

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

GABRIEL OTAVIO DE CARLI

**AUTOMAÇÃO RESIDENCIAL CONTROLADA POR APLICATIVO UTILIZANDO
A PLATAFORMA ESP32**

TOLEDO

2023

GABRIEL OTAVIO DE CARLI

**AUTOMAÇÃO RESIDENCIAL CONTROLADA POR APLICATIVO UTILIZANDO
A PLATAFORMA ESP32**

Application Controlled Home Automation Using the ESP32 Platform

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do
título de Bacharel em Engenharia Eletrônica
do Curso de Bacharelado em Engenharia
Eletrônica da Universidade Tecnológica Federal
do Paraná.

Orientador: Prof. Alessandro Paulo De Oliveira

TOLEDO

2023



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

GABRIEL OTAVIO DE CARLI

**AUTOMAÇÃO RESIDENCIAL CONTROLADA POR APLICATIVO UTILIZANDO
A PLATAFORMA ESP32**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do
título de Bacharel em Engenharia Eletrônica
do Curso de Bacharelado em Engenharia
Eletrônica da Universidade Tecnológica Federal
do Paraná.

Data de aprovação: 06/dezembro/2023

Prof. Dr. Alessandro Paulo de Oliveira
Orientador
Universidade Tecnológica Federal do Paraná

Prof. Dr. Jose Dolores Vergara Dietrich
Universidade Tecnológica Federal do Paraná

Prof. Dr. Gerson Filippini
Universidade Tecnológica Federal do Paraná

TOLEDO
2023

RESUMO

O artigo aborda a escassa popularização da automação residencial devido aos custos elevados. Para superar essa barreira, o projeto propõe um modelo de sistema residencial automatizado, destacando a eficácia e acessibilidade. Utilizando o microcontrolador ESP32, conhecido por sua confiabilidade e flexibilidade, o protótipo de baixo custo controla dispositivos elétricos fundamentais. A conclusão destaca a necessidade de análise de custos de *software* e a busca por fornecedores para a viabilidade comercial em larga escala. O estudo sugere direções para futuros projetos na área.

Palavras-chave: automação; microcontrolador; esp32.

ABSTRACT

The article addresses the limited adoption of home automation due to associated high costs. To overcome this barrier, the project proposes a model of an automated residential system, emphasizing effectiveness and accessibility. Leveraging the ESP32 microcontroller, known for its reliability and flexibility, the low-cost prototype controls essential electrical devices. The conclusion highlights the need for a comprehensive analysis of software development costs and the search for suppliers to enable widespread commercial viability. The study suggests directions for future projects in the field.

Keywords: automation; microcontroller; esp32.

LISTA DE FIGURAS

Figura 1 – Periféricos de uma automação residencial	12
Figura 2 – ESP32	18
Figura 3 – Diagrama de blocos do sistema completo	21
Figura 4 – Módulo de relés modelo JQC-3FF-S-Z	22
Figura 5 – Motor de passo 28BYJ-48	23
Figura 6 – Sensor de presença - módulo PIR DYP-ME003	24
Figura 7 – Sensor de gases - módulo MQ-2	24
Figura 8 – Sensor de temperatura e umidade - módulo DHT11	25
Figura 9 – Diagrama de conexões	26
Figura 10 – Variáveis no banco de dados	27
Figura 11 – Inclusão de endereço para adição do ESP32 no Arduino IDE	28
Figura 12 – <i>Layout</i> do aplicativo desenvolvido	29
Figura 13 – Fluxograma da programação ESP32	31
Figura 14 – Fluxograma da programação do aplicativo	32
Figura 15 – Testes com o módulo de relés	32
Figura 16 – Testes com os sensores e motor de passo	33

LISTA DE ABREVIATURAS E SIGLAS

Siglas

CC	Corrente Contínua
FreeRTOS	<i>Free Real-Time Operating System</i>
GPIO	<i>General Purpose Input Output</i>
IDE	<i>Integrated Development Environment</i>
IoT	<i>Internet of Things</i>
NoSQL	<i>Not Only Structured Query Language</i>
ppm	partes por milhão

SUMÁRIO

1	INTRODUÇÃO	8
1.1	Problema	8
1.2	Objetivos	9
1.2.1	Objetivos específicos	9
1.3	Justificativa	9
1.4	Metodologia	9
2	REVISÃO LITERÁRIA	11
2.1	Automação Residencial	11
2.1.1	Atual Mercado da Automação	12
2.2	Eletrônica e Componentes Eletrônicos	13
2.2.1	Relés	14
2.2.2	Sensores	14
2.2.2.1	Sensores de temperatura	14
2.2.2.2	Sensores de umidade	14
2.2.2.3	Sensores de presença	15
2.2.3	Motor Elétrico	15
2.3	Internet das coisas	15
2.3.1	Redes sem fio	16
2.4	Firestore	16
2.5	Visual Studio Code	16
2.6	Flutter	17
2.7	Microcontroladores	17
2.7.1	ESP32	17
2.8	Linguagem de programação	18
2.8.1	Linguagem C/C++	18
2.8.2	DART	19
3	DESENVOLVIMENTO	20
3.1	Levantamento de dados	20
3.2	Funcionamento do sistema	21
3.2.1	Ponto de iluminação e sistema de ventilação	21

3.2.2	Acionamento da persiana	22
3.2.3	Sistema de monitoramento e alarmes	23
3.2.4	Monitoramento de temperatura	24
3.3	Diagrama e componentes do projeto	25
3.4	Banco de dados	26
3.5	Programação do ESP32	27
3.6	Interface com o usuário	29
3.7	Fluxograma	30
3.8	Testes de funcionamento	30
3.8.1	Testes do modulo relé	30
3.8.2	Testes dos sensores e motor de passo	30
3.9	Dificuldades no desenvolvimento	32
4	RESULTADOS	34
4.1	Protótipo	34
4.2	Aplicativo	34
4.3	Valores	34
5	CONCLUSÃO	36
	REFERÊNCIAS	37
	ANEXO A	40
	ANEXO B	45
	ANEXO C	48

1 INTRODUÇÃO

O século XVIII foi marcado pela construção da máquina a vapor, o que resultou no aumento da produção de artigos manufaturados, marcando o início da revolução industrial. Nos anos seguintes, com as indústrias tomando formas grandiosas, foram criados dispositivos mecânicos chamados relés, que foram aperfeiçoados e tornaram o controle da indústria mais eficiente e centralizado, dando origem ao termo "Automação Industrial"(SILEVIRA, 2003). A atuação de dispositivos de forma supervisionada ou não, possibilita a execução de tarefas complexas, interagindo com o meio físico e com pessoas, o que inclui a automação residencial (BOLZANI, 2004).

A automação está presente na vida das pessoas de diversas formas, desde simples mecanismos como portões e travas eletrônicas, até sistemas mais complexos, como eletrodomésticos, controle de sistemas de energia, sistemas hidráulicos, entre outros (OGATA, 2013). A expansão da automação industrial foi impulsionada pela necessidade de produção, enquanto a expansão da automação residencial pode ser atribuída ao desenvolvimento da eletrônica e à redução de custos dos componentes. Isso torna a automação residencial mais acessível, aumentando a demanda no mercado (WORTMEYER FERNANDO FREITAS, 2005).

A automação residencial oferece vários benefícios, com destaque para a economia de energia, conforto e praticidade. Um dos itens mais flexíveis para automação são as lâmpadas, que podem ser controladas manualmente por meio de dispositivos eletrônicos, como celulares e tablets, ou automaticamente por sensores e atuadores programáveis. Além de tornar o lar mais sofisticado, a automação residencial oferece maior segurança e eficiência energética, permitindo o monitoramento e controle por meio de dispositivos móveis (BOLZANI, 2007).

A automação residencial abrange uma ampla gama de componentes eletrônicos que possibilitam a criação de sistemas integrados para o controle de lâmpadas, consumo de energia, qualidade do ar, entre outros. Cada desenvolvedor utiliza o método de desenvolvimento com o qual está mais familiarizado, tornando-o prático e confiável para atender às necessidades dos usuários finais.

1.1 Problema

Embora os componentes eletrônicos estejam amplamente disponíveis e o avanço da tecnologia tenha reduzido os custos, a automação residencial ainda é pouco acessível para a maioria das residências brasileiras devido ao alto custo de implementação de sistemas automatizados. Como é possível tornar a automação residencial mais acessível financeiramente aos usuários?

1.2 Objetivos

Este trabalho possui como objetivo principal desenvolver um protótipo para automatizar os principais periféricos de uma residência, como lâmpadas, ventiladores, persianas e sensores de gases tóxicos, controlados e monitorados por um aplicativo desenvolvido para dispositivos móveis.

1.2.1 Objetivos específicos

Os objetivos específicos do projeto são:

- Levantamento das variáveis do ambiente.
- Instalação integrada de todos os componentes e dispositivos.
- Desenvolvimento de um aplicativo para dispositivos móveis para o comando e supervisão.
- Análise do microcontrolador ESP32 para a finalidade proposta.
- Programação do microcontrolador.
- Montagem de um protótipo para testar a funcionalidade do sistema.
- Análise dos resultados.

1.3 Justificativa

Segundo uma matéria publicada pela Aureside, a automação residencial cresceu 12,5% entre 2015 e 2020 (AURESIDE, 2016). Uma pesquisa da GS1 mostrou que, entre 3.795 entrevistados, 83% planejam melhorar a automação em suas vidas diárias. Dos 17% restantes que não estão interessados, 27% alegam que o alto custo é o motivo (BRASIL, 2017).

Diante desses dados, fica claro que para aumentar a implementação de sistemas automatizados em residências brasileiras, é necessário torná-los mais acessíveis, oferecendo comodidade e praticidade sem comprometer significativamente os recursos financeiros. Isso justifica o presente trabalho.

1.4 Metodologia

Este trabalho emprega atividades de pesquisa para identificar os principais periféricos relevantes para a elaboração do projeto de automação residencial. O objetivo principal é controlar os dispositivos elétricos de forma a reduzir o consumo de energia e melhorar o conforto e a

segurança de maneira eficiente. Inicialmente, é necessário realizar uma análise para identificar as variáveis de maior relevância para o projeto, como microcontroladores, sensores, lâmpadas e motores. Em seguida, será realizada a implementação completa do sistema, seguida por testes para comprovar a eficácia e identificar possíveis falhas no sistema. Por fim, serão apresentados dados documentados para análise de custos e as principais vantagens da utilização do sistema.

2 REVISÃO LITERÁRIA

A fim de realizar o desenvolvimento do trabalho, foi necessário aplicar diferentes assuntos abordados durante o curso, tais como linguagens de programação, microcontroladores, sensores e atuadores. Além dos conteúdos específicos do curso, foi necessário estudar sobre Internet das Coisas. Portanto, neste capítulo, será apresentada uma revisão bibliográfica dos conceitos abordados.

2.1 Automação Residencial

A ideia de automação consiste basicamente em utilizar potência elétrica ou mecânica para acionar algum tipo de máquina, sendo essa máquina programada com a inteligência necessária para que ela desempenhe suas tarefas de forma mais eficiente e vantajosa economicamente (RIBEIRO, 2001).

Com a criação do circuito integrado em meados dos anos 1960 e do microprocessador em 1970, a possibilidade de adicionar inteligência às máquinas aumentou significativamente, assim como a liberdade de reprogramação desses chips tornou o processo mais econômico. No ano 2000, já havia uma predominância da eletrônica microprocessada na área industrial (RIBEIRO, 2001).

O desenvolvimento da tecnologia possibilitou a consolidação dos sistemas autônomos, com uma ramificação voltada especificamente para residências, proporcionando principalmente segurança e conforto. Chamada de automação residencial, essa ramificação também pode ser definida como domótica, originada da fusão das palavras "*domus*", que significa "casa" em latim, com robótica (BOLZANI, 2004).

A domótica impulsiona a progressão na interconexão de dispositivos domésticos, destacando-se pela conectividade sem fio. Isso facilita a interação entre aparelhos eletrônicos, centralizando o controle e permitindo uma supervisão eficaz, bem como reprogramação simples. Essa flexibilidade é essencial para ajustar as configurações de acordo com as necessidades dos usuários, e essa centralização de controle pode ser facilmente incorporada, por exemplo, por meio de um dispositivo móvel, *smartphone* ou *tablet* (MURATORI, 2011).

O sistema de automação residencial completo possui dispositivos de entrada e dispositivos de saída, e essa integração deve ter uma percepção aprofundada do espaço físico, tanto na esfera sensorial quanto na esfera de controle. Os dispositivos de entrada podem ser sensores de iluminação, sensores de presença, câmeras, entre outros, enquanto os dispositivos de saída podem ser sistemas de ventilação, sistemas de iluminação, entre outros (BOLZANI, 2004).

Na Figura 1, são apresentados, de forma ilustrativa, itens que podem ser incluídos e manipulados por um sistema de automação residencial.

Os dispositivos interconectados permitem a comunicação através da rede mundial de computadores, seja ela sem fio ou com fio, o que dá origem ao conceito de *Internet of Things*

Figura 1 – Periféricos de uma automação residencial



Fonte: Securitел 2023.

(IoT). A IoT visa coletar, processar e compartilhar informações na rede, viabilizando ações remotas, como a dimerização da iluminação por meio de sensores de luminosidade, o acionamento do ar-condicionado controlado por sensores de temperatura, entre outros sistemas similares (AFONSO ROBERTO BENEDITO DE O. PEREIRA, 2015).

2.1.1 Atual Mercado da Automação

Os sistemas de controle residencial estão se tornando cada vez mais integrados ao cotidiano das habitações modernas. O controle programado em sistemas computacionais para comandar alarmes, sistemas de aquecimento e outras aplicações é bem-vindo em residências. No vasto campo da automação, muitas tecnologias vêm se desenvolvendo para acrescentar mais poder ao sistema. Entre esses desenvolvimentos, é possível citar o controle do sistema de iluminação como o mais comum, proporcionando economia e conforto (KELLY, 1997).

A automação é amplamente difundida nos países desenvolvidos. No entanto, a falta de padronização e uniformidade de protocolos de comunicação para sistemas de automação residencial é frequentemente um problema, o que faz com que o projeto de automação tenha que se adaptar a construções já existentes, encarecendo a implementação e utilização do sistema (MAINARDI S. BANZI, 2005).

Para analisar o mercado da automação, deve-se levar em consideração os avanços na comunicação que possibilitaram a conexão de componentes de diversos fabricantes diferentes. Estes, que no passado dificultavam a interconexão dos equipamentos e a globalização dos sistemas de automação. Este problema foi superado após uma série de sistemas promoverem uma evolução na comunicação entre os mais variados tipos de equipamentos, como o sistema *LonWorks*, desenvolvido ainda na década de 90, o sistema *European Home System*, o sistema *CEBus*, o sistema *BatiBus*, entre outros, promoveram a conexão entre dispositivos de diferentes fabricantes (BRANDÃO, 2008).

Felipe Belmonte, um profissional especialista em projetos de áudio e vídeo, automação e soluções residenciais de alto padrão, que atua na empresa Acatar Automação Residencial, aponta que as melhores empresas desenvolvedoras de equipamentos aplicáveis em automação residencial são: a *Lutron* (atuante no mercado desde 1961), a *Crestron* (atuante no mercado desde 1971), a *Savant* (atuante no mercado desde 2005), a *Scenario* e a *Controlart*. Ele também menciona que o preço de aquisição do sistema é variável, dependendo dos requisitos do projeto desenvolvido de acordo com o perfil do usuário final. No entanto, Felipe Belmonte afirma que uma empresa certificada pode cobrar um valor inicial de R\$ 8.000,00 para pequenas residências, chegando a um valor de até R\$ 40.000,00 em instalações de alto padrão. Ele destaca ainda que esse é o valor investido apenas em profissionais especializados, sem incluir o investimento na aquisição dos equipamentos (BELMONTE, 2022).

Por meio desta breve análise, podemos concluir que implementar a automação residencial e definir como controlador principal o ESP32, com componentes alternativos aos mais caros utilizados comercialmente em sistemas de automação residencial, como sensores e acionadores de baixo custo, torna viável o desenvolvimento do sistema com um menor custo, possibilitando controlar e monitorar uma residência à distância sem depender de conexão cabeada para a comunicação do sistema, controlando-o por um aplicativo instalado em dispositivo móvel.

2.2 Eletrônica e Componentes Eletrônicos

A eletrônica constitui o âmbito da ciência e engenharia que se dedica à investigação dos modos de gerenciar a energia elétrica por meio de dispositivos e meios condutivos ou semicondutores. Um dos marcos iniciais do estudo nessa área foi a descoberta dos raios catódicos por Hittford em 1869. Maxwell, com sua teoria do eletromagnetismo, previu por cálculos a existência de ondas eletromagnéticas, que foram detectadas e estudadas por Hertz em 1888. Em 1895, Marconi realizou a primeira aplicação da eletrônica, transmitindo sinais de telegrafia sem fio a dezenas de quilômetros de distância através da antena de transmissão, inventada por Popov no mesmo ano. Em seguida, houve a criação das válvulas eletrônicas, dos diodos e dos triodos. Esses componentes possibilitam a manipulação de ondas eletromagnéticas. A fase de miniaturização dos equipamentos começou em 1947, após a descoberta dos transistores. Nos anos

60, foi possível desenvolver vários transistores em um substrato de silício, surgindo a integração de componentes em larga escala (FUENTES, 2013).

Os componentes eletrônicos são responsáveis pelo controle total do fluxo de corrente elétrica que flui através de um circuito, construídos por materiais que podem ser denominados como isolantes, semicondutores e condutores. O estudo dos semicondutores é o principal fator para a existência da eletrônica (SCHULER, 2013).

2.2.1 Relés

Os relés são chaves eletromecânicas, onde seus comutadores se movimentam através do acionamento de um eletroímã. Os relés possuem duas configurações de funcionamento: normalmente fechado, caso em que seus comutadores se mantêm em contato quando o eletroímã está desenergizado, e o normalmente aberto, caso em que os comutadores se mantêm afastados e se aproximam após o acionamento do eletroímã. Os relés são usados para acionar cargas de elevada potência, como o acionamento de motores potentes que consomem grandes valores de corrente a partir de um circuito de baixa potência (circuito digital) (SILVA, 2012).

Esses componentes são amplamente utilizados nos setores industriais. No mercado eletrônico atual, é fácil encontrar módulos de relés prontos para uso, com conexões de acionamento e devidamente protegidos para utilização segura.

2.2.2 Sensores

O termo sensor é empregado para denominar dispositivos sensíveis a variações de grandezas físicas, como temperatura, velocidade, umidade, e outros, que as transformam em variações de grandezas elétricas, como tensão ou corrente (WENDDLING, 2010).

2.2.2.1 Sensores de temperatura

Sensores de temperatura detectam a variação de temperatura no ambiente em que estão localizados, transformando essa variação em uma mudança na tensão ou corrente. Por exemplo, o módulo DHT11 mede a temperatura por meio de um componente resistivo que aumenta sua resistência à medida que a temperatura sobe, resultando em um aumento na tensão na saída do componente (SRIVASTAVA AWANISH KESARWANI, 2018).

2.2.2.2 Sensores de umidade

Sensores de umidade detectam a variação da umidade no ambiente em que estão localizados, transformando essa variação em uma mudança na tensão ou corrente. Por exemplo, o

módulo DHT11 mede a umidade por meio de um componente resistivo que diminui sua resistência à medida que a umidade aumenta, resultando em uma diminuição na tensão na saída do componente (SRIVASTAVA AWANISH KESARWANI, 2018).

2.2.2.3 Sensores de presença

Os dispositivos de detecção de presença são compostos por infravermelho passivo e funcionam reconhecendo mudanças no ambiente em que estão inseridos por meio da detecção de raios infravermelhos. Sua utilização visa identificar o deslocamento e as entradas de indivíduos no raio de sua detecção (SANTOS JuLIO CeZAR COÊLHO BARBOSA TORQUATO, 2017).

2.2.3 Motor Elétrico

Um motor é um atuador que transforma energia elétrica em energia mecânica de rotação. Um motor de Corrente Contínua (CC) é composto por três principais partes: rotor ou armadura, comutador por escovas e bobinas de campo. O rotor é o elemento rotativo acoplado ao eixo, onde recebe energia elétrica através de anéis deslizantes acoplados ao mesmo eixo, que giram através da interação do campo magnético emitido pelas bobinas de campo. Em motores de baixa potência, as bobinas de campo são substituídas por ímãs permanentes (PATANE, 2008).

2.3 Internet das coisas

A Internet das Coisas - IoT (do inglês "Internet of Things") é uma extensão da rede atual que possibilita a conexão com objetos do dia a dia, com uma conexão direta a computadores e sistemas de comunicação. A conexão com a internet viabiliza o controle desses objetos remotamente, permitindo que os próprios objetos atuem como atuadores de serviços. A evolução da IoT emergiu do avanço de várias áreas em conjunto, como sistemas embarcados, micro-eletrônica, comunicação e sensoriamento. Ela teve uma evolução constante em paralelo com a evolução de seus componentes. A evolução do *hardware* possibilitou maior capacidade de armazenamento e processamento, e a evolução do *software* possibilitou um maior número de funcionalidades, melhorando a capacidade de processamento, padronizando atividades e popularizando o desenvolvimento de programas e aplicativos. Tudo isso em conjunto popularizou o termo IoT, permitindo o controle de aparelhos eletrônicos por meio de programas computacionais através da conexão com a internet. (SANTOS LUCAS A. M. SILVA, 2016).

2.3.1 Redes sem fio

As redes sem fio são um sistema flexível de transmissão de dados, que podem ser usadas como uma extensão ou alternativa para redes locais. Essa tecnologia combina conectividade de dados com mobilidade, utilizando a tecnologia de radiofrequência. Devido às necessidades dos usuários de desfrutarem da mobilidade, esse tipo de conexão é amplamente utilizada devido à facilidade de uso e versatilidade. (MORAES, 2010). Dessa maneira, a popularidade desse modo de conexão torna viável a sua utilização na automação residencial, proporcionando simplicidade na construção estrutural e economia.

2.4 Firebase

O Firebase é um banco de dados *Not Only Structured Query Language* (NoSQL), traduzido para o português "Banco de Dados Não Relacionais", que permite aos usuários armazenar e acessar dados de forma simples. Devido à sua plataforma ser básica e unificada, ela é amplamente utilizada no desenvolvimento de aplicativos que necessitam de compartilhamento e armazenamento de dados. (CHATTERJEE SOUVIK CHAKRABORTY, 2018). Este sistema é conhecido pela facilidade de uso, documentação abrangente e integração perfeita com várias ferramentas do *Google*. É uma solução versátil para desenvolvedores que desejam criar aplicativos robustos, escaláveis e interativos, tanto para dispositivos móveis quanto para a web.

2.5 Visual Studio Code

O *Visual Studio Code* mais conhecido como VSCode, é um ambiente de desenvolvimento integrado *Integrated Development Environment* (IDE) de código aberto desenvolvido pela *Microsoft*. É amplamente utilizado por programadores e desenvolvedores para escrever, editar e depurar código em várias linguagens de programação. O VSCode é conhecido por sua leveza, flexibilidade e extensibilidade, o que o torna uma escolha popular entre desenvolvedores de diferentes plataformas e projetos. (VISUAL. . . ,).

O VSCode oferece uma ampla gama de recursos, incluindo realce de sintaxe, auto-completar, integração com controle de versão, depuração, terminal integrado, gerenciamento de extensões e muito mais. Uma característica distintiva do VSCode é sua capacidade de ser estendido por meio de uma vasta biblioteca de extensões disponíveis em sua loja *online*. Isso permite que os desenvolvedores personalizem e adaptem o ambiente conforme suas necessidades, adicionando suporte para diferentes linguagens, frameworks e ferramentas. (VISUAL. . . ,).

2.6 Flutter

O Flutter é um framework de código aberto desenvolvido pela Google que permite a criação de aplicativos nativos para diferentes plataformas a partir de um único código-base. Ele é especialmente conhecido por facilitar a construção de interfaces de usuário ricas e interativas, com uma abordagem de desenvolvimento baseada em widgets. (FLUTTER,). É uma ferramenta poderosa para o desenvolvimento de aplicativos multiplataforma, permitindo que os desenvolvedores criem aplicativos bonitos, rápidos e consistentes em várias plataformas, com um único conjunto de código.

2.7 Microcontroladores

Em geral, microcontroladores podem ser definidos como processadores encapsulados com memória, interface de entrada e saída de dados e periféricos. Esses periféricos podem ser conversores analógico para digital, conversores digital para analógico, temporizadores, interface de comunicação serial, entre outros. De forma geral, eles são como computadores de baixa capacidade de processamento. Microcontroladores são comumente empregados em robótica, visão computacional e automação. (BORGES ANTÔNIO PADILHA LANARI BO, 2008).

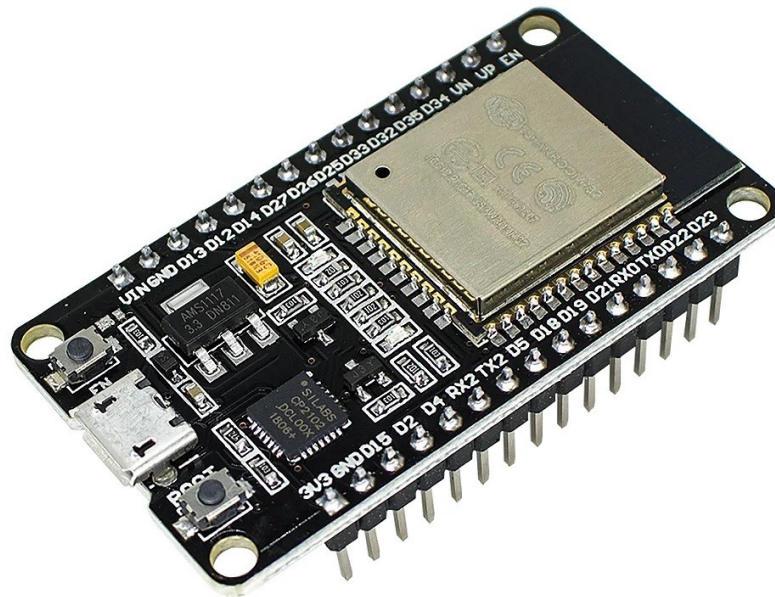
2.7.1 ESP32

O microcontrolador ESP32 é uma plataforma de desenvolvimento de código aberto, de baixo custo e baixa potência, produzido pela empresa *Espressif Systems*. Ele foi lançado no mercado no ano de 2016 e é um microcontrolador robusto, caracterizado por sua alta velocidade de processamento e boa conectividade através da conexão de rede sem fio. Este dispositivo, apresentado na Figura 2, possui um microprocessador dual core de 32 bits, que opera a 240 MHz, com conexões de internet sem fio e Bluetooth, além de contar com uma grande capacidade de armazenamento. (MACHESO, 2022).

Em seguida é apresentado as principais características no ESP32:

- Conexão *Wi-fi* 2,4 GHz com taxa de transmissão máxima de 150Mbps.
- Padrão de transmissão sem fio: *WiFi* 802.11 b/g/n: 2.4 à 2.5 GHz.
- Tensão de alimentação via Micro-USB: 4,5 à 12,0 Volts.
- Tensão de nível lógico: 3,3 Volts.
- Corrente de consumo tipicamente de 80 mA e máxima de 500 mA.
- Contém 32 portas *General Purpose Input Output* (GPIO).

Figura 2 – ESP32



Fonte: Saravati 2023.

A programação descomplicada do ESP32 é viável por meio de várias ferramentas compatíveis. Destacam-se linguagens de programação como C/C++, as quais podem ser elaboradas utilizando o kit de desenvolvimento de *software* disponibilizado pela fabricante do microcontrolador. Alternativamente, é amplamente utilizado o ambiente de desenvolvimento *Arduino IDE*. Devido aos fatos evidenciados pelas referências, torna-se viável a aplicação do ESP32 como controlador do sistema de automação residencial devido ao seu baixo custo e alta performance.

2.8 Linguagem de programação

Uma linguagem de programação é um método padronizado que usamos para expressar as instruções de um programa de computador programável, sendo que este realizará a leitura do código compilado e executará a determinada tarefa descrita no programa (GOTARDO, 2015).

2.8.1 Linguagem C/C++

A linguagem de programação C é uma das linguagens mais influentes e amplamente utilizadas na história da computação. Foi desenvolvida inicialmente por *Dennis Ritchie* nos laboratórios *Bell da AT&T* nos anos 1970. Sua influência é tão significativa que muitas linguagens de programação modernas têm elementos que derivam diretamente do C ou foram influenciadas por suas características. (DEITEL, 1999).

C++ é uma linguagem de programação que se origina do C, mas com recursos adicionais, tais como duas estruturas essenciais que a tornam inerentes, conhecidas como classes e funções, tornando-a uma linguagem mais poderosa e versátil. Ela foi desenvolvida por *Bjarne*

Stroustrup nos anos 1980 com o objetivo de melhorar as capacidades do C, mantendo sua eficiência e controle de *hardware*. (DEITEL, 2006).

2.8.2 DART

Dart é uma linguagem de programação estruturada, *open source* e puramente orientada a objetos. Foi desenvolvida pela empresa Google para a criação de aplicações *web* complexas, baseadas em navegadores *web*. Uma das metas dos criadores desta linguagem era que *Dart* fosse o mais familiar possível. A sintaxe da linguagem Dart em si é muito semelhante ao C. Uma de suas várias características é o modelo de simultaneidade chamado de *Isolates*, que permite execuções paralelas. (WOLFF, 2012)

3 DESENVOLVIMENTO

O presente projeto voltou-se para o desenvolvimento de um protótipo de baixo custo e baixa complexidade, focado na automação de acionamentos elétricos existentes em uma residência, oferecendo conforto e acessibilidade para todos os usuários, caso fosse de fato implementado em uma residência. A aplicação tomou como referência uma residência de pequeno porte. Para avaliar sua funcionalidade, criou-se um modelo compacto que serviu de representação. Isso abrirá caminho para sua futura utilização em maquetes que recriem de maneira precisa e didática os conceitos de um projeto de ambiente real. O projeto tem como aplicação atuar no controle e monitoramento remoto das instalações de iluminação, ventilação, alarmes acionados por sensor de presença e pelo sensor de fumaça, utilizando um aplicativo para telefones móveis para monitoramento e tem como programa base um servidor que pode ser acessado por todos os dispositivos conectados à rede de internet. Toma como base os conhecimentos adquiridos durante a graduação e os assuntos citados na revisão bibliográfica do Capítulo 2.

3.1 Levantamento de dados

O local escolhido para simular o projeto de automação foi um apartamento de pequeno porte, com dimensões de aproximadamente 50m², automatizando 3 cômodos: sala de estar, cozinha e quarto. A seguir, são citados os cômodos e os respectivos elementos a serem controlados: Na sala de estar estão presentes os seguintes itens controlados:

- Um ponto de iluminação;
- Sistema de monitoramento de temperatura.
- Sistema de ventilação;
- Sistema de monitoramento de presença direcionado para a porta de entrada, equipado com alarme sonoro;
- Motor elétrico para acionamento da persiana.

Na cozinha:

- Um ponto de iluminação;
- Sistema de detecção de gás e/ou fumaça, equipado com alarme sonoro.

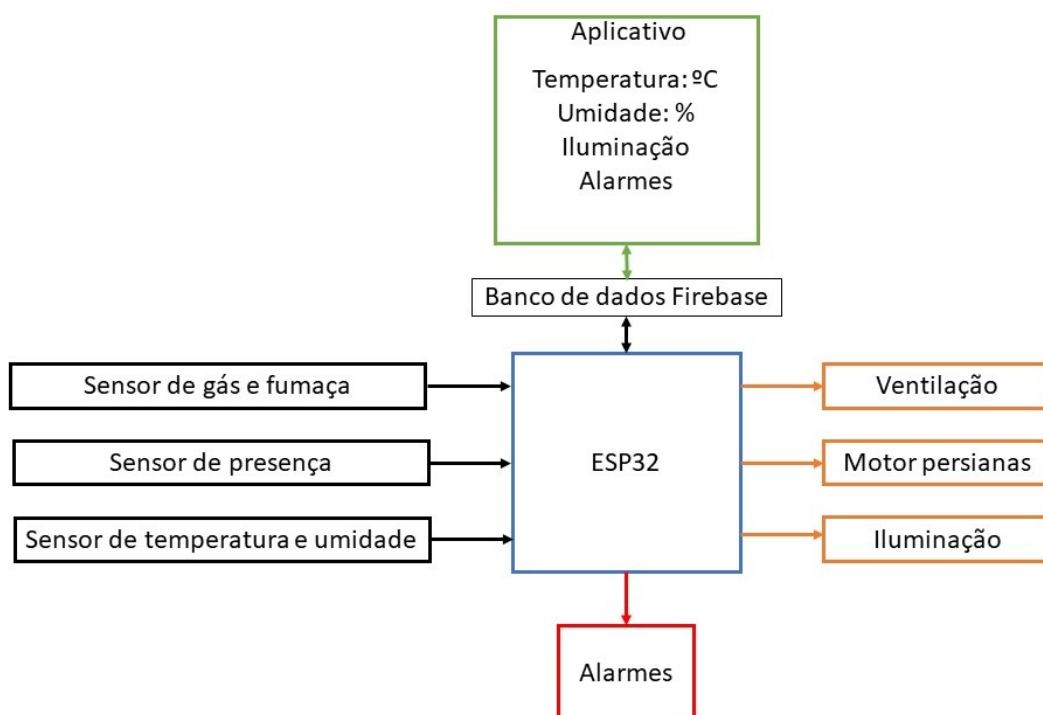
Quarto:

- Um ponto de iluminação.

3.2 Funcionamento do sistema

O sistema proposto tem como fundamento para sua implementação o diagrama de blocos mostrado na Figura 3. Nas seguintes seções, será explanado este diagrama, cuja análise irá detalhar os componentes de cada bloco.

Figura 3 – Diagrama de blocos do sistema completo



Fonte: Autoria própria 2023.

3.2.1 Ponto de iluminação e sistema de ventilação

Para realizar o projeto, não foram selecionados modelos de luminárias ou lâmpadas, tanto quanto sistema de ventilação, ficando a critério do usuário final definir qual modelo convém instalar no ambiente. Dessa maneira, para realizar o controle e monitoramento de cada dispositivo, basta enviar um sinal para o servidor, fazendo com que o microcontrolador reconheça a mudança de estado e, em seguida, atue o relé conectado diretamente com o circuito do ponto de iluminação. Da mesma forma, funciona o sistema de ventilação. Assim que o sistema reconhece uma mudança de estado no servidor, é enviado um sinal para o microcontrolador acionar o relé do circuito de ventilação. Para que o funcionamento não dependa apenas da interface do usuário, o projeto conta com botões, facilitando assim que o sistema de iluminação e/ou ventilação seja acionado por pessoas presentes no local que não tenham acesso ao aplicativo de controle e monitoramento. Para a realização deste trabalho foi necessário utilizar um módulo com quatro relés, o modelo selecionado foi o JQC-3FF-S-Z, ilustrado na Figura 4.

Em seguida é apresentado as principais características do módulo JQC-3FF-S-Z (TONG LING RELAYS,):

- Trabalha com tensão de operação de 5 VCC.
- Corrente nominal: 71,4 mA.
- Permite controlar cargas de até 250 VCA.
- Possui as pinagens: Normal Aberto (NA), Normal Fechado (NF) e Comum.
- Tensão máxima que as chaves suportam: (28 VCC a 10 A) ou (250 VCA a 10 A) ou (125 VCA a 15 A).
- Possui um tempo de resposta de 5 a 10 ms.

Figura 4 – Módulo de relés modelo JQC-3FF-S-Z



Fonte: Loja da Robotica 2023.

3.2.2 Acionamento da persiana

Para controlar a persiana de forma automatizada e evitar a entrada excessiva de luz solar, é comum a utilização de persianas controladas manualmente por meio de cordas. No entanto, para tornar o processo automatizado, considerou-se o uso de um motor de passo controlado pelo microcontrolador, conectado diretamente ao eixo giratório da persiana. Esse motor de passo abre ou fecha as abas da persiana quando acionado, permitindo o controle da entrada de luz externa.

Para a realização deste trabalho foi utilizado o motor de passo 28BYJ-48 e está apresentado na Figura 5.

Em seguida é apresentado as principais características do motor de passo 28BYJ-48 (TRONICS,):

- Tensão de trabalho 5 VDC.
- Ângulo do Passo: $5,625^\circ$ (num total de 64 passos para uma volta completa).

- Torque: 34,3 mN.m.

Figura 5 – Motor de passo 28BYJ-48



Fonte: FilipFlop 2022.

3.2.3 Sistema de monitoramento e alarmes

A instalação de alarmes em uma residência é uma maneira de direcionar a atenção para sistemas que podem funcionar de maneira incoerente com o esperado, causando sérias consequências caso medidas corretivas não sejam tomadas. Para abordar essa questão, o projeto foi desenvolvido com dois sensores de monitoramento. Um dos sensores utilizados no projeto é um sensor de presença, instalado direcionado à porta de entrada para detectar movimentos indesejados de pessoas, quando ativo. Para desenvolver o projeto foi usado o módulo PIR DYP-ME003, em seguida é apresentado suas principais características (OPENIMPULSE,):

- Tensão de operação: 4,5-20 Volts.
- Tensão de saída no pino de dados: 0 a 3.3 Volts.
- Distância detectável ajustável de 3 a 7 metros.
- Tempo de *delay* programável de 5 a 200 segundos.
- Temperatura de trabalho de -20 a +80 °C.

Na Figura 6 é mostrado módulo PIR DYP-ME003.

O outro sensor é um sensor de gás e fumaça, posicionado estrategicamente próximo ao fogão na cozinha, onde há uma fonte significativa de calor devido à utilização de gás em uma residência. Para o desenvolvimento do projeto foi selecionado o módulo sensor MQ-2. Em seguida é apresentado as principais características do MQ-2 (CO.,):

Figura 6 – Sensor de presença - módulo PIR DYP-ME003

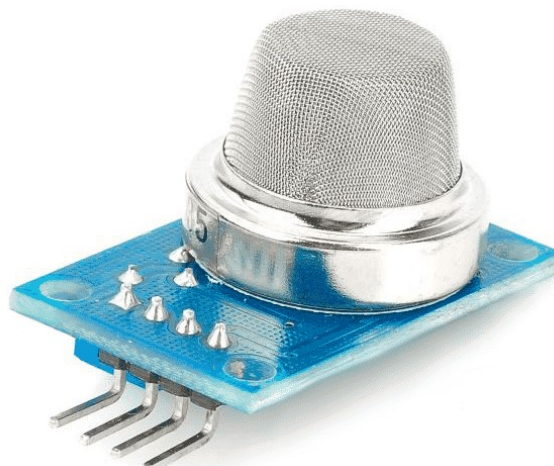


Fonte: FilipeFlop 2022.

- Detecta os principais gases: GLP, Metano, Propano, Butano, Hidrogênio, Gás Natural e também fumaça.
- Concentração de detecção: 200-5000 partes por milhão (ppm) dependendo do gás.
- Trabalha com tensão de operação de 5 Volts.
- Possui saída digital e analógica.

Na Figura 7 é mostrado um módulo do MQ-2.

Figura 7 – Sensor de gases - módulo MQ-2



Fonte: FilipeFlop 2022.

3.2.4 Monitoramento de temperatura

Garantir a temperatura adequada no ambiente residencial não apenas proporciona conforto individual, mas também é fundamental para proteger a saúde, preservar os objetos ma-

teriais e otimizar a utilização de energia. Tal prática pode resultar em um espaço mais saudável, agradável e economicamente vantajoso. Para o desenvolvimento do projeto foi selecionado o módulo sensor DHT-11. Em seguida é apresentado as principais características do DHT11 (MOUSER ELECTRONICS,):

- Faixa de detecção de temperatura: 0-50 °C
- Precisão de +/-2 °C
- Faixa de detecção de umidade: 20-90% UR
- Com precisão de +/-5% UR
- Tensão de alimentação: 3-5,5 Volts
- Com uma tensão de alimentação de 5 V e a 25 °C, o sensor terá um consumo de 0,1 mA em Standby e 0,5mA em utilização
- Possui período de amostragem de 1 segundo

Na Figura 8 é mostrado um módulo do DHT11.

Figura 8 – Sensor de temperatura e umidade - módulo DHT11



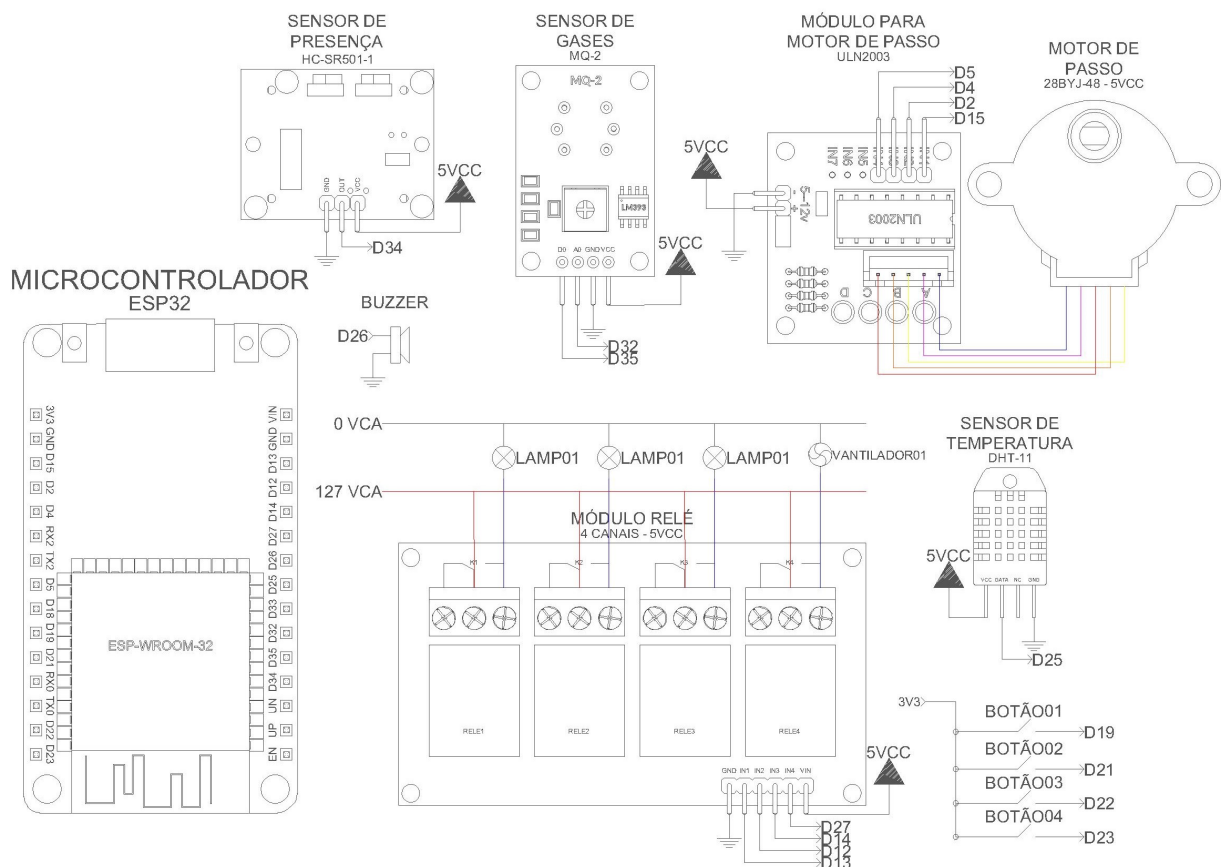
Fonte: FilipeFlop 2022.

3.3 Diagrama e componentes do projeto

Para facilitar a montagem de todos os componentes, foi necessário o uso de uma matriz de contatos devido à sua simplicidade na aplicação para fins experimentais, o que facilitou a verificação da funcionalidade do sistema desenvolvido, além de apresentar baixo custo e versatilidade. O ESP32 é um dispositivo completo para aplicação na automação, com módulos de transmissão sem fio integrados em sua estrutura. Oferece segurança e confiabilidade para o trabalho remoto, eliminando a necessidade de conexões com a internet por cabos. É importante mencionar também sua facilidade de programação, pois é compatível com várias linguagens

de programação, sendo o C++ escolhido como linguagem padrão para o desenvolvimento do programa. Devido às vantagens do módulo ESP32 em termos de programação e transmissão de dados, um banco de dados hospedado em nuvem foi utilizado para permitir a sincronização entre o aplicativo e o módulo ESP32, ambos podendo receber e enviar dados para o banco de dados. O equipamento selecionado para acionar os dispositivos atuadores integrados com o microcontrolador é o módulo de relé. Esse equipamento possibilita a transmissão dos dados enviados pelo ESP32, este chamado de parte de controle, para os equipamentos de tensão mais elevada esta chamada de parte de potência. O diagrama de conexões desenvolvido para o projeto está ilustrado na Figura 9.

Figura 9 – Diagrama de conexões



Fonte: Autoria própria 2023.

3.4 Banco de dados

O Firebase Realtime Database é um serviço de banco de dados em tempo real oferecido pelo Google como parte da plataforma Firebase. Ele permite que os desenvolvedores armazenem e sincronizem dados em tempo real na nuvem, facilitando o compartilhamento de

informações entre aplicativos e dispositivos em tempo real. O Firebase Realtime Database é particularmente útil para aplicativos de chat, jogos multiplayer e colaborativos, pois permite que aplicativos atualizem automaticamente quando os dados são modificados. Para incluir o banco de dados no projeto, é necessário criar variáveis que guardarão os dados, como apresentado na Figura 10.

Figura 10 – Variáveis no banco de dados

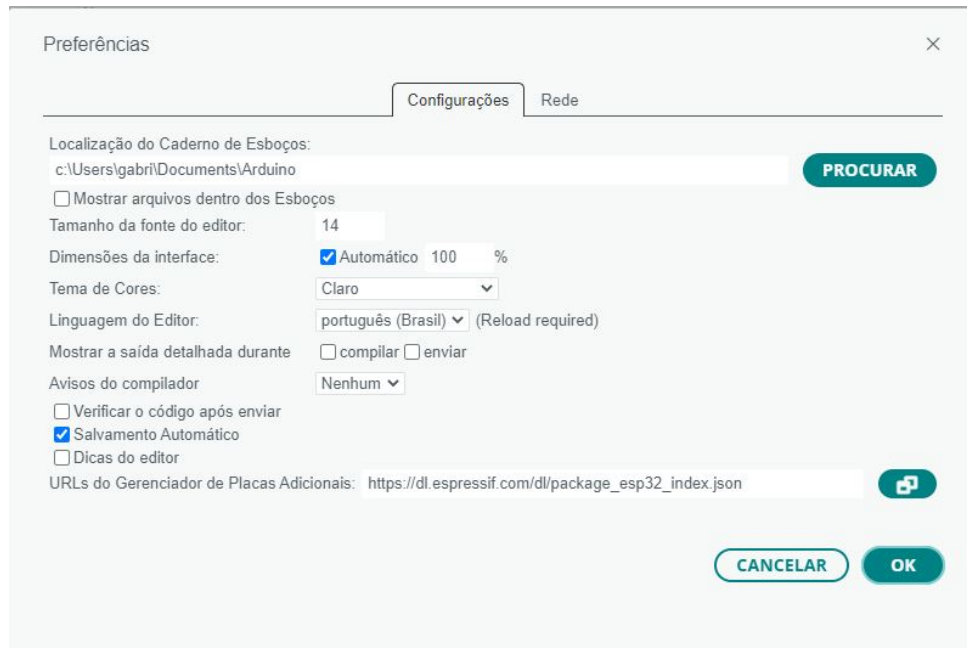


Fonte: Autoria própria 2023.

3.5 Programação do ESP32

O programa executado pelo ESP32 foi desenvolvido no software de programação Arduino IDE para possibilitar a conexão do software com o módulo do microcontrolador. Para que a IDE reconheça as informações das placas de programação do ESP32, é preciso alterar algumas configurações de preferências, consistindo em disponibilizar um endereço, como mostrado na Figura 11.

Figura 11 – Inclusão de endereço para adição do ESP32 no Arduino IDE



Fonte: Autoria própria 2023.

A programação do ESP32 é realizada usando comandos, diretivas e funções. No entanto, dependendo da aplicação, pode ser necessário adicionar algumas bibliotecas. Para a programação deste projeto, foram adicionadas as bibliotecas: *WiFi.h* para executar a conexão com a internet, *IOXhop_FirebaseESP32.h* e *ArduinoJson.h* para a conexão com o banco de dados, *DHT.h* para programar o sensor de temperatura e umidade, e *math.h* para realizar cálculos matemáticos. O código do programa inicia com uma série de definições, incluindo a criação de variáveis utilizadas na programação, criadas como variáveis globais. Na função *setup()*, as entradas e saídas do microcontrolador são definidas, juntamente com funções essenciais para a comunicação do ESP32 com o banco de dados, este trecho do código está presente no Anexo A.

A função *loop()* é um loop infinito que é executado continuamente após a inicialização do programa e reconhece as alterações no banco de dados referentes a comandos a serem executados, este trecho do código está presente no Anexo B.

Para lidar com vários sensores e atuadores de forma simultânea, na programação foi utilizada o método do *Free Real-Time Operating System* (FreeRTOS). O FreeRTOS é um sistema operacional de código aberto projetado para sistemas embarcados e de tempo real, oferecendo uma estrutura eficiente para gerenciar tarefas, temporizadores, semáforos, filas e outros recursos, este trecho do código está presente no Anexo C.

3.6 Interface com o usuário

A interface desempenha um papel fundamental ao viabilizar a interação, gerenciamento e supervisão de um sistema autônomo. Ela utiliza representações visuais para se comunicar com o usuário e fornece comandos intuitivos que podem ser facilmente operados por indivíduos autorizados. Para produzir a interface do usuário, foi utilizado o programa VSCODE. A linguagem de programação Dart, desenvolvida pela Google, foi utilizada para criar o aplicativo.

A arquitetura do aplicativo foi desenvolvida de forma básica, apresentando dados como temperatura e umidade do ambiente, juntamente com botões interativos responsáveis pelo controle e acionamento dos componentes do projeto, o *Layout* final do aplicativo é apresentado na Figura 12.

Na parte dos Cômodos ao acionar os botões "SALA", "COZINHA" e "QUARTO" tem-se como resposta do microcontrolador o acionamento ou desacionamento a iluminação do determinado ambiente, o botão "TUG" ativa uma tomada genérica, que, para fins do projeto, foi escolhida como um ventilador. Na parte de Alarmes, ao acionar os botões "PRESENÇA" e "GÁS/FUMAÇA", no projeto é acionado ou desacionado os alarmes. Na parte da Persiana o botão deslizante informa qual a posição das abas da persiana, ao deslizar para a direita a persiana abre e ao deslizar para a esquerda a persiana fecha, conforme posição estabelecida pelo cursor.

Figura 12 – Layout do aplicativo desenvolvido



Fonte: Autoria própria 2023.

3.7 Fluxograma

A Figura 13 ilustra o fluxograma que representa a programação do "meio físico", ou seja, a lógica de funcionamento dos componentes, enquanto Figura 14 representa a lógica de funcionamento do aplicativo.

3.8 Testes de funcionamento

Em seguida é apresentado imagens dos testes realizados para avaliar o funcionamento. O projeto foi dividido em duas partes para os testes, na primeira parte foi conectado ao ESP32 apenas o módulo relé e em seguida os outros sensores junto com o motor de passo.

3.8.1 Testes do modulo relé

Na Figura 15 é apresentado o circuito do módulo de relés junto ao ESP32 montados e interligados pela protoboard. O sistema funcionou como o esperado, respondendo aos estímulos feitos tanto pelo aplicativo, quanto pelos botões.

3.8.2 Testes dos sensores e motor de passo

Na Figura 16 é mostrado o circuito dos sensores e motor junto ao ESP32 montados e interligados pela protoboard.

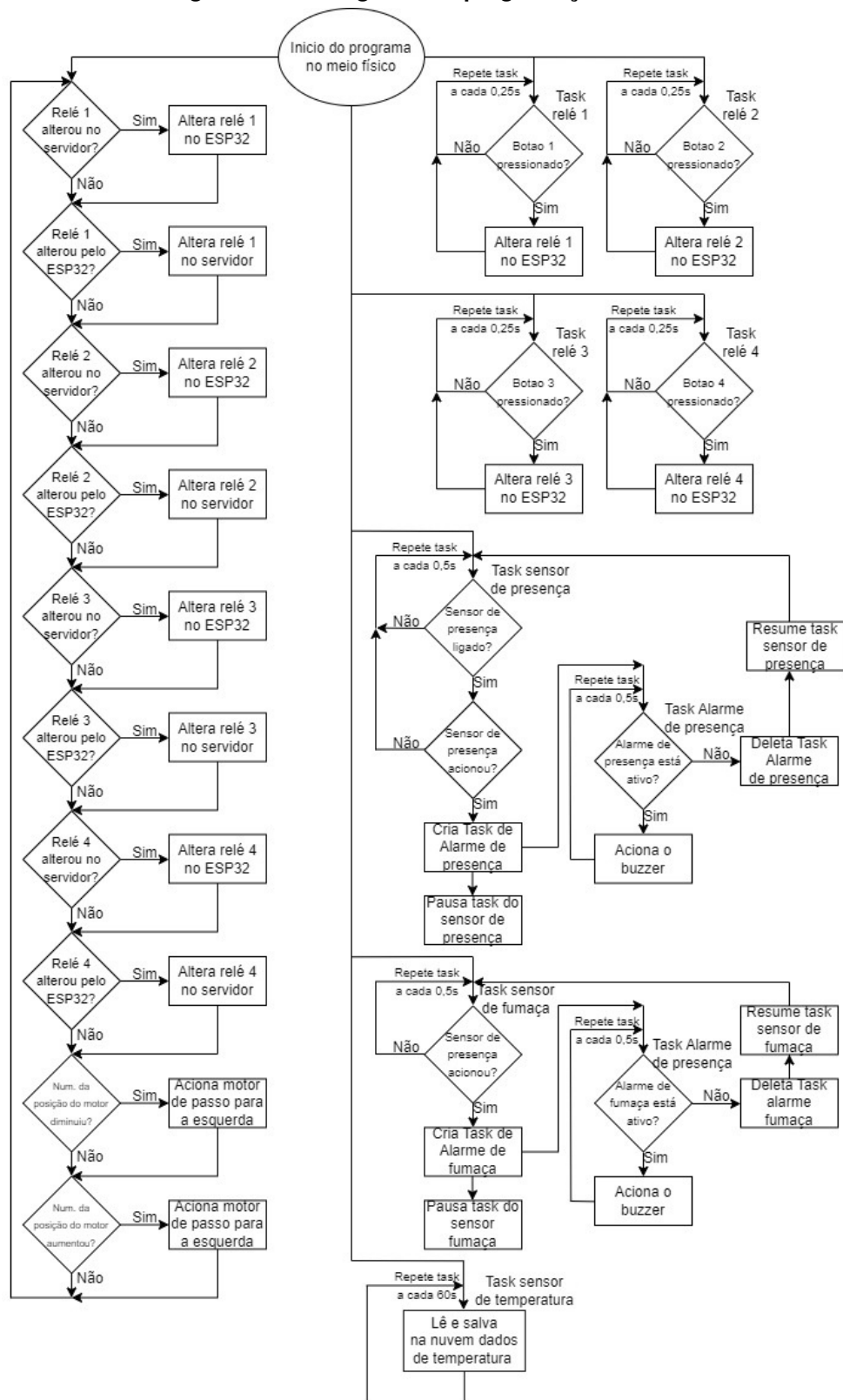
Para testar o sensor DHT11, bastou apenas forçar uma corrente de ar sobre o módulo do sensor mudando sua temperatura superficial e assim apresentando um bom funcionamento.

Para testar o funcionamento do sensor de gases e fumaça foi utilizado um potenciômetro, evitando assim ter que produzir fumaça no local dos testes. O MQ2 e o potenciômetro possuem uma forma semelhante de resposta, ambos apresentam como resposta um sinal analógico conforme o estímulo recebido, acionando um sinal luminoso quando reconhece um nível alto nos dados de entrada, sinal luminoso este adicionado para simular o sinal sonoro.

Foi testado também o sensor de presença apenas com movimentos de um objeto aproximando ou distanciando do foco do sensor, acionando também o sinal luminoso.

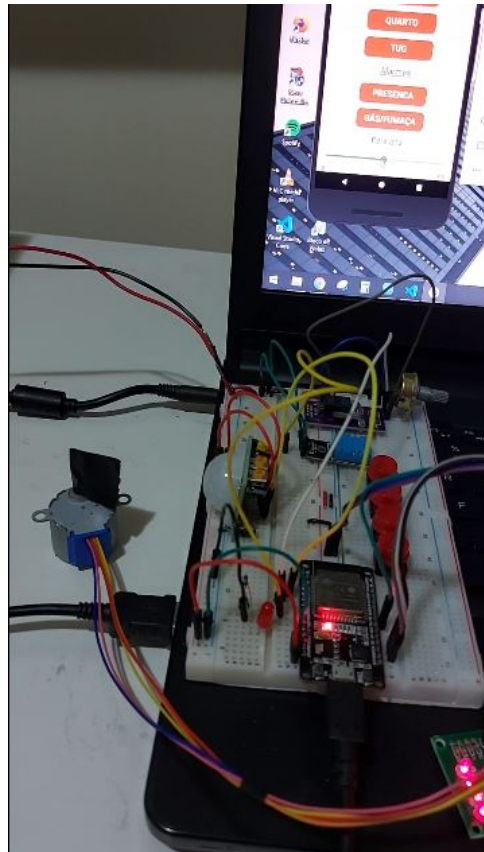
Para melhor visibilidade do movimento do motor de passo foi inserido uma fita isolante preta no eixo do motor, constatando o correto funcionamento após arrastar o botão deslizante para um lado e para o outro e observando o eixo realizar um movimento e depois o outro.

Figura 13 – Fluxograma da programação ESP32



Fonte: Autoria própria 2023.

Figura 16 – Testes com os sensores e motor de passo



Fonte: Autoria própria 2023.

afetou principalmente os testes de bancada, onde se constatou que as respostas do sistema tinham um intervalo de tempo considerável entre os estímulos do usuário e a reação do sistema, levando a considerar possíveis falhas na programação do microcontrolador.

4 RESULTADOS

Em seguida, os resultados serão apresentados em seções, dividindo os assuntos e apresentando comentários de cada parte do trabalho.

4.1 Protótipo

Todos os sensores e atuadores usados para controlar cada sistema funcionaram corretamente, sem oscilações abruptas nos testes, operando dentro dos valores esperados e respondendo rapidamente aos comandos enviados pelo microcontrolador ESP32.

Entretanto, devido a velocidade de conexão com a internet, dado apresentado na subseção 2.7.1, as respostas aos comandos apresentam um intervalo de tempo considerável entre o comando enviado pelo aplicativo e a resposta do microcontrolador.

4.2 Aplicativo

O desenvolvimento do aplicativo por meio da plataforma Flutter apresentou bons resultados no que diz respeito à funcionalidade da interface, a rapidez na execução dos comandos e as características que deixam o aplicativo intuitivo e com uma estética agradável ao usuário. A popularidade crescente desta plataforma, aliada à vasta quantidade de recursos disponíveis para integração, contribuiu significativamente para a rápida assimilação da ferramenta e da linguagem Dart durante o processo de aprendizado.

O aplicativo não possui sistema de autenticação e foi desenvolvido diretamente a uma conta Google pessoal, por este motivo, o aplicativo não deve ser distribuído nas lojas de aplicativo. Para o aplicativo ser reproduzido deverá haver o desenvolvimento de uma parte de login de usuário e autenticação de segurança.

4.3 Valores

Na Tabela 1 são apresentados todos os componentes e seus respectivos custos de mercado, bem como a quantidade deles utilizados no projeto.

Como podemos analisar na Tabela 1 para montar o projeto com o ESP32 e outros sensores, o motor de passo, os módulos de relé, os botões e dispositivos de teste, o gasto total foi de R\$ 237,73. Este conjunto forma um sistema completo que pode ser empregado diretamente na implementação de um sistema de automação residencial, automatizando os principais acionamentos elétricos no ambiente.

Tabela 1 – Apresentação de valores e quantidade dos componentes utilizados.

Componente	Qtd	Valor unitário	Total
Microcontrolador ESP32	1	R\$ 22,94	R\$ 22,94
Cabos jumper	40	R\$ 0,40	R\$ 16,00
Matriz de contatos	2	R\$ 18,99	R\$ 37,98
Botões	4	R\$ 1,29	R\$ 5,16
Sensor de presença PIR HC-SR501	1	R\$ 19,40	R\$ 19,40
Sensor de Fumaça MQ-02	1	R\$ 15,00	R\$ 15,00
Sensor de temperatura DHT11	1	R\$ 11,99	R\$ 11,99
Sensor de corrente ACS758	1	R\$ 29,42	R\$ 29,42
Módulo 4 relés	1	R\$ 19,84	R\$ 19,84
Motor de passo 28BYJ-48 5V com Drive ULN3002	1	R\$ 20	R\$ 20
Fonte RTX Fx 450 Lpj2-20 24p 200w	1	R\$ 40	R\$ 40
TOTAL			R\$ 237,73

Fonte: Autoria própria..

5 CONCLUSÃO

A automação residencial é pouco difundida por causa dos custos elevados associados à sua instalação. Assim, estabeleceram-se metas para alcançá-las ao criar um modelo que simula um sistema residencial automatizado e eficiente. O objetivo é melhorar a acessibilidade do ambiente, abrangendo aspectos relacionados à comodidade e economia de energia, com base na gestão e monitoramento remotos dos principais dispositivos elétricos em uma casa.

A fim de atingir tais características, foi essencial fazer uso de um microcontrolador confiável e extremamente versátil, atributos encontrados no ESP32, graças ao seu excelente poder de processamento e à inclusão de um módulo Wi-Fi em sua estrutura. Isso se mostra como uma opção economicamente viável, diminuindo a necessidade de cabos e espaço para conexões físicas.

Analisando os elementos mencionados no presente projeto, chegamos à conclusão de que todos os objetivos propostos foram cumpridos. O resultado obtido é um protótipo de baixo custo que, do ponto de vista funcional, tem a capacidade de controlar e supervisionar os principais dispositivos elétricos de uma residência. No entanto, para a implementação prática e comercial, faz-se necessário realizar uma análise detalhada dos custos de desenvolvimento do software, além de encontrar um fornecedor que ofereça o circuito como produto final, pronto para uso. Além disso, em uma escala maior, é importante buscar uma redução maior nos custos dos componentes. Essas são sugestões que devem ser levadas em consideração para futuros projetos.

REFERÊNCIAS

- AFONSO ROBERTO BENEDITO DE O. PEREIRA, M. F. L. P. B. S. Utilização da internet das coisas para o desenvolvimento de miniestação de baixo custo para monitoramento de condições do tempo em áreas agrícolas. **Anais ERI-MT 2015**, Nov 2015.
- AURESIDE. **Previsões para o mercado global de Aut. Residencial**. 2016. Disponível em: <http://www.aureside.org.br/noticias/previsoes-para-o-mercado-global-de-aut.-residencial>. Acesso em: 08 nov. 2023.
- BELMONTE, F. **Automação Residencial Preço: Quanto custa uma casa Automatizada?** 2022. Acatar Automação Residencial. Disponível em: <https://acatarautomacao.com.br/blog/automacao-residencial-preco/>. Acesso em: 08 nov. 2023.
- BOLZANI, C. A. M. Desmitificando a domótica. **Sinergia**, v. 8, n. 1, p. 17–20, jan. 2007.
- BOLZANI Caio Augustus Moraes. **Desenvolvimento de um Simulador de Controle de Dispositivos Residenciais Inteligentes: Uma Introdução aos Sistemas Domóticos**. 2004. 130 p. Dissertação (Mestrado) — Escola Politécnica da Universidade de São Paulo, São Paulo, 2004.
- BORGES ANTÔNIO PADILHA LANARI BO, A. S. M. L. C. C. M. F. G. F. E. D. B. A. C. C. F. G. A. **Desenvolvimento com microcontroladores Atmel AVR**. 2008. 38 p.
- BRANDÃO, R. A domotica ao serviço da sociedade. **NEUTRO à TERRA**, set. 2008.
- BRASIL, G. Estudo de Automação do Mercado Brasileiro: Os impactos da automação entre empresas e consumidores no Brasil. **Associação Brasileira de Energia**, mar. 2017.
- CHATTERJEE SOUVIK CHAKRABORTY, A. D. D. A. N. N. Real-time communication application based on android using google firebase. **International Journal of Advance Research in Computer Science and Management Studies**, v. 6, n. 4, abr. 2018.
- CO., L. H. E. **TECHNICAL DATA MQ-2 GAS SENSOR**. [S./].
- DEITEL, H. M. **C++: Como Programar**. [S./]: ARTMED EDITORA LTDA, 2006.
- DEITEL, H. M. D. P. J. **Como programar Em C**. [S./]: LTC Editora, 1999.
- FLUTTER. Disponível em: <https://flutter.dev/>. Acesso em: 08 nov. 2023.
- FUENTES, C. R. d. N. R. C. **Eletrônica**. [S./]: Colégio Técnico Industrial, 2013.
- GOTARDO, R. A. **Linguagem de Programação**. [S./]: Fabrica de Conhecimento, 2015.
- KELLY, G. Home automation: past, present future. **Eletronics Australia**, p. 76–79, fev. 1997.
- MACHESO, D. T. P. S. Esp32 based electric energy consumption meter. **International Journal of Computer Communion and Informatics**, maio 2022.
- MAINARDI S. BANZI, m. B. S. B. E. A low-cost home automation system based on power-line communications links. **International Symposium on Automation and Robotics in Construction**, set. 2005.
- MORAES, A. F. de. **Redes sem fio: Instalação, configuração e segurança**. [S./]: Saraiva, 2010.
- MOUSER ELECTRONICS. **DHT11 Humidity Temperature Sensor datasheet**. [S./].

MURATORI, P. H. D. B. J. R. Automação residencial: historico, definições e conceitos. **O Setor Eltrico**, v. 62, 2011.

OGATA Katsuhiko. **Engenharia de Controle Moderno**. [S./]: Pearson, 2013.

OPENIMPULSE. **DYP-ME003 PIR Sensor Module Datasheet**. [S./].

PATANE, E. J. **Implementação de controle de velocidade em malha fechada para motores de corrente continua utilizando sistema de aquisição de dados**. 2008. Dissertação (Mestrado) — Escola de Engenharia Maua do Centro Universitario do Instituto Maua, 2008.

RIBEIRO, M. A. **Automação Industrial**. Bahia, BR, 2001. 498 p.

SANTOS JuLIO CeZAR COÊLHO BARBOSA TORQUATO, V. R. S. S. E. S. d. S. A. F. D. V. R. S. Desenvolvimento de um sistema de monitoramento e controle de condicionadores de ar a baixo custo. **Encontro internacional de jovens investigadores**, out. 2017.

SANTOS LUCAS A. M. SILVA, C. S. F. S. C. J. B. B. N. B. S. P. M. A. M. V. L. F. M. V. O. N. G. e. A. A. F. L. B. P. Internet das coisas: da teoria à pratica. **Departamento de Cincia da Computação Universidade Federal de Minas Gerais**, jan. 2016.

SCHULER, C. A. **Eletrônica I: Habilidades basicas em eletricidade, eletrônica e telecomunicações**. [S./]: AMGH EDITORA LTDA, 2013.

SILEVIRA, W. Q. L. L. Um breve histórico conceitual da Automação Industrial e Redes para Automação Industrial. **UFRN – Univesidade Federal do Rio Grande do Norte**, maio 2003.

SILVA, M. G. M. Measurement of temperature and humidity by using arduino tool and dht11. **Universidade Federal do Rio de Janeiro**, fev. 2012.

SRIVASTAVA AWANISH KESARWANI, S. D. D. Avaliação de desempenho de reles de proteção digitais. **Academy of Technical Education - Noida - India**, dez. 2018.

TONG LING RELAYS. **JQC 3FF Product Espcification**. [S./].

TRONICS, K. **28BYJ-48 – 5V Stepper Motor**. [S./].

VISUAL Studio Code. Disponível em: <https://code.visualstudio.com/>. Acesso em: 08 nov. 2023.

WENDDLING, M. Sensores. **Academy of Technical Education - Noida - India**, 2010.

WOLFF, B. Aplicações para a web moderna com dart. **Centro Universitario Facvest**, nov. 2012.

WORTMEYER FERNANDO FREITAS, L. C. C. Automação residencial: Busca de tecnologias visando o conforto, a economia, a praticidade e a segurança do usuário. **Associação Educacional Dom Bosco – AEDB**, 2005.

ANEXO A –

Listing A.1 – Código para chamada de bibliotecas, declaração de variáveis globais e configuração de entradas e saídas digitais ou analógicas.

```

1      //Biblioteca para conexao wifi
2      #include <WiFi.h>
3      //Biblioteca para conexao com o firebase
4      #include <IOXhop_FirebaseESP32.h>
5      //Biblioteca para manipular servidor web
6      #include <ArduinoJson.h>
7      //Biblioteca do sensor DHT11
8      #include <DHT.h>
9      //Biblioteca para funções matematicas
10     #include <math.h>
11
12     //Login e senha wifi
13     #define WIFI_SSID "#####" //ID da rede WiFi
14     #define WIFI_PASSWORD "#####" //Senha da rede WiFi
15
16     //Login e senha firebase
17     #define FIREBASE_HOST "#####"
18     #define FIREBASE_AUTH "#####"
19
20     #define DHTPIN 25 //Porta logica Dht11
21     #define DHTTYPE DHT11 //Modelo do sensor DHT
22     //-----Inicializacao do sensor DHT-----
23     DHT dht(DHTPIN, DHTTYPE);
24
25     bool AlterarRele1Firebase = false , rele1 = false ;
26     bool AlterarRele2Firebase = false , rele2 = false ;
27     bool AlterarRele3Firebase = false , rele3 = false ;
28     bool AlterarRele4Firebase = false , rele4 = false ;
29     bool flagSensorMov = false ;
30     bool flagAlarmeSonoro = false ;
31     bool sinalSensorPresenca = false ;
32     bool alarmeSonoro = false ;
33

```

```

34     int Dig13 = 13;
35     int Dig12 = 12;
36     int Dig14 = 14;
37     int Dig27 = 27;
38     int tecla1 = 19;
39     int tecla2 = 21;
40     int tecla3 = 22;
41     int tecla4 = 23;
42     int senMov = 34;
43     int senFumA = 35;
44     int senFumD = 32;
45     int buzzer = 26;
46     int pinAdcCorrente = 33;
47     int passoPin_1 = 15;
48     int passoPin_2 = 12;
49     int passoPin_3 = 4;
50     int passoPin_4 = 5;
51     int setNumeroPassoMotor = 0;
52     int posicaoMotor = 0;
53
54     //declaracao dos handles para tasks
55     static TaskHandle_t sensorPresenca = NULL;
56     static TaskHandle_t alarmePresenca = NULL;
57     static TaskHandle_t sensorMQ02 = NULL;
58     static TaskHandle_t alarmeMQ02 = NULL;
59
60     void setup() {
61         Serial.begin(115200);
62         ConectarWifi(); //Conectando o Wifi
63         ConectarFirebase(); //Conectando o Firebase
64
65         dht.begin(); //Inicializacao do sensor DHT
66
67         //Declaracao de entradas

```

```

68     pinMode( tecla1 ,INPUT_PULLDOWN);
69     pinMode( tecla2 ,INPUT_PULLDOWN);
70     pinMode( tecla3 ,INPUT_PULLDOWN);
71     pinMode( tecla4 ,INPUT_PULLDOWN);
72     pinMode(senMov, INPUT);
73     pinMode(senFumA, INPUT);
74     pinMode(senFumD, INPUT);
75     pinMode(pinAdcCorrente , INPUT);
76     pinMode( passoPin_1 ,OUTPUT);
77     pinMode( passoPin_2 ,OUTPUT);
78     pinMode( passoPin_3 ,OUTPUT);
79     pinMode( passoPin_4 ,OUTPUT);
80
81     //Declaracao de saidas
82     pinMode( Dig13 , OUTPUT);
83     pinMode( Dig12 , OUTPUT);
84     pinMode( Dig14 , OUTPUT);
85     pinMode( Dig27 , OUTPUT);
86     pinMode( buzzer , OUTPUT);
87
88      //(Funcao a ser chamada, Nome da tarefa , Tamanho (bytes), Parametro a
89      ser passado, Prioridade da Tarefa , Task handle)
90     xTaskCreate(taskPressBotao1 , "Botao 1 pressionado", 1000, NULL, 1, NULL);
91     xTaskCreate(taskPressBotao2 , "Botao 2 pressionado", 1000, NULL, 1, NULL);
92     xTaskCreate(taskPressBotao3 , "Botao 3 pressionado", 1000, NULL, 1, NULL);
93     xTaskCreate(taskPressBotao4 , "Botao 4 pressionado", 1000, NULL, 1, NULL);
94     xTaskCreate(tasksensorPresenca , "Sensor de movimento", 5000,
95     NULL, 1, &sensorPresenca);
96     xTaskCreate(taskSensorDHT , "Sensor de Temp e Hum", 5000,
97     NULL, 1, NULL );
98     xTaskCreate(taskSensorMQ02 , "Sensor de Fumaça", 5000,
99     NULL, 1, &sensorMQ02 );
100    xTaskCreate(taskSensorCorrente , "Sensor de Corrente", 5000,
101    NULL, 1, NULL );

```

102	}
-----	---

ANEXO B –

Listing B.1 – Código da rotina *Loop()*

```

1 void loop() {
2     //Recolhe as informacoes do Firebase
3     bool novoEstadoRele1 = Firebase.getBool("/CASA/SALA");
4     bool novoEstadoRele2 = Firebase.getBool("/CASA/COZINHA");
5     bool novoEstadoRele3 = Firebase.getBool("/CASA/QUARTO");
6     bool novoEstadoRele4 = Firebase.getBool("/CASA/TUG");
7     bool sinalSensorPresenca = Firebase.getBool("/CASA/PRESENA");
8     bool alarmeSonoro = Firebase.getBool("/CASA/ALARME_SONORO");
9     bool flagAlarmeMQ02 = Firebase.getBool("/CASA/GAS_FUMACA");
10    int setNumeroPassoMotor = Firebase.getInt("/CASA/PERSIANA");
11    delay(80);
12
13    //Testa se houve alteracao do estado do rele 1 na nuvem
14    if(novoEstadoRele1 != rele1){
15        rele1 = novoEstadoRele1;
16        AlterarRele1Firebase= rele1;
17        digitalWrite(Dig13, rele1 ? LOW : HIGH );
18    }
19    //Testa se houve clique no botao 1
20    if (AlterarRele1Firebase != rele1) {
21        rele1 = AlterarRele1Firebase;
22        Firebase.setBool("/CASA/LUZ",rele1);
23    }
24    //Testa se houve alteração do estado do rele 2 na nuvem
25    if (novoEstadoRele2 != rele2) {
26        rele2 = novoEstadoRele2;
27        AlterarRele2Firebase= rele2;
28        digitalWrite(Dig12, rele2 ? LOW : HIGH );
29    }
30    //Testa se houve clique no botao 2
31    if (AlterarRele2Firebase != rele2) {
32        rele2 = AlterarRele2Firebase;
33        Firebase.setBool("/CASA/LUZ2",rele2);

```

```

34     }
35     //Testa se houve alteracao do estado do rele 3 na nuvem
36     if (novoEstadoRele3 != rele3) {
37         rele3 = novoEstadoRele3;
38         AlterarRele3Firebase= rele3;
39         digitalWrite(Dig14, rele3 ? LOW : HIGH );
40     }
41     //Testa se houve clique no botao 3
42     if (AlterarRele3Firebase != rele3) {
43         rele3 = AlterarRele3Firebase;
44         Firebase.setBool("/CASA/LUZ3",rele3);
45     }
46     //Testa se houve alteracao do estado do rele 4 na nuvem
47     if (novoEstadoRele4 != rele4) {
48         rele4 = novoEstadoRele4;
49         AlterarRele4Firebase= rele4;
50         digitalWrite(Dig27, rele4 ? LOW : HIGH );
51     }
52     //Testa se houve clique no botao 4
53     if (AlterarRele3Firebase != rele4) {
54         rele4 = AlterarRele4Firebase;
55         Firebase.setBool("/CASA/TUG",rele3);
56     }
57     //testa se a posicao do motor foi alterada no servidor
58     if(posicaoMotor > setNumeroPassoMotor){
59         giraEsquerda(posicaoMotor – setNumeroPassoMotor);
60         posicaoMotor = setNumeroPassoMotor;
61     }
62     if(posicaoMotor < setNumeroPassoMotor){
63         giraDireita(setNumeroPassoMotor – posicaoMotor);
64         posicaoMotor = setNumeroPassoMotor;
65     }
66 }

```

ANEXO C –

Listing C.1 – Código das Tasks FreeRTOS

```

1      //Task para reconhecer clique no botao 1
2      void taskPressBotao1(void* parameter){
3          for(;;){ // loop infinito
4              if(digitalRead(tecla1)==HIGH){
5                  AlterarRele1Firebase=!AlterarRele1Firebase;
6                  digitalWrite(Dig13, AlterarRele1Firebase ? LOW : HIGH );
7              }
8              vTaskDelay(250 / portTICK_PERIOD_MS);
9          }
10     }
11
12     //Task para reconhecer clique no botao 2
13     void taskPressBotao2(void* parameter){
14         for(;;){ // loop infinito
15             if(digitalRead(tecla2)==HIGH){
16                 AlterarRele2Firebase =! AlterarRele2Firebase;
17                 digitalWrite(Dig12, AlterarRele2Firebase ? LOW : HIGH );
18             }
19             vTaskDelay(250 / portTICK_PERIOD_MS);
20         }
21     }
22
23     //Task para reconhecer clique no botao 3
24     void pressBotao3(void* parameter){
25         for(;;){ // loop infinito
26             if(digitalRead(tecla3)==HIGH){
27                 AlterarRele3Firebase =! AlterarRele3Firebase;
28                 digitalWrite(Dig14, AlterarRele3Firebase ? LOW : HIGH );
29             }
30             vTaskDelay(250 / portTICK_PERIOD_MS);
31         }
32     }
33

```

```

34      //Task para reconhecer clique no botao 4
35      void pressBotao4(void* parameter){
36          for(;;){ // loop infinito
37              if(digitalRead(tecla4)==HIGH){
38                  AlterarRele4Firebase =! AlterarRele4Firebase;
39                  digitalWrite(Dig27, AlterarRele4Firebase ? LOW : HIGH );
40              }
41              vTaskDelay(250 / portTICK_PERIOD_MS);
42          }
43      }
44
45      //Task para reconhecer sensor de movimento
46      void tasksensorPresenca(void* parameter){
47          for(;;){ // loop infinito
48              if(sinalSensorPresenca && !flagSensorMov){
49                  //Se no servidor for ativado o sensor de presenca e a flag
50                  //estiver baixa, ativa a task do sensor
51                  if(digitalRead(senMov)==HIGH){
52                      flagSensorMov=!flagSensorMov;
53                      flagAlarmeSonoro = !flagAlarmeSonoro;
54                      Firebase.setBool("/CASA/ alarmeSonoro",flagAlarmeSonoro);
55                      xTaskCreate(taskalarmePresenca , "Alarme presenca", 1000, NULL,
56                          1, &alarmePresenca);
57                      vTaskSuspend(sensorPresenca);
58                  }
59              }
60              vTaskDelay(500 / portTICK_PERIOD_MS);
61          }
62      }
63
64      //Task para alarme do sensor de presenca
65      void taskalarmePresenca(void* parameter){
66          for(;;){ // loop infinito
67              if(flagAlarmeSonoro && alarmeSonoro){

```

```

68     Serial.println("Buzzer");
69     digitalWrite(buzzer, HIGH);
70     delay(500);
71     digitalWrite(buzzer, LOW);
72 }else{
73     Firebase.setBool("/CASA/alarmeSonoro",false);
74     alarmeSonoro=false;
75     vTaskDelete(alarmePresenca);
76 }
77 vTaskDelay(500 / portTICK_PERIOD_MS);
78 }
79 }
80
81 //Task para leitura do sensor DHT11
82 void taskSensorDHT(void* parameter){
83     for(;;){ // loop infinito
84         float t = dht.readTemperature(); //Lendo Temperatura
85         float h = dht.readHumidity(); //Lendo umidade relativa
86
87         Firebase.setFloat("/CASA/TEMP",t); //Seta temperatura no firebase
88         Firebase.setFloat("/CASA/HUMI",h); //Seta humidade no firebase
89
90         vTaskDelay(60000 / portTICK_PERIOD_MS);
91     }
92 }
93
94 //Task para leitura do sensor MQ02
95 void taskSensorMQ02(void* parameter){
96     for(;;){ // loop infinito
97         int sensorLimite = 600;
98         if(analogRead(senFumA) > sensorLimite){
99             Firebase.setBool("/CASA/alarmeFumaca",true);
100             xTaskCreate(taskAlarmeMQ02, "Alarme Sensor de Fumaca", 5000,
101             NULL, 1, &alarmeMQ02);

```

```

102         vTaskSuspend(sensorMQ02);
103     }
104     vTaskDelay(1000 / portTICK_PERIOD_MS);
105 }
106 }
107
108 //Task para alarme do sensor MQ02
109 void taskAlarmeMQ02(void* parameter){
110     for(;;){ // loop infinito
111         if(Firebase.getBool("/CASA/alarmeFumaca")){
112             digitalWrite(buzzer, HIGH);
113             delay(500);
114             digitalWrite(buzzer, LOW);
115         }
116         else{
117             Firebase.setBool("/CASA/alarmeFumaca", false);
118             vTaskResume(sensorMQ02);
119             vTaskDelete(alarmeMQ02);
120         }
121         vTaskDelay(500 / portTICK_PERIOD_MS);
122     }
123 }
124
125 //Função gira motor pra esquerda
126 void giraEsquerda(int Voltas){
127     int step_number = 0;
128     for(int i=0;i<=Voltas;i++){
129         Serial.println(i);
130         switch(step_number){
131             case 0:
132                 digitalWrite(passoPin_1, HIGH);
133                 digitalWrite(passoPin_2, LOW);
134                 digitalWrite(passoPin_3, LOW);
135                 digitalWrite(passoPin_4, LOW);

```

```

136         break;
137     case 1:
138         digitalWrite (passoPin_1 ,LOW);
139         digitalWrite (passoPin_2 ,HIGH);
140         digitalWrite (passoPin_3 ,LOW);
141         digitalWrite (passoPin_4 ,LOW);
142     break;
143     case 2:
144         digitalWrite (passoPin_1 ,LOW);
145         digitalWrite (passoPin_2 ,LOW);
146         digitalWrite (passoPin_3 ,HIGH);
147         digitalWrite (passoPin_4 ,LOW);
148     break;
149     case 3:
150         digitalWrite (passoPin_1 ,LOW);
151         digitalWrite (passoPin_2 ,LOW);
152         digitalWrite (passoPin_3 ,LOW);
153         digitalWrite (passoPin_4 ,HIGH);
154     break;
155 }
156 step_number++;
157 if (step_number>3){
158     step_number=0;
159 }
160 delay (2);
161 }
162 }
163
164 //Função gira motor pra direita
165 void giraDireita(int Voltas){
166     int step_number = 0;
167     for(int i=0;i<=Voltas;i++){
168         Serial.println(i);
169         switch(step_number){

```

```
170         case 0:
171             digitalWrite (passoPin_1 ,LOW);
172             digitalWrite (passoPin_2 ,LOW);
173             digitalWrite (passoPin_3 ,LOW);
174             digitalWrite (passoPin_4 ,HIGH);
175         break ;
176         case 1:
177             digitalWrite (passoPin_1 ,LOW);
178             digitalWrite (passoPin_2 ,LOW);
179             digitalWrite (passoPin_3 ,HIGH);
180             digitalWrite (passoPin_4 ,LOW);
181         break ;
182         case 2:
183             digitalWrite (passoPin_1 ,LOW);
184             digitalWrite (passoPin_2 ,HIGH);
185             digitalWrite (passoPin_3 ,LOW);
186             digitalWrite (passoPin_4 ,LOW);
187         break ;
188         case 3:
189             digitalWrite (passoPin_1 ,HIGH);
190             digitalWrite (passoPin_2 ,LOW);
191             digitalWrite (passoPin_3 ,LOW);
192             digitalWrite (passoPin_4 ,LOW);
193         break ;
194     }
195     step_number++;
196     if ( step_number>3){
197         step_number=0;
198     }
199     delay (2);
200 }
201 }
```