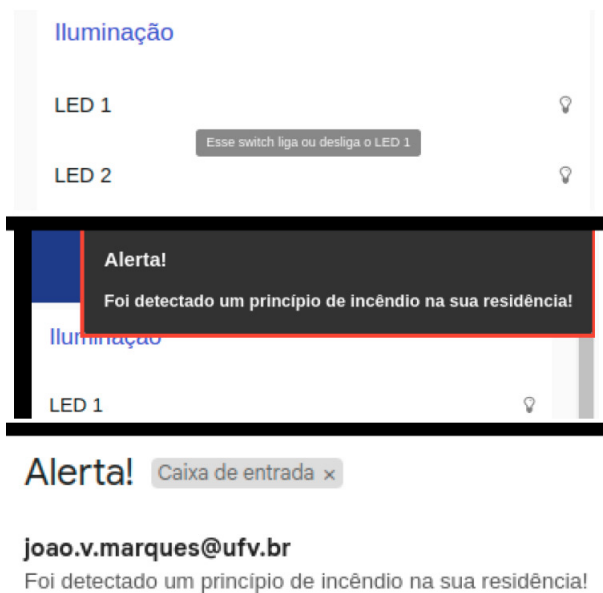


Fig. 17. Exibição das funcionalidades extras desenvolvidas: *tooltips*, notificações e *e-mail* disparado para o usuário do sistema, respectivamente.

residência, resultando em um menor custo de implementação e baixo consumo de energia. A adoção do protocolo *MQTT* garantiu excelente desempenho na transmissão de dados, proporcionando segurança e confiabilidade. Além disso, a ferramenta *Node-RED*, incorporada pela plataforma *Thingable*, facilitou a criação de uma interface gráfica acessível e fácil de usar, com amplas opções de personalização e integração com serviços externos.

Para implementar o sistema em uma aplicação real, é importante considerar a atenuação dos sinais quando sensores ou atuadores estão distantes do microcontrolador. Nessas situações, o uso de amplificadores ou repetidores pode ser necessário para reforçar o sinal. Além disso, a inclusão de baterias ou *nobreaks* é essencial para assegurar a continuidade da operação em caso de interrupções no fornecimento de energia.

A inteligência artificial tem um grande potencial para aprimorar futuros sistemas de automação residencial. Em especial, no cuidado com idosos, pode-se implementar uma assistência proativa, onde o sistema antecipa as necessidades e ajusta o ambiente conforme os padrões de uso, como acender as luzes no horário habitual e regular a temperatura por meio de aquecedores ou ar-condicionado. Além disso, a integração com assistentes virtuais que permitem o controle por voz pode ser muito útil, proporcionando uma maneira prática de controlar dispositivos quando estão fora de alcance e facilitando a comunicação em situações de emergência.

#### REFERÊNCIAS

- [1] HC-SR501 PIR MOTION DETECTOR.
- [2] AMORAN, A. E., OLUWOLE, A. S., FAGOROLA, E. O., AND DIARAH, R. Home automated system using bluetooth and an android application. *Scientific African 11* (2021), e00711.
- [3] ARDUINO LLC. *Arduino IDE Documentation*, 2024. Disponível em: <https://docs.arduino.cc/software/ide-v1/tutorials/>. Acessado em: 11-ago-2024.
- [4] ARYABOT. Small microphone sound sensor module hw-496. Disponível em: <https://aryabot.in/small-microphone-sound-sensor-module-hw-496/>. Acessado em: 12-ago-2024.
- [5] ASADULLAH, M., AND RAZA, A. An overview of home automation systems. In *2016 2nd international conference on robotics and artificial intelligence (ICRAI)* (2016), IEEE, pp. 27–31.
- [6] DING, D., COOPER, R. A., PASQUINA, P. F., AND FICI-PASQUINA, L. Sensor technology for smart homes. *Maturitas 69*, 2 (2011), 131–136.
- [7] EBAY. Sensor de chama ky-026 módulo sensor receptor infravermelho para arduino. Disponível em: <https://www.ebay.com/itm/355063574536>. Acessado em: 12-ago-2024.
- [8] ESPRESSIF SYSTEMS. *ESP32-WROOM-32 Datasheet*, 2023. Versão 3.4.
- [9] FUTURE ELECTRONICS LTD. *Flame Sensor Module*.
- [10] FÁCIL, C. Sensor de chuva. Disponível em: <https://www.circuitofacil.com.br/produto/sensor-de-chuva/>. Acessado em: 12-ago-2024.
- [11] HENAN HANWEI ELECTRONICS CO., LTD. *MQ-9 Semiconductor Sensor for CO/Combustible Gas*.
- [12] HOKORIKU SENSOR INOVATION. *Tactile Switches*.
- [13] JOY-IT. *KY-016 RGB 5mm LED module*.
- [14] MOUSER ELECTRONICS. *DHT 11 Humidity Temperature Sensor*.
- [15] NGEREM, E., MISRA, S., OLURANTI, J., CASTILLO-BELTRAN, H., AHUJA, R., AND DAMASEVICIUS, R. A home automation system based on bluetooth technology using an android smartphone. In *Evolving Technologies for Computing, Communication and Smart World: Proceedings of ETCCS 2020* (2021), Springer, pp. 527–536.
- [16] OPENJS FOUNDATION. *Node-RED documentation*, 2024.
- [17] QUINCOZES, S., EMILIO, T., AND KAZIENKO, J. Mqtt protocol: fundamentals, tools and future directions. *IEEE Latin America Transactions 17*, 09 (2019), 1439–1448.
- [18] RAJGURU ELECTRONICS. *Sound Detection Sensor*.
- [19] RANSING, R. S., AND RAJPUT, M. Smart home for elderly care, based on wireless sensor network. In *2015 International Conference on Nascent Technologies in the Engineering Field (ICNTE)* (2015), IEEE, pp. 1–5.
- [20] ROBBINS, J. N. *Learning web design: A beginner's guide to HTML, CSS, JavaScript, and web graphics*. "O'Reilly Media, Inc.", 2012.
- [21] ROBOCORE. Esp32 - wifi + bluetooth. Disponível em: <https://www.roboconet.net/wifi/esp32-wifi-bluetooth>. Acessado em: 12-ago-2024.
- [22] ROBOCORE. Sensor de presença pir - hc-sr501. Disponível em: <https://www.roboconet.net/sensor-ambiente/sensor-de-presenca-pir-hc-sr501>. Acessado em: 12-ago-2024.
- [23] SHAWON, M. S. H., DAS, C., AHAMMED, M. T., BISWAS, G., MIA, M. S., EVA, E. A., AND SAKIB, M. N. Voice controlled smart home automation system using bluetooth technology. In *2021 4th International Conference on Recent Trends in Computer Science and Technology (ICRTCTST)* (2022), IEEE, pp. 67–72.
- [24] SHOP, F. Mq-9 - sensor for carbon monoxide and combustible gases. Disponível em: <https://funduino.com/en/electronic-modules/sensors/gases/mq-9-sensor-for-carbon-monoxide-and-combustible-gases>. Acessado em: 12-ago-2024.
- [25] SHOPEE. Módulo led smd rbg hw-478. Disponível em: <https://shopee.com.br/Modulo-Led-Smd-Rbg-Hw-478-Para-Arduino-Entre-Outros-Sistemas-Embarcados-i.513644768.10266760583>. Acessado em: 12-ago-2024.
- [26] SHOPEE. Módulo push button hw-483. Disponível em: <https://shopee.com.br/Modulo-Push-Button-Hw-483-Para-Arduino-Entre-Outros-Sistemas-Embarcados-i.513644768.10066755798>. Acessado em: 12-ago-2024.
- [27] STORE, M. R. Módulo eletrônico integrado com um sensor de temperatura e umidade dht-11. Disponível em: <https://www.myrobotstore.com.br/grupo-padrao/subgrupo-padrao/modulo-eletronico-integrado-com-um-sensor-de-temperatura-e-umidade-dht-11>. Acessado em: 12-ago-2024.
- [28] THINGABLE! Bem vindo a wiki da thingable!, 2023. Disponível em: <https://docs.thingable.com/bem-vindo-a-wiki-da-thingable>. Acessado em: 11-ago-2024.
- [29] VAISALA. *YI-83 Rain Detector*.

**JOÃO VICTOR MOREIRA ALVES MARQUES**

**DESENVOLVIMENTO DE UM SISTEMA SUPERVISÓRIO WEB PARA  
AUTOMAÇÃO RESIDENCIAL**

Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 402 – Projeto de Engenharia II e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

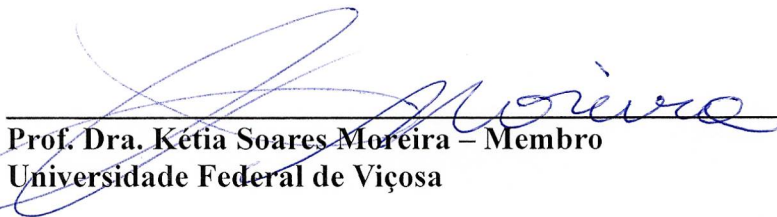
Aprovada em 29 de agosto de 2024.

**COMISSÃO EXAMINADORA**



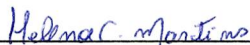
---

**Prof. Dr. André Gomes Torres – Orientador**  
Universidade Federal de Viçosa



---

**Prof. Dra. Kétia Soares Moreira – Membro**  
Universidade Federal de Viçosa



---

**Msc. Helena Cristo Martins – Membro**  
Universidade Federal de Viçosa