

Arquitetura de Automação Residencial Baseada em Internet das Coisas com NodeMCU, Firebase e Aplicativo Móvel

Eduardo Eugenio Rodrigues Sartor¹, Alessandro Paulo de Oliveira², Felipe Walter Dafico Pfrimer²

¹Universidade Tecnológica Federal do Paraná, Toledo, Brasil (edu-sartor@hotmail.com)

²Universidade Tecnológica Federal do Paraná, Toledo, Brasil

Resumo: Este trabalho apresenta uma arquitetura de automação residencial baseada em Internet das Coisas, integrando NodeMCU ESP8266, Firebase Realtime Database e aplicativo móvel desenvolvido em Flutter. A solução reúne, em uma única plataforma, o monitoramento de temperatura e umidade e o controle da iluminação, da persiana, do portão eletrônico e do sistema de alarme. Os resultados validaram a arquitetura proposta, evidenciando sua viabilidade para aplicações residenciais de baixo custo.

Palavras-chave: Internet das Coisas; Automação residencial; NodeMCU; Firebase.

INTRODUÇÃO

A automação residencial consiste na integração de sensores, atuadores e sistemas computacionais capazes de monitorar e controlar automaticamente diferentes funções de uma residência. O avanço da Internet das Coisas (*Internet of Things* – IoT) possibilitou a comunicação entre esses dispositivos por meio da Internet, permitindo o monitoramento remoto e a automação de tarefas relacionadas à iluminação, climatização, segurança, monitoramento ambiental e acionamento de dispositivos elétricos. Como consequência, a automação residencial vem se consolidando como uma das principais aplicações da IoT, proporcionando maior conforto, praticidade e conectividade aos usuários (Stolojescu-Crişan et al., 2021; Addow et al., 2025).

O crescimento dessas aplicações está diretamente relacionado à evolução das tecnologias de comunicação sem fio, à redução do custo dos dispositivos eletrônicos e à ampla disponibilidade de plataformas de *hardware* e *software* de código aberto. Nesse cenário, a plataforma NodeMCU tornou-se uma alternativa amplamente utilizada no desenvolvimento de aplicações IoT devido ao seu baixo custo, conectividade Wi-Fi integrada e facilidade de programação. Associada aos serviços de computação em nuvem, essa plataforma possibilita o desenvolvimento de sistemas distribuídos para monitoramento e controle remoto em aplicações residenciais, industriais e acadêmicas (Ayeni e Adesoba, 2025).

Paralelamente, a computação em nuvem passou a desempenhar papel fundamental nas arquiteturas modernas baseadas em IoT. Plataformas como o *Firebase Realtime Database* (FRD) permitem o

armazenamento e a sincronização de dados em tempo real entre dispositivos embarcados e aplicações móveis, simplificando a comunicação entre sensores, atuadores e usuários. Essa abordagem elimina a necessidade de servidores dedicados e reduz a complexidade de implementação de sistemas distribuídos (Tripathy e Das, 2022).

Além de proporcionar maior conforto aos usuários, a automação residencial vem sendo empregada para aumentar a segurança dos ambientes e auxiliar no gerenciamento do consumo de energia. O monitoramento contínuo de variáveis como temperatura, umidade, presença e movimento permite que sistemas inteligentes executem ações automáticas de acordo com as condições do ambiente. Nesse contexto, soluções baseadas em IoT vêm sendo amplamente investigadas para o desenvolvimento de edificações mais inteligentes, conectadas e sustentáveis (Poyyamozi et al., 2024; Addow et al., 2025).

Apesar da evolução tecnológica observada nos últimos anos, muitas soluções comerciais ainda utilizam plataformas proprietárias, apresentam elevado custo de aquisição ou oferecem limitada flexibilidade para adaptações e integração de novos dispositivos. Essas características dificultam sua utilização em aplicações educacionais, projetos de pesquisa e residências de baixo orçamento, evidenciando a necessidade de soluções abertas, modulares e baseadas em tecnologias de baixo custo (Stolojescu-Crişan et al., 2021; Ayeni e Adesoba, 2025).

Embora Ayeni e Adesoba (2025) tenham demonstrado a viabilidade de sistemas de automação residencial baseados em NodeMCU e Firebase, o

presente trabalho propõe uma arquitetura integrada que reúne, em uma única plataforma, diferentes estratégias de controle, incluindo iluminação, persiana motorizada, portão eletrônico baseado em máquina de estados, sistema de alarme e monitoramento ambiental, ampliando as funcionalidades da arquitetura proposta pelos autores.

Nesse contexto, este trabalho propõe e implementa uma arquitetura de automação residencial baseada na plataforma NodeMCU, integrada ao FRD e a um aplicativo móvel desenvolvido na plataforma Flutter. A solução reúne, em uma única plataforma, funcionalidades de monitoramento de temperatura, umidade e movimento, além do controle remoto da iluminação, das persianas, do portão eletrônico e do sistema de alarme. A arquitetura integra dispositivos embarcados, banco de dados em nuvem e aplicativo móvel utilizando tecnologias de código aberto e componentes eletrônicos de baixo custo e ampla disponibilidade comercial. O objetivo deste trabalho é implementar e avaliar essa arquitetura em um protótipo funcional de automação residencial baseado em Internet das Coisas.

MATERIAL E MÉTODOS

A Figura 1 apresenta o diagrama de blocos da arquitetura proposta para a implementação do sistema de automação residencial.

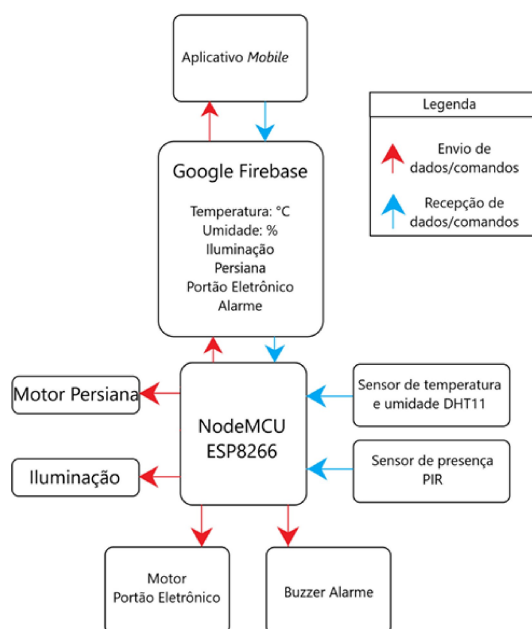


Figura 1. Diagrama de blocos da arquitetura do sistema proposto.

Para possibilitar o monitoramento e o controle das funcionalidades implementadas, foi desenvolvido um aplicativo móvel na plataforma Flutter, responsável pela interface entre o usuário e o sistema de automação residencial. Por meio dessa interface, o

usuário pode visualizar, em tempo real, as informações provenientes dos sensores, como temperatura, umidade e presença, além de enviar comandos para o acionamento da iluminação, das persianas, do portão eletrônico e do sistema de alarme.

Inicialmente, foi considerada a utilização de uma interface Web hospedada diretamente no servidor embarcado da plataforma NodeMCU. Entretanto, essa abordagem restringia o acesso ao sistema aos usuários conectados à mesma rede Wi-Fi do dispositivo, limitando o monitoramento e o controle remoto das funcionalidades implementadas. Em razão dessa limitação, optou-se pelo desenvolvimento de um aplicativo móvel integrado ao FRD.

Na arquitetura proposta, o aplicativo móvel constitui a interface de comunicação entre o usuário e o sistema, enquanto o FRD é responsável pela sincronização, em tempo real, das informações entre o aplicativo e a plataforma NodeMCU. Dessa forma, os comandos enviados pelo usuário são armazenados no banco de dados e, ao serem detectados pelo NodeMCU em sua rotina de monitoramento contínuo, acionam os dispositivos correspondentes. No sentido inverso, os dados adquiridos pelos sensores são inicialmente processados pela plataforma NodeMCU e, em seguida, enviados ao FRD, onde são armazenados e disponibilizados ao aplicativo móvel para visualização pelo usuário. Essa arquitetura permite que o monitoramento das variáveis ambientais e o controle dos dispositivos sejam realizados a partir de qualquer local com acesso à Internet, proporcionando maior flexibilidade, praticidade e acessibilidade às aplicações de automação residencial.

APLICATIVO MÓVEL

O aplicativo foi desenvolvido no editor de código-fonte Visual Studio Code, utilizando o framework Flutter e a linguagem de programação Dart. O Flutter permite o desenvolvimento de aplicações multiplataforma a partir de uma única base de código, possibilitando a compilação para sistemas operacionais Android e iOS. Neste trabalho, a validação do aplicativo foi realizada em dispositivos Android, sendo a instalação efetuada por meio de um arquivo no formato APK (Android Application Package), gerado pelo ambiente de desenvolvimento Flutter.

Ao ser iniciado, o aplicativo exibe uma tela de autenticação, apresentada na Figura 2, na qual usuários já cadastrados podem realizar o login para acessar as demais funcionalidades. Para novos usuários, está disponível uma opção de cadastro, opção Registre-se, que redireciona o usuário a uma tela de criação de conta, mostrada na Figura 3, na

qual deve ser informado um endereço de e-mail válido e ainda não registrado no sistema. A verificação da validade do e-mail informado é realizada automaticamente pelo serviço de autenticação do Firebase, por meio da biblioteca FirebaseAuth integrada à programação do aplicativo.

Após a autenticação, o usuário é direcionado à tela principal do aplicativo, apresentada na Figura 4, na qual são exibidos, em tempo real, os valores de temperatura (°C) e umidade (%) monitorados pelo sistema. A partir dessa interface, o usuário pode acessar as principais funcionalidades de automação por meio de quatro opções de controle.

A funcionalidade de iluminação, mostrada na Figura 5, permite o acionamento e o desligamento das lâmpadas controladas pelo sistema. Cada ponto de iluminação é representado por um botão interativo, cujo estado é indicado pela cor e pelo texto exibidos no botão. A cor amarela identifica uma lâmpada ligada, enquanto a cor preta indica que a lâmpada está desligada, facilitando a identificação visual do estado de cada dispositivo.



Figura 2. Tela inicial do aplicativo.

A opção Persianas, apresentada na Figura 6, permite o controle da persiana da sala por meio de três comandos: abrir, parar e fechar. Os comandos de abertura e fechamento acionam o motor no sentido correspondente, enquanto o comando de parada interrompe o movimento da persiana na posição desejada pelo usuário.

O controle do portão eletrônico é realizado por meio de um único botão, mostrado na Figura 4. A cada acionamento, o sistema executa a sequência correspondente ao estado atual do portão, permitindo iniciar a abertura, interromper o movimento ou inverter o sentido de deslocamento para realizar o fechamento.

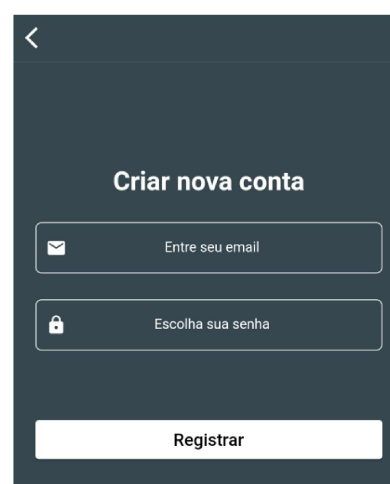


Figura 3. Tela de registro de novas contas.

Por fim, o aplicativo disponibiliza o controle do sistema de alarme, apresentado na Figura 4, por meio de um botão com dois estados: ligado e desligado. Quando ligado, o sistema monitora continuamente o ambiente por meio do sensor PIR e, ao detectar movimento, aciona o buzzer. Quando desligado, o monitoramento é suspenso, impedindo o acionamento do buzzer.



Figura 4. Tela principal do aplicativo.



Figura 5. Tela para controle da iluminação.

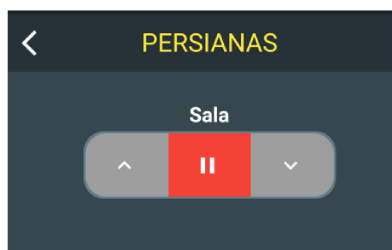


Figura 6. Tela para controle de persianas.

SISTEMA DE MONITORAMENTO DE TEMPERATURA E UMIDADE

O circuito de interfaceamento do sensor DHT11 é apresentado na Figura 7. Segundo o fabricante, o DHT11 possui precisão nominal de $\pm 2^\circ\text{C}$ para temperatura e $\pm 5\%$ para umidade relativa. A comunicação entre o sensor e a plataforma NodeMCU é realizada por meio de uma interface digital de um único fio, conectada a um pino GPIO da plataforma. Para garantir a estabilidade do nível lógico durante a transmissão dos dados, foi utilizado um resistor de *pull-up* de $10\text{ k}\Omega$ conectado à linha de dados, valor que se encontra dentro da faixa recomendada pelo fabricante ($4,7\text{ k}\Omega$ a $10\text{ k}\Omega$). A utilização desse resistor contribui para a confiabilidade da comunicação e, consequentemente, da aquisição dos dados de temperatura e umidade.

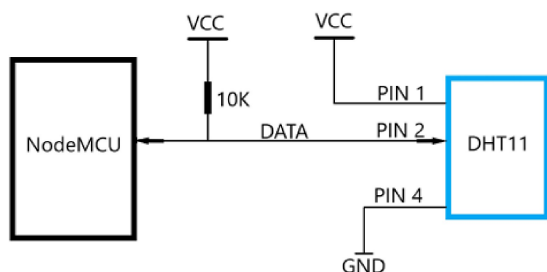


Figura 7 – Circuito de interfaceamento entre a plataforma NodeMCU e o sensor DHT11.

A plataforma NodeMCU realiza continuamente a aquisição dos dados de temperatura e umidade fornecidos pelo sensor DHT11 e os sincroniza com o FRD. O aplicativo móvel consulta essas informações

no banco de dados e apresenta ao usuário os valores monitorados em tempo real.

SISTEMA DE CONTROLE DA ILUMINAÇÃO

A Figura 8 apresenta o circuito de controle da iluminação, implementado por meio de um módulo relé de 5 Vcc acionado pela plataforma NodeMCU. A entrada de controle do relé é conectada a um pino GPIO configurado como saída digital, enquanto sua alimentação é fornecida por uma fonte de 5 Vcc comum ao sistema. Os contatos do relé realizam a comutação da alimentação da lâmpada conectada à rede elétrica de 127 Vca, utilizando o contato normalmente aberto (NO), de modo que a lâmpada permanece desligada na ausência de acionamento do relé. Essa configuração proporciona isolamento galvânico entre o circuito de controle e o circuito de potência, aumentando a segurança da plataforma embarcada.

O acionamento da iluminação é realizado remotamente por meio do aplicativo móvel. Quando o usuário solicita o acionamento ou desligamento de uma lâmpada, o comando é enviado ao FRD, onde é imediatamente sincronizado com a plataforma NodeMCU. O microcontrolador monitora continuamente as alterações registradas no banco de dados e, ao identificar uma mudança de estado, atualiza o nível lógico do pino GPIO correspondente, acionando ou desacionando o módulo relé e, consequentemente, ligando ou desligando a lâmpada.

Neste trabalho foram implementados dois pontos de iluminação, cada um controlado por um módulo relé independente conectado a um pino GPIO da plataforma NodeMCU. Essa arquitetura permite o acionamento individual de cada lâmpada por meio do aplicativo móvel, proporcionando maior flexibilidade ao sistema e facilitando a inclusão de novos pontos de iluminação conforme a disponibilidade de pinos GPIO e módulos relé.

SISTEMA DE CONTROLE DA PERSIANA

A Figura 9 apresenta o circuito utilizado para o controle da persiana. O sistema é composto por um motor de passo 28BYJ-48, acionado pelo driver ULN2003 e alimentado por uma fonte externa de 5 Vcc. As entradas de controle do driver (IN1 a IN4) são conectadas aos pinos GPIO da plataforma NodeMCU, configurados como saídas digitais, enquanto o terminal de referência (GND) é compartilhado entre a fonte, o driver e o microcontrolador. O driver ULN2003 recebe a sequência de sinais enviada pela plataforma NodeMCU e realiza o acionamento sequencial das bobinas do motor de passo, permitindo o controle do sentido de rotação da persiana. O controle da persiana é realizado remotamente por meio do aplicativo móvel, que disponibiliza três comandos:

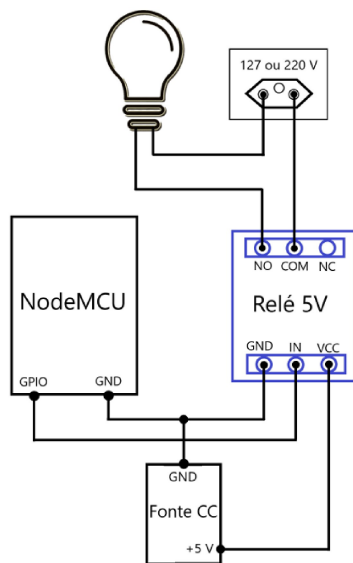


Figura 8 – Circuito de controle da iluminação utilizando a plataforma NodeMCU e módulo relé.

abrir, parar e fechar. Quando um desses comandos é acionado pelo usuário, a informação correspondente é armazenada no FRD e sincronizada com a plataforma NodeMCU. O microcontrolador monitora continuamente as alterações registradas no banco de dados e, ao detectar um novo comando, envia ao driver ULN2003 a sequência de sinais necessária para o acionamento do motor de passo. Dessa forma, o motor pode girar no sentido de abertura ou de fechamento da persiana, enquanto o comando parar interrompe a sequência de acionamento enviada ao motor, permitindo posicionar a persiana em qualquer abertura desejada.

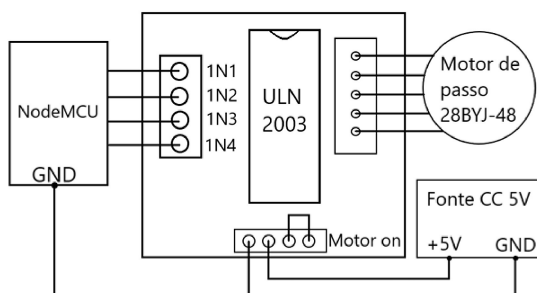


Figura 9. Circuito de controle do motor de passo da persiana.

SISTEMA DE CONTROLE DO PORTÃO ELETRÔNICO

O circuito utilizado para o controle do portão eletrônico é semelhante ao empregado no sistema de controle da persiana, apresentado na Figura 9, diferindo apenas na utilização de um conjunto distinto de pinos GPIO da plataforma NodeMCU para o acionamento do driver ULN2003. Assim como no sistema de controle da persiana, foi utilizado um motor de passo 28BYJ-48, alimentado

por uma fonte externa de 5 Vcc, enquanto a plataforma NodeMCU é responsável pela geração da sequência de acionamento aplicada às entradas do driver.

O controle do portão eletrônico foi implementado por meio de uma máquina de estados finitos (*Finite State Machine* – FSM), cujo diagrama é apresentado na Figura 10. A máquina é composta por quatro estados: E0, correspondente ao estado inicial, em que o portão permanece parado; E1, correspondente ao movimento de abertura; E2, em que o portão permanece novamente parado; e E3, correspondente ao movimento de fechamento. A Figura 10 apresenta, para cada estado, os valores assumidos pelas variáveis Motor e Sentido, responsáveis, respectivamente, pelo acionamento do motor de passo e pela definição do seu sentido de rotação. A cada acionamento do botão Portão Eletrônico no aplicativo ocorre a transição para o próximo estado da sequência E0 → E1 → E2 → E3 → E0, repetindo-se ciclicamente durante o funcionamento do sistema.

O comportamento da máquina de estados é determinado pelas variáveis Motor e Sentido, armazenadas no FRD e atualizadas sempre que o usuário aciona o botão de controle do portão no aplicativo. A plataforma NodeMCU monitora continuamente as alterações registradas no banco de dados e, ao detectar uma transição de estado, gera a sequência de acionamento correspondente e a envia ao driver ULN2003, que energiza as bobinas do motor de passo de acordo com o estado atual da máquina. Dessa forma, o sistema executa, de maneira sequencial e controlada, as operações de abertura, parada e fechamento do portão eletrônico.

Para garantir o correto funcionamento do sistema, foi implementado um mecanismo de *software* para evitar múltiplos acionamentos consecutivos do botão Portão Eletrônico no aplicativo móvel. Essa estratégia assegura que cada interação do usuário gere apenas uma única transição de estado, evitando o processamento repetitivo de comandos e garantindo o funcionamento adequado da máquina de estados.

SISTEMA DE CONTROLE DO ALARME

A Figura 11 apresenta o circuito utilizado no sistema de controle do alarme, evidenciando a integração entre a plataforma NodeMCU, o sensor de movimento PIR (*Passive Infrared Sensor*) e o módulo buzzer. Ambos os dispositivos são alimentados por uma fonte externa de 5 Vcc. A saída digital do sensor PIR é conectada a um pino GPIO da plataforma NodeMCU configurado como entrada, enquanto o buzzer é acionado por outro pino GPIO configurado como saída. Todos os dispositivos compartilham a mesma referência de terra (GND), garantindo o correto funcionamento do sistema.

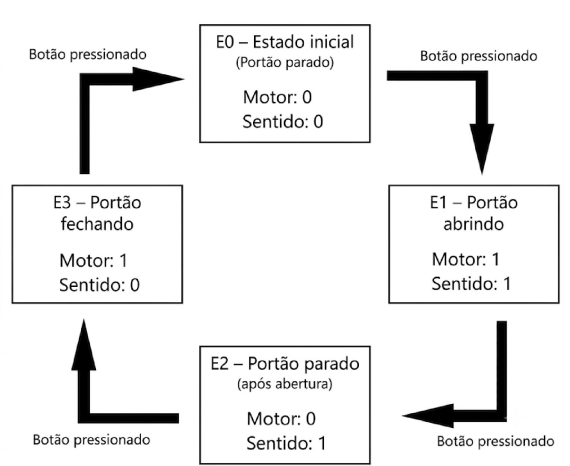


Figura 10. Máquina de estados finitos do sistema de controle do portão eletrônico.

O controle do alarme é realizado remotamente por meio do aplicativo móvel, que disponibiliza um botão com dois estados, Ligado e Desligado. O estado selecionado pelo usuário é armazenado no FRD e sincronizado com a plataforma NodeMCU. O microcontrolador monitora continuamente as alterações registradas no banco de dados, bem como o sinal proveniente do sensor de movimento PIR.

Quando o sistema de alarme está habilitado (Ligado) e o sensor PIR detecta movimento, a plataforma NodeMCU aciona o buzzer, emitindo um sinal sonoro de alerta. Caso o sistema esteja desabilitado (Desligado), o buzzer permanece desacionado, independentemente da detecção de movimento pelo sensor. Da mesma forma, quando o sistema está habilitado, mas não ocorre detecção de movimento, o buzzer também permanece desacionado. Essa estratégia garante que o alarme seja acionado exclusivamente quando o monitoramento estiver habilitado e houver detecção de movimento, evitando acionamentos indevidos.

PROGRAMAÇÃO DA PLATAFORMA NODEMCU

A programação da plataforma NodeMCU foi desenvolvida na Arduino IDE, utilizando bibliotecas específicas para a plataforma ESP8266, comunicação com o FRD, aquisição dos dados do sensor DHT11 e acionamento dos demais dispositivos do sistema.

Inicialmente, foram definidos os parâmetros de configuração do sistema, incluindo as credenciais da rede Wi-Fi, as informações de autenticação do FRD, o modelo do sensor DHT11 e os pinos GPIO utilizados para conexão dos sensores e atuadores. Em seguida, foram inicializadas as variáveis responsáveis pelo armazenamento dos dados de temperatura, umidade, estado da iluminação, controle da persiana, controle do portão eletrônico e sistema de alarme.

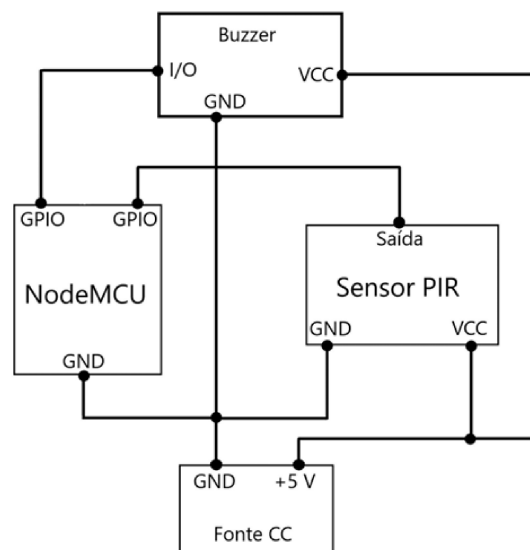


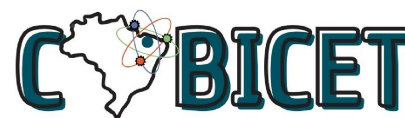
Figura 11. Circuito do sistema de controle do alarme.

Durante a inicialização da plataforma, são configurados os pinos GPIO como entradas ou saídas, estabelecida a conexão com a rede Wi-Fi e realizada a autenticação junto ao FRD. Concluída essa etapa, a plataforma NodeMCU passa a operar continuamente, realizando a aquisição dos dados dos sensores de temperatura, umidade e movimento e enviando essas informações ao FRD para atualização do banco de dados em nuvem. Simultaneamente, a plataforma monitora as alterações registradas no FRD, correspondentes aos comandos enviados pelo aplicativo móvel, e, quando detecta uma mudança de estado, executa a rotina de controle correspondente, acionando os módulos relé, o motor de passo da persiana, a máquina de estados responsável pelo controle do portão eletrônico ou o buzzer do sistema de alarme, conforme o comando recebido.

RESULTADOS E DISCUSSÃO

Os experimentos realizados demonstraram o correto funcionamento da arquitetura proposta, validando a integração entre a plataforma NodeMCU, o FRD, o aplicativo móvel, os sensores e os atuadores. Todos os módulos implementados, controle da iluminação, controle da persiana, controle do portão eletrônico, monitoramento de temperatura e umidade e sistema de alarme, operaram de forma integrada, executando corretamente as funções previstas durante o desenvolvimento.

O aplicativo móvel apresentou boa estabilidade durante os testes, permitindo o envio dos comandos de controle e a visualização das informações provenientes dos sensores em tempo real. A comunicação com o FRD ocorreu de forma consistente, possibilitando a sincronização entre o aplicativo e a plataforma NodeMCU sem falhas observadas durante a validação experimental.



Os módulos de hardware também apresentaram comportamento compatível com o esperado. Os módulos relé responderam corretamente aos comandos de acionamento da iluminação, o driver ULN2003 realizou o controle do motor de passo nos sistemas da persiana e do portão eletrônico, o sensor de movimento PIR detectou adequadamente os eventos de movimento e o buzzer foi acionado sempre que as condições definidas para o sistema de alarme foram satisfeitas.

No que se refere ao tempo de resposta dos subsistemas de controle remoto, observou-se que a latência percebida pelo usuário é resultado da composição de três etapas sequenciais: o processamento e envio do comando pelo aplicativo móvel, a sincronização da informação com o FRD e o processamento executado pela plataforma NodeMCU antes do acionamento dos dispositivos. Esse comportamento é inerente à arquitetura baseada em banco de dados em nuvem e foi considerado satisfatório para aplicações de automação residencial, nas quais não se exige resposta em tempo real estrito.

O sistema de alarme apresentou comportamento distinto dos demais subsistemas em relação à latência de resposta. Nesse módulo, a decisão de acionamento do buzzer é tomada diretamente pela plataforma NodeMCU com base no sinal do sensor PIR, sem depender da comunicação com o FRD. A comunicação em nuvem é utilizada exclusivamente para habilitar ou desabilitar o monitoramento, enquanto a lógica de detecção permanece embarcada no microcontrolador. Essa arquitetura reduz significativamente a latência de resposta do alarme em comparação aos subsistemas que dependem integralmente da sincronização com o banco de dados em nuvem.

O monitoramento de temperatura e umidade foi realizado por meio do sensor DHT11. Durante os experimentos, as leituras obtidas foram adquiridas pela plataforma NodeMCU, sincronizadas com o FRD e disponibilizadas em tempo real no aplicativo móvel, confirmando o correto funcionamento da arquitetura proposta para o monitoramento remoto de variáveis ambientais.

De modo geral, os resultados obtidos demonstram que a arquitetura proposta permitiu integrar, em uma única plataforma de baixo custo, o monitoramento remoto de variáveis ambientais e o controle de dispositivos elétricos residenciais, utilizando exclusivamente tecnologias abertas e componentes eletrônicos de ampla disponibilidade comercial.

CONCLUSÃO

Este trabalho apresentou a implementação de uma arquitetura de automação residencial baseada em Internet das Coisas, integrando a plataforma

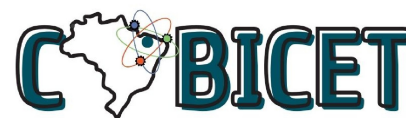
NodeMCU, o Firebase Realtime Database e um aplicativo móvel desenvolvido em Flutter. A solução reuniu, em uma única plataforma, funcionalidades de monitoramento de temperatura, umidade e movimento, além do controle remoto da iluminação, da persiana motorizada, do portão eletrônico e do sistema de alarme. Os resultados experimentais confirmaram o correto funcionamento de todos os módulos implementados, evidenciando a viabilidade da arquitetura proposta para aplicações de automação residencial de baixo custo baseadas em IoT.

A utilização do FRD como camada intermediária de comunicação mostrou-se uma estratégia eficaz para desacoplar o aplicativo móvel da plataforma embarcada, permitindo o monitoramento e o controle remotos a partir de qualquer localização com acesso à Internet. A implementação do portão eletrônico por meio de uma máquina de estados finitos demonstrou ser uma abordagem robusta para o gerenciamento de sequências de controle com múltiplos estados, evitando transições indevidas decorrentes de acionamentos consecutivos.

Como desdobramentos naturais deste trabalho, propõe-se a expansão do sistema para outros cômodos da residência, a inclusão de um modo de operação local para situações de ausência de conectividade e a integração de novos sensores e atuadores, ampliando as funcionalidades da solução proposta. Adicionalmente, a realização de ensaios de caracterização do desempenho da arquitetura constitui uma direção promissora para a avaliação quantitativa do sistema em diferentes condições de operação.

REFERÊNCIAS

- Addow, M. A., Jimale, A. D., Nageye, A. Y., Abdullahi, M. O. e Hilowle, S. M. A low-cost IoT-based smart home automation system for urban sustainability in Mogadishu, Somalia. *Discover Internet of Things*, v. 5, art. 99, 2025.
- Ayeni, P. O. e Adesoba, O. C. IoT-based home control system using NodeMCU and Firebase. *Journal of Edge Computing*, v. 4, n. 1, p. 17–34, 2025.
- Poyyamozi, M., Murugesan, B., Rajamanickam, N., Shorfuzzaman, M. e Aboelmagd, Y. IoT — A promising solution to energy management in smart buildings: a systematic review, applications, barriers, and future scope. *Buildings*, v. 14, n. 11, art. 3446, 2024.
- Stoljescu-Crişan, C., Crisan, C. e Butunoi, B.-P. An IoT-based smart home automation system. *Sensors*, v. 21, n. 11, art. 3784, 2021.
- Tripathy, S. e Das, S. S. Automated home using Firebase and Google Assistant. *Journal of*



Information and Optimization Sciences, v. 43, n. 5, p.
1021–1028, 2022.