

Controle Residencial com Node-RED e ESP32

Hericles Henriqueson Tavares Ramos - 42542

Trabalho realizado sob a orientação de

Prof. Gustavo Silva Funchal

Prof. Paulo Jorge Pinto Leitão

Licenciatura em Engenharia Eletrotécnica e de Computadores

2022-2023

Controle Residencial com Node-RED e ESP32

Relatório da UC de Projeto

Licenciatura em Engenharia Eletrotécnica e de Computadores

Escola Superior de Tecnologia e Gestão

Hericles Henriqueson Tavares Ramos - 42542

2022-2023

A Escola Superior de Tecnologia e de Gestão não se responsabiliza pelas opiniões expressas neste relatório.

Declaro que o trabalho descrito neste relatório é da minha autoria e é da minha vontade que o mesmo seja submetido a avaliação.

Hericles Henriqueson Tavares Ramos - 42542

Agradecimentos

Gostaria de expressar minha gratidão ao meu orientador Gustavo Silva Funchal por sua orientação, apoio e conhecimento ao longo deste processo. Suas sugestões foram inestimáveis para o desenvolvimento deste trabalho, e estou muito grato por sua orientação.

Resumo

O projeto em questão, centrado na automação residencial impulsionada pela Internet das Coisas (IoT), visa criar um ambiente residencial mais inteligente, onde a análise, controle e integração de diversos ambientes são fundamentais. O foco principal é aprimorar a qualidade de vida dos moradores, oferecendo conforto, segurança e eficiência energética.

Ao escolher a automação residencial como aplicação da IoT, o projeto busca facilitar as tarefas diárias através da integração de ferramentas de código aberto, como o protocolo MQTT e o Mosquitto Broker, a comunicação entre dispositivos se torna eficiente e flexível.

O microcontrolador ESP32 foi escolhido como um dos clientes neste sistema, devido à sua versatilidade e eficiência energética. Ele desempenha um papel crucial na coleta e transmissão de dados relevantes para a automação residencial, contribuindo para a conectividade inteligente do ambiente.

Além disso, a ferramenta Node-RED foi adotada para estabelecer a comunicação entre o broker e criar uma interface intuitiva para o usuário. Isso permite que os moradores controlem e monitorem diversos aspectos da casa de forma acessível e personalizada.

Em resumo, o projeto não apenas adota tecnologias de ponta, como o ESP32, Node-RED e MQTT, mas também visa democratizar a automação residencial, tornando-a acessível e eficiente para um público mais amplo. Ao integrar essas tecnologias, o projeto busca proporcionar uma experiência mais inteligente, conectada e personalizada para os moradores, representando um passo em direção ao futuro das residências inteligentes.

Palavras-chave: Casa Inteligente, ESP32, Node-RED, MQTT.

Abstract

The project at hand, centered on home automation driven by the Internet of Things (IoT), aims to create a smarter residential environment where the analysis, control, and integration of various elements are paramount. The primary focus is to enhance the quality of life for residents by providing comfort, security, and energy efficiency. By choosing home automation as an application of IoT, the project seeks to streamline daily tasks through the integration of open-source tools, such as the MQTT protocol and the Mosquitto Broker, facilitating efficient and flexible communication between devices.

The ESP32 microcontroller was selected as one of the clients in this system due to its versatility and energy efficiency. It plays a crucial role in collecting and transmitting relevant data for home automation, contributing to the intelligent connectivity of the environment. Additionally, the Node-RED tool was adopted to establish communication between the broker and create an intuitive interface for the user. This enables residents to control and monitor various aspects of the home in an accessible and personalized manner.

In summary, the project not only embraces cutting-edge technologies like ESP32, Node-RED, and MQTT but also aims to democratize home automation, making it accessible and efficient for a broader audience. By integrating these technologies, the project seeks to provide a smarter, connected, and personalized experience for residents, representing a step toward the future of smart homes.

Keywords: Smart home, ESP32, Node-RED, MQTT..

Conteúdo

1	Introdução	1
1.1	Automação Residencial: Definição e Evolução	1
1.2	Objetivos	2
1.3	Organização do documento	3
2	Tecnologias e Ferramentas Utilizadas	5
2.1	MQTT (Message Queuing Telemetry Transport)	5
2.1.1	QoS (Quality of Service)	6
2.2	Node-RED	7
2.3	ESP32	8
2.3.1	Diferenças entre os microcontroladores	9
2.4	Sensor de temperatura e Humidade DHT11	9
2.4.1	Diferenças entre os sensores DHT11 e DHT22	10
2.5	Sensor de distância ultrasónico HC-SR04	10
2.6	Sensor de intensidade de luz BH1750	11
2.7	Servo motor SG90 9g Tower Pro	12
2.8	Arduino IDE	12
3	Desenvolvimento	15
3.1	Implementação	15
3.1.1	Arquitetura do projeto	16
3.1.2	Conexão entre o ESP32 ao wifi	17

3.1.3	Conexão entre o MQTT e o node-RED	18
3.1.4	Inscrições e Publicações	18
3.1.5	Aplicação dos sensores	19
4	Resultados	23
5	Conclusões e Trabalhos Futuros	27
A	Pratica	A1

Lista de Tabelas

2.1	Características de níveis de segurança QoS do MQTT	6
2.2	Diferença entre a ESP8266 e a ESP32	9
2.3	Comparação entre os sensores DHT11 e DHT22	10
2.4	Especificações do Sensor ultrassônico HC-SR04	11
2.5	Especificações do Sensor de intensidade de luz BH1750	11
2.6	Especificações do Servo motor SG90	12
3.1	Ligação dos sensores no ESP32	15
3.2	Inscrições e Publicações	18

Lista de Figuras

2.1	Esquema de publisher-subscriber do protocolo MQTT.	6
2.2	Interface gráfica do software Node-RED.	7
2.3	ESP32 Pinout.	8
2.4	Sensor de temperatura e humidade DHT11.	10
2.5	Sensor ultrassônico HC-SR04.	11
2.6	Sensor de intensidade de luz BH1750.	11
2.7	Servo motor SG90.	12
2.8	Interface do Arduino IDE.	13
3.1	Protótipo do projeto.	16
3.2	Arquitetura do projeto	17
3.3	Trecho do código para realizar a conexão WiFi e MQTT.	17
3.4	Conexão MQTT.	18
3.5	Representação da ligação no node-RED.	19
3.6	Nós do servo no Node-RED.	20
3.7	Código do sensor de intensidade da luz.	20
3.8	Painel de controle do led.	21
3.9	Configuração do microfone Node-RED.	21
4.1	Protótipo implementado na placa de ensaio.	23
4.2	Fluxo geral implementado no Node-RED.	24
4.3	Dashboard gerada para a interação com o usuário.	24

Capítulo 1

Introdução

A automação residencial está se tornando cada vez mais relevante na era tecnológica. Essa tendência é impulsionada pela capacidade de monitorizar e controlar dispositivos de forma remota em uma residência, proporcionando comodidade, segurança e economia. Este relatório tem como objetivo apresentar o desenvolvimento de um sistema de casa inteligente. Um dos principais benefícios da automação residencial baseada em tecnologias como o ESP32, Node-RED e MQTT é o custo acessível em comparação com sistemas proprietários. Na era da Internet das Coisas (IoT), a conectividade torna-se a espinha dorsal de inovações em diversas áreas. Ashton, em sua obra seminal [1], destaca a transformação de objetos comuns em entidades inteligentes, capazes de coletar e compartilhar dados. Essa revolução tecnológica não apenas redefine a interação humana com o ambiente, mas também cria oportunidades infinitas para a automação residencial.

1.1 Automação Residencial: Definição e Evolução

A automação residencial, buscando facilitar as atividades humanas em diversos contextos, como comercial, industrial, doméstico e rural, é um campo em constante evolução. Esse processo envolve a aplicação de dispositivos automáticos, eletrônicos e inteligentes para automatizar uma variedade de tarefas.

Dentro desse amplo cenário, a automação residencial se destaca como uma categoria

específica, concentrada na automação de processos em residências, apartamentos e escritórios. Conhecida também como automação domiciliar, automatização residencial ou domótica, essa área visa melhorar a qualidade de vida integrando tecnologias inteligentes às rotinas diárias.

É interessante notar que o desenvolvimento da automação residencial começou mais tarde em comparação com áreas industriais e comerciais, principalmente devido a considerações econômicas e de escala de produção. O conceito solidificou-se nos anos 60 e, desde então, experimentou um crescimento significativo, impulsionado pelas demandas contemporâneas relacionadas à segurança, qualidade de vida e eficiência energética.

Esses sistemas representam áreas-chave onde a automação residencial busca aprimorar a eficiência, segurança e conforto dos moradores, tornando as residências mais inteligentes e adaptáveis às necessidades modernas. O constante desenvolvimento nesse campo reflete a busca por soluções inovadoras que atendam às exigências contemporâneas[2].

1.2 Objetivos

A automação residencial é um campo em constante crescimento, transformando residências comuns em ambientes altamente eficientes e conectados. No decorrer deste trabalho, exploraremos como a automação residencial funciona, realizando uma simulação em um quarto específico. Nessa simulação, será possível controlar a lâmpada, as janelas, monitorizar a temperatura e a humidade do ambiente. Além disso, entenderemos como o microcontrolador ESP32 se comunica com os dispositivos e a plataforma de controle (dashboard).

Através desse estudo, também buscaremos adquirir uma compreensão mais profunda sobre a interação e controle de dispositivos em uma casa inteligente, o que servirá como base para projetos futuros. A intenção é explorar e implementar funcionalidades mais complexas à medida que avançamos no campo da automação residencial.

De modo a cumprir o objetivo principal, os objetivos específicos foram estabelecidos:

- Desenvolvimento de um dashboard;

- Comunicação com o ESP32;
- Controle remoto e monitoramento;
- Criar fluxos de controle de dispositivos.

1.3 Organização do documento

Este documento está organizado em 5 capítulos, começando pelo presente capítulo onde foram apresentadas a proposta do trabalho, a contextualização e os objetivos. O capítulo 2 apresenta a descrição das tecnologias utilizadas para a implementação do projeto e o capítulo 3 apresenta a implementação prática realizada. O capítulo 4 apresenta os resultados obtidos e, por fim, o capítulo 5 apresenta as conclusões e trabalhos futuros.

Capítulo 2

Tecnologias e Ferramentas Utilizadas

Neste capítulo são apresentadas o funcionamento das ferramentas utilizadas para a implementação do projeto, nomeadamente o protocolo MQTT, o microcontrolador ESP32, o software Node-RED, e os sensores e atuadores empregados. Este aprofundamento teórico permitirá uma compreensão mais abrangente de como essas tecnologias desempenham um papel fundamental na automação residencial.

2.1 MQTT (Message Queuing Telemetry Transport)

O MQTT (do inglês *Message Queue Telemetry Transport*) [3] é um protocolo de comunicação entre máquinas que se populariza a cada dia que passa. Isto porque o MQTT está se tornando um dos principais protocolos para implementações de Internet das Coisas (IoT), uma tendência de aplicação na indústria ou mesmo no dia a dia das pessoas. Este protocolo permite o envio de mensagens e comandos entre os dispositivos que utilizam MQTT por meio do TCP/IP (Transmission Control Protocol/Internet Protocol). Os clientes informam de quais tópicos desejam se tornar assinantes e passam a receber as publicações realizadas nesses tópicos. Com relação à segurança dos dados, o MQTT provê três níveis de Quality of Service (QoS), sendo este responsável por gerenciar a qualidade do tráfego de dados na rede. A escolha entre níveis depende da necessidade de cada aplicação. A figura 2.1 apresenta o esquema de publisher-subscriber do protocolo MQTT.

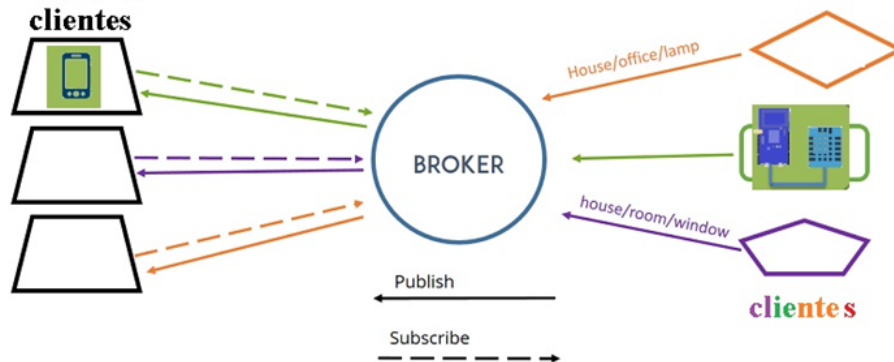


Figura 2.1: Esquema de publisher-subscriber do protocolo MQTT.

2.1.1 QoS (Quality of Service)

QoS (Quality of Service) trata a qualidade de serviço entregue da mensagem, sendo descritos em três níveis caracterizadas na tabela 2.1.

quality of service	retransmissão	confirmação de recebimento	Risco de multiplas entregas
N0	Não	Não	Não
N1	Sim	Sim	Sim
N2	Sim	Sim	Não

Tabela 2.1: Características de níveis de segurança QoS do MQTT

- Qos 0 – A mensagem é enviada uma única vez, porém não há garantia de que a mensagem será entregue pois não existe uma retransmissão da mensagem. No entanto é a mais efetiva por ter menos latência e menos uso da largura de banda.
- QoS 1 – A mensagem é enviada, notando se houver algum tipo de falha é enviada mais uma vez. Porém há risco de duplicidade na entrega. Normalmente usado na impossibilidade de perda da mensagem.
- QoS 2 – A mensagem segue um critério maior para que a mensagem seja entregue apenas uma vez e não haja duplicidade .

2.2 Node-RED

Node-RED [4] é uma plataforma de código aberto que oferece uma interface gráfica baseada na web para o desenvolvimento de fluxos de automação e integração. Ele é projetado para ser acessível a uma ampla gama de usuários, desde iniciantes até desenvolvedores experientes. A principal característica do Node-RED é a sua abordagem visual para a criação de automações, na qual os usuários podem definir a lógica do fluxo conectando nós (nodes).

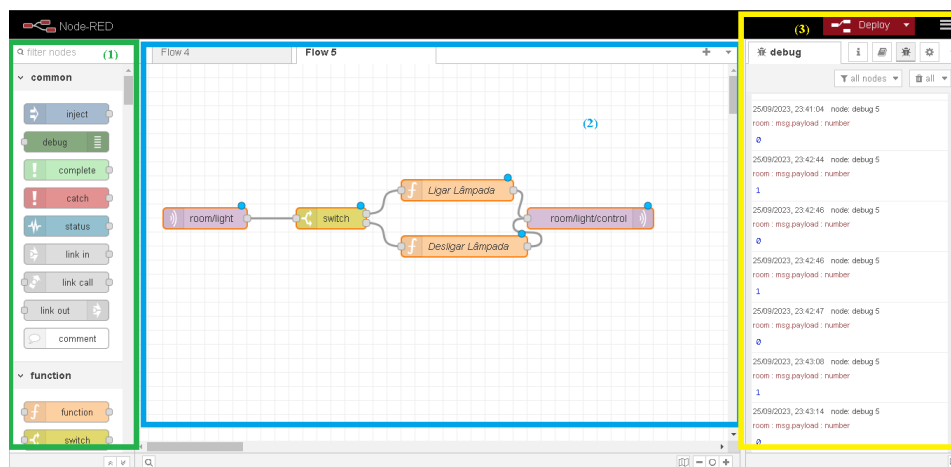


Figura 2.2: Interface gráfica do software Node-RED.

A Figura 2.2 apresenta a página principal da ferramenta Node-RED, na qual a área verde destaca onde se encontra os *nodes* que são os blocos de construção fundamentais usados para criar fluxos de automação e integração. Cada nó desempenha uma função específica e pode ser configurado para executar uma ação, os nós são conectados em um ambiente de programação visual. A área destacada em azul é onde se pode definir a lógica de automação conectando os nós da maneira desejada. A área destacada em amarelo é onde podendo ver o *debug* das informações e das configurações do flow.

2.3 ESP32

O ESP32 [5] é um microcontrolador de código aberto muito popular no mundo da eletrônica e da Internet das Coisas (IoT). Ele é produzido pela Espressif Systems, uma empresa chinesa especializada em chips e módulos sem fio. O ESP32 é amplamente utilizado para projetos de IoT, automação residencial, dispositivos conectados e muito mais devido à sua combinação de recursos e preço acessível. Uma das características mais destacadas do ESP32 é a sua conectividade. Ele suporta Wi-Fi, permitindo a conexão com redes sem fio, e também possui suporte para Bluetooth, incluindo Bluetooth Low Energy (BLE). Isso permite que o ESP32 se comunique com outros dispositivos sem fio, como smartphones, tablets e outros dispositivos. Além disso, o ESP32 é conhecido por sua capacidade de processamento e pode ser facilmente programado com a utilização da plataforma Arduino, o que é amplamente reconhecido pela comunidade de desenvolvedores. O ESP-WROOM-32 é uma placa que contém o módulo ESP32, conforme ilustra a figura 2.3, onde é possível notar suas várias funcionalidades distribuídas em sua versão com 36 pinos. Possui pinos para interface GPIO, UART, I2C, sensores de toque, sensor de efeito hall, conversores AD e DA, entre outros.

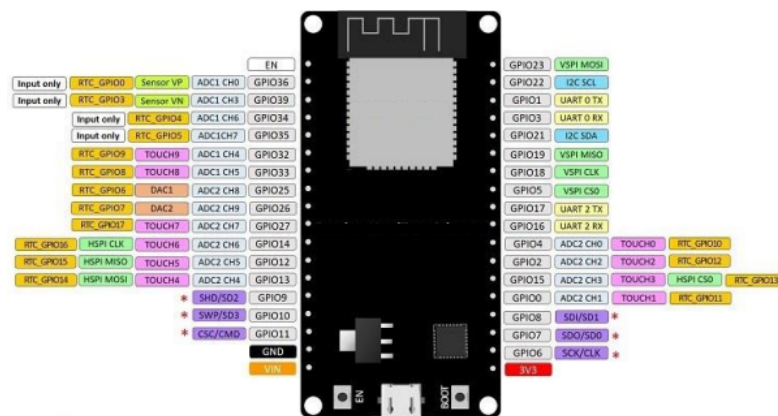


Figura 2.3: ESP32 Pinout.

2.3.1 Diferenças entre os microcontroladores

Os microcontroladores possuem diferenças sutis nas versões de seus componentes. No entanto, o ESP32 possui mais funcionalidades e também maior número de GPIOs e interfaces em relação ao ESP8266, permitindo uma implementação mais completa de um projeto. Na Tabela 2.2 apresenta as diferenças entre os dois microcontroladores.

Especificações	ESP8266	ESP32
MCU	xtensa® single 32-bit 06	xtensa® DUAL-CORE32-bit Lx6 600 DMIPS
802.11 b/g/n Wi-Fi	HT20	HT40
Bluetooth	Nao	Bluetooth 4.2 Le
Frequencia	80 MHz	160 MHz
SRAM	160 Kbytes	S12 Kbytes
Flash	SPI Flash, 16 Mbytes	SPI Flash, 16 Mbytes
GPIO	17	36
Hardware/ Software PWM	Nao/ 8 canais	1/16 can
SPI 2c/i2s/uaRT	2/1/2/2	4/2/2/2
ADC	10-bit	12-bit
CAN	Não	1
Interface Ethernet Mac	Não	1
Sensor Capacitico	Não	Sim
Sensor de Temperatura	Não	Sim
Temperatura de trabalho	-40°C a 125°C	-40°C a 125°C

Tabela 2.2: Diferença entre a ESP8266 e a ESP32

2.4 Sensor de temperatura e Humidade DHT11

O DHT11 [6] é um sensor de temperatura e humidade que permite realizar leituras precisas de temperatura e humidade do ambiente. Este sensor é conhecido por fornecer saídas digitais, o que significa que os dados capturados são codificados digitalmente, tornando-os facilmente interpretáveis por um microcontrolador. O DHT11 é uma escolha popular para projetos de automação residencial e IoT devido à sua capacidade de fornecer informações ambientais críticas de maneira confiável e acessível. A figura 2.4 apresenta o sensor DHT11.



Figura 2.4: Sensor de temperatura e humidade DHT11.

2.4.1 Diferenças entre os sensores DHT11 e DHT22

O DHT22 é geralmente mais preciso, tem uma faixa de operação mais ampla e é resistente à umidade, mas é mais caro e consome mais energia. O DHT11 é mais econômico, tem uma faixa de operação mais limitada e é menos preciso, mas oferece uma taxa de atualização mais rápida, na Tabela 2.3 apresenta a comparação entre os sensores .

	DHT11	DHT22
Alimentação	3v-5.5v	3.3v-6v
Faixa de leitura - Humidade	20-80 %	0-100 %
Precisão - Humidade	5 %	5 %
Faixa de leitura- Temperatura	0-50°C	-40-125°C
Precisão - Temperatura	+/-2°C	+/-0,5°C
Intervalo entre medidas	1s	2s

Tabela 2.3: Comparação entre os sensores DHT11 e DHT22

2.5 Sensor de distância ultrasónico HC-SR04

O sensor de distância [7] composto por um emissor e um receptor, com capacidade de medir distâncias de 2cm até 4m, com uma precisão de aproximadamente 3mm. Este sensor emite sinais ultra-sónicos que refletem no objeto a ser atingido e retornam ao sensor, indicando a distância do alvo. A figura 2.5 apresenta o sensor ultrassónico e a Tabela 2.4 apresenta as especificações.

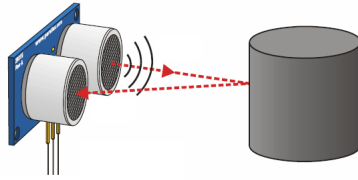


Figura 2.5: Sensor ultrassônico HC-SR04.

Alimentacao	5v (DC)
Corrente de trabalho:	<2mA;
Distância de Trabalho	2cm até 500cm;
Sinal de Saida	5V de alto nível, 0V nível baixo
Sinal de entrada trigger	10us impulso TTL
Sinal Echo	saída TTL PWL sinal

Tabela 2.4: Especificações do Sensor ultrassônico HC-SR04

2.6 Sensor de intensidade de luz BH1750

O Sensor de luz BH1750-FVI GY-30 [8] é capaz de medir a intensidade da luz ambiente em lux na faixa de 1 a 65535 lx, este módulo é ideal para aplicação em sistemas de controle de iluminação automático, ou qualquer circuito para medição da luz. A leitura é facilmente interpretada e usada para ajustar a iluminação ou tomar ações com base na luz ambiente. A figura 2.6 apresenta o sensor BH1750 e a Tabela 2.5 apresenta as especificações.



Figura 2.6: Sensor de intensidade de luz BH1750.

Alimentacao	3 V á 5 V
Resolução	1lx (lux);
Faixa de medição	1~65535 lux
Baixa variação de medição	+/- 20

Tabela 2.5: Especificações do Sensor de intensidade de luz BH1750

2.7 Servo motor SG90 9g Tower Pro

O servo motor [9] é um módulo que apresenta movimentos proporcionais aos comandos indicados, controlando o girar e a posição, diferente da maioria dos motores. Possui um ângulo de rotação de 180 graus e é usado em projetos de robótica e automação para realizar movimentos precisos. A figura 2.7 apresenta o servo motor SG90 e a Tabela 2.6 apresenta as especificações.



Figura 2.7: Servo motor SG90.

Tensão da operação:	3,0 - 7,2V
Ângulo de rotação	180 graus
Velocidade	0,12 seg/60° (4,8V) sem carga
Torque	1,2 kg.cm (4,8V) e 1,6 kg.cm (6,0V)
Temperatura da Operação	30°C - +60°C

Tabela 2.6: Especificações do Servo motor SG90

2.8 Arduino IDE

A IDE do Arduino [10] é um ambiente aberto de desenvolvimento integrado que foi desenvolvido para programar microcontroladores, principalmente para o Arduino. É um software de código aberto que dispõe de uma interface amigável e simplificada para diversos níveis de experiência em desenvolvimento de programadores.

Ela possui recursos que permite escrever, compilar e carregar código nos microcontroladores de forma simples e prática. Sua interface é de fácil entendimento e visualização,

sendo muito intuitiva.

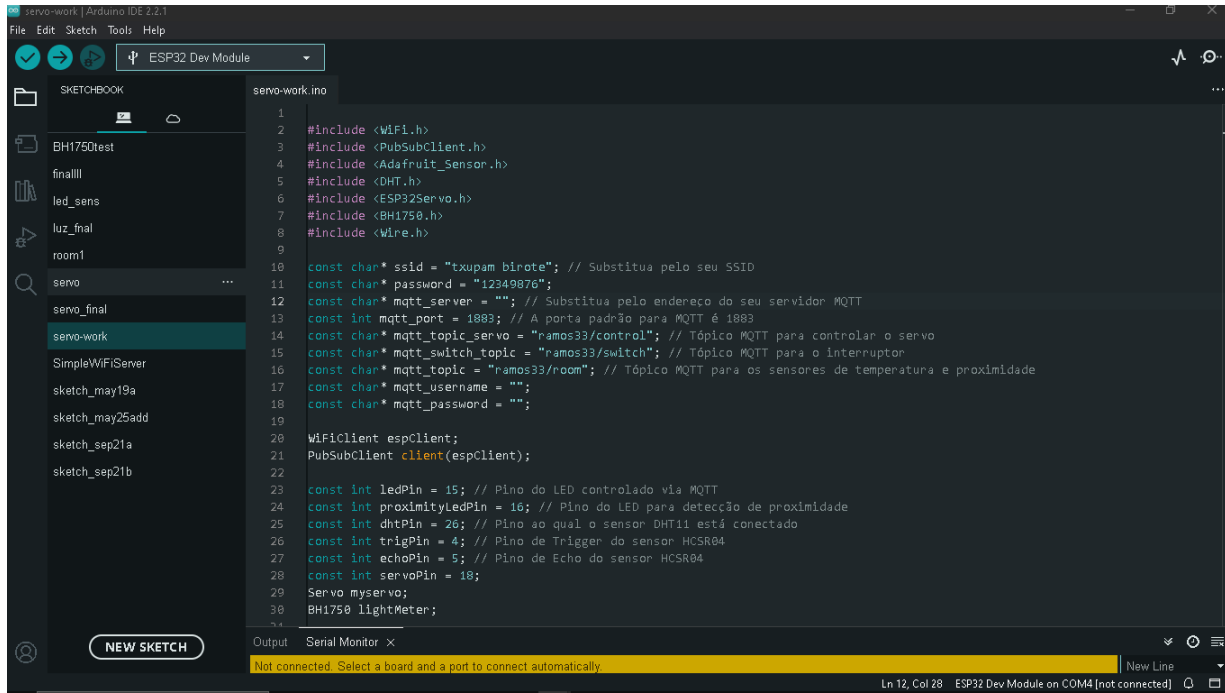


Figura 2.8: Interface do Arduino IDE.

Capítulo 3

Desenvolvimento

3.1 Implementação

O objetivo principal deste projeto é criar um sistema para controlar aparelhos e dispositivos. No exemplo de um quarto, é possível controlar a lâmpada remotamente pela interface web, enquanto outro LED é acionado pelo sensor de distância e movimento, permanecendo ativo por um período determinado. O controle das janelas é realizado por um servo motor, que pode ser ajustado remotamente e possui um modo automático baseado na intensidade luminosa.

O sensor de umidade e temperatura foi adicionado para monitorar as condições climáticas. O protótipo, representado na Figura 3.1, foi concebido com essas funções específicas em mente, antes de ser implementado na prática usando o simulador de projetos Workwi [11]. A tabela 3.1 apresenta a ligação dos sensores e indicando em qual porta se encontra ligado.

DHT11	Sensor ultrassônico	Sensor de intensidade de luz	Servo motor	LED
VCC	VCC	VCC	VCC	LED1-15
Data-26	TRIG-4	SDA-21	SP-18	LED2-16
GND	ECHO-5	SCL-22	GND	
	GND	GND		

Tabela 3.1: Ligação dos sensores no ESP32

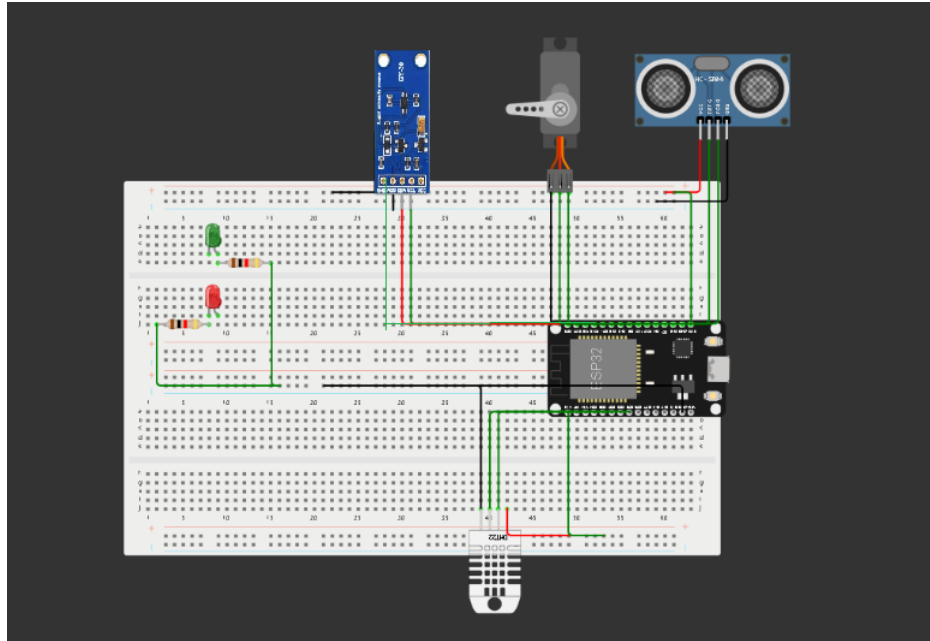


Figura 3.1: Protótipo do projeto.

3.1.1 Arquitetura do projeto

A arquitetura do projeto é apresentada na figura 3.2 tendo como componentes principais:

- Microcontrolador: Responsável pelo controle do sistema, ele se comunica com os sensores e atuadores.
- Sensores e atuadores: Utilizados para medir e interagir com o ambiente.
- Node-RED: Um sistema de automação que é utilizado como cliente para controlar sistemas é projetado para disponibilizar uma interface gráfica que permite a interação do usuário. Adicionalmente, é desejável que alguns mecanismos avançados, como o reconhecimento de comandos de voz, sejam incorporados.
- Mosquitto Broker: Funciona como um intermediário na comunicação entre os sistemas (clientes), seguindo o modelo de publicação/assinatura do protocolo MQTT (Message Queuing Telemetry Transport).

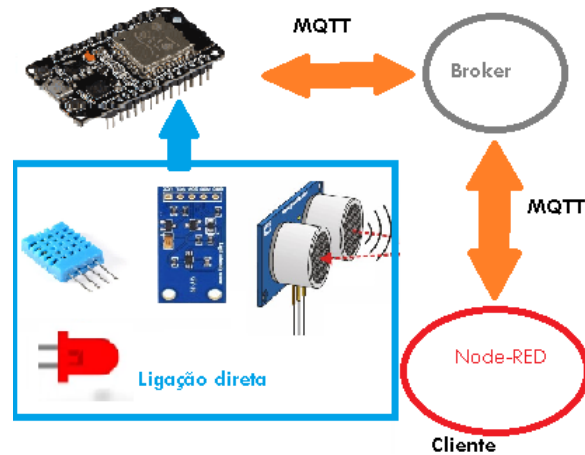


Figura 3.2: Arquitetura do projeto

3.1.2 Conexão entre o ESP32 ao wifi

Os primeiros passos para o desenvolvimento do projeto foi conectar o ESP32 ao WiFi, para isso, foi utilizado o arduino IDE para realizar toda a programação. A Figura 3.3 apresenta o código usado para conectar o ESP32 ao WiFi e ao MQTT broker, enviando informação sobre a conexão.

```

//
const char* ssid = "txupamirrote"; // Substitua pelo seu SSID
const char* password = "12349876";
const char* mqtt_server = "broker.hivemq.com"; // Substitua pelo endereço do seu servidor MQTT
const int mqtt_port = 1883; // A porta padrão para MQTT é 1883
const char* mqtt_topic = "servo/control"; // Tópico MQTT para controlar o servo
const char* mqtt_username = "";
const char* mqtt_password = "";

const int servoPin = 18;
Servo myservo;
WiFiClient espClient;
PubSubClient client(espClient);

void setup_wifi() {
  delay(10);
  // Conecta-se à rede Wi-Fi
  Serial.println();
  Serial.print("Conectando-se a ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi conectado");
  Serial.println("Endereço IP: ");
  Serial.println(WiFi.localIP());
}

```

Figura 3.3: Trecho do código para realizar a conexão WiFi e MQTT.

3.1.3 Conexão entre o MQTT e o node-RED

A conexão entre o MQTT e o node-RED é estabelecida quando usamos o recetor MQTT definindo o broker e o tópico da mensagem que deve receber, após receber o node-RED é capaz de representar estes valores, como por exemplo na Figura 3.4, podemos ver que este recetor MQTT está configurado para receber informações sobre a humidade e este será representado graficamente.

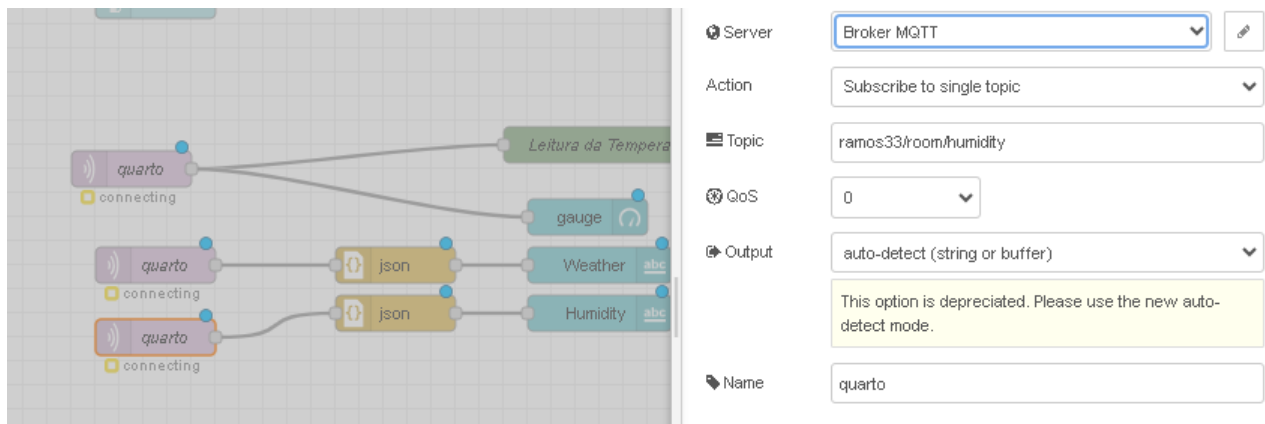


Figura 3.4: Conexão MQTT.

3.1.4 Inscrições e Publicações

O programa carregado no ESP32 através da Arduino IDE configura o dispositivo como um assinante MQTT para tópicos específicos listados na Tabela 3.2. Quando o ESP32 recebe mensagens em um desses tópicos, o programa interpreta o conteúdo da mensagem e escreve um valor digital correspondente. Em termos simples, o ESP32 reage às mensagens MQTT, executando ações com base no conteúdo recebido.

Tabela 3.2: Inscrições e Publicações

Dispositivo	Inscrição	Publicação
ESP32	ramos33/control ramos33/switch ramos33/room	ramos33/room/temperature ramos33/room/humidity

3.1.5 Aplicação dos sensores

O primeiro sensor a ser aplicado foi o DHT11 conectado ao ESP32, com a função de ler a temperatura e a humidade do quarto e enviar os dados por MQTT. Os dados enviados por MQTT são recebidos pelo Node-RED e os valores são representados graficamente na dashboard, também foi adicionado um LED na dashboard que é ligado quando a temperatura atingir um certo valor definido pelo usuario, indicando que o sistema de ar condicionado estaria ativado ou desativado. A Figura 3.5 mostra como é ligado no node-RED.

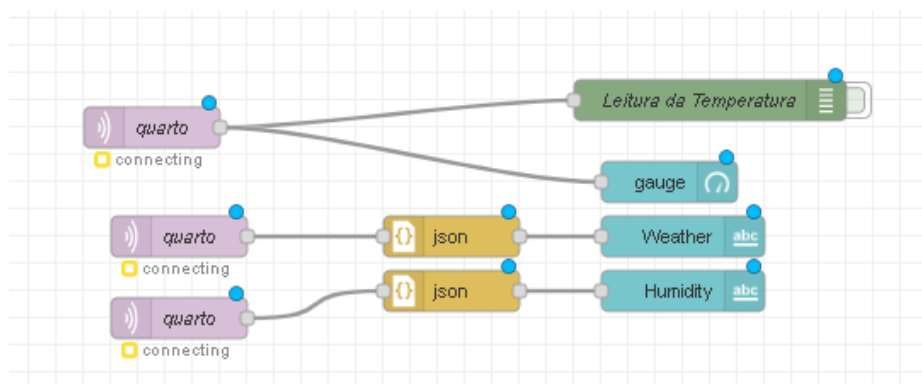


Figura 3.5: Representação da ligação no node-RED.

Na figura 3.6, mostra como internamente é ligado o controle do servo no Node-RED, os botões já com valores definidos quando é pressionado envia os dados por MQTT, a mensagem chega ao broker que instantaneamente envia uma mensagem para o ESP32 que faz uma ação sobre o servo mudando assim de posição, também foi adicionado um slider para controlar a posição, o microfone para o abrir e fechar, e um switch para ativar o modo automático e manual.

Para o sistema de controle automático, a posição do servo é ajustada com base nos valores de intensidade de luz detectados pelo sensor. Com base nos dados de leitura os valores são agrupados em 3 grupos que estão ligadas as suas devidas posições como pode ser visto na Figura 3.7.

Para controlar o LED podemos usar os botões ON/OFF ou simplesmente com um click no ícone do microfone que esta programado para reconhecer a fala do utilizador ao

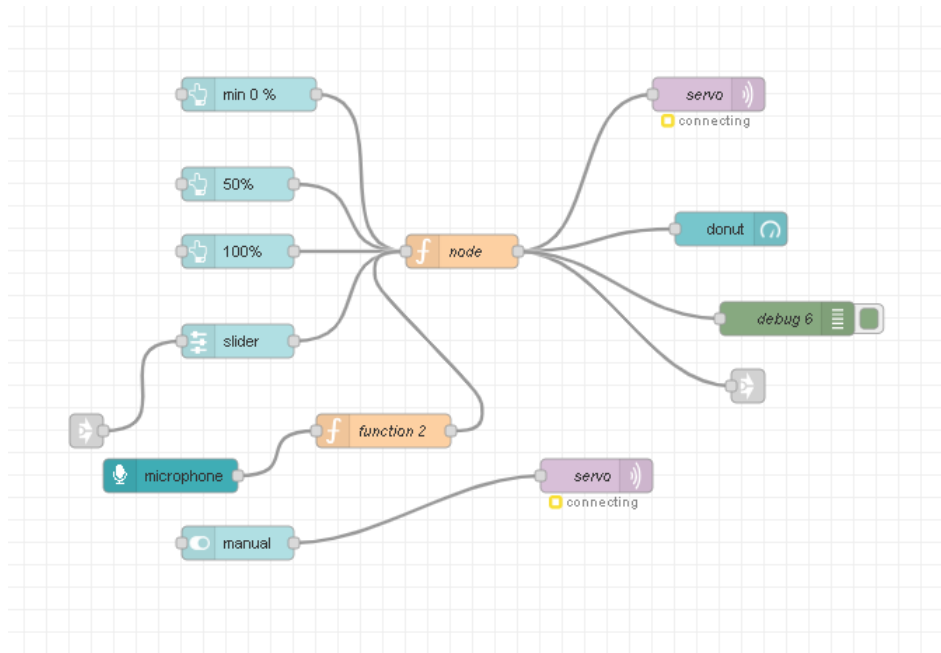


Figura 3.6: Nós do servo no Node-RED.

```
// Verifique o valor do interruptor para determinar se o sensor de luz deve ser lido
if (switchValue == 1) {
  unsigned long currentTime = millis();
  if (currentTime - lastLightReadTime >= lightReadInterval) {
    float lux = lightMeter.readLightLevel();
    Serial.print("Light: ");
    Serial.print(lux);
    Serial.println(" lx");

    // Defina os limites para os níveis de luz e controle o servo
    if (lux >= 450) {
      servoAngle = 0; // Posição 0%
    } else if (lux < 450 && lux >= 185) {
      servoAngle = 90; // Posição 50%
    } else {
      servoAngle = 180; // Posição 100%
    }
  }
}
```

Figura 3.7: Código do sensor de intensidade da luz.

falar "light on / light off", acendendo ou apagando o LED. Na Figura 3.8 podemos ver os dois botões e o microfone.

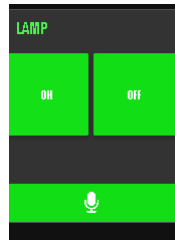


Figura 3.8: Painel de controle do led.

O controle por voz funciona a partir da conexão entre o node-RED e o MQTT não sendo preciso ter um microfone instalado na ESP32, pois nesse caso o microfone do computador ou *smartphone* do utilizador que será utilizado. Quando damos o node-RED o poder de ouvir conseguimos criar funções que enviam valores para o MQTT, no nosso caso precisamos acender e apagar a luz então programamos o node-RED que quando ouvir light on/off envia os valores 1 / 0 como está representada na Figura 3.9.

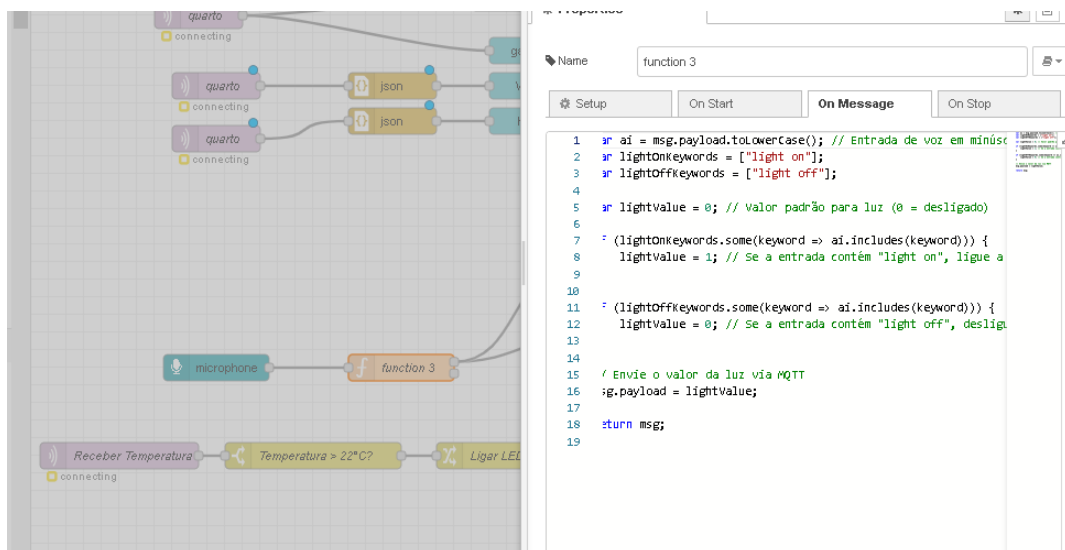


Figura 3.9: Configuração do microfone Node-RED.

E por ultimo o sensor de distância HC-sr04, quando o sensor detecta movimento a menos de 20cm, um LED será ligado por 10 segundos, representando um sensor de movimento e uma lâmpada.

Capítulo 4

Resultados

Na Figura 4.1 é apresentada a montagem dos sensores, leds e microcontrolador ESP32 na placa de ensaio (*breadboard*).

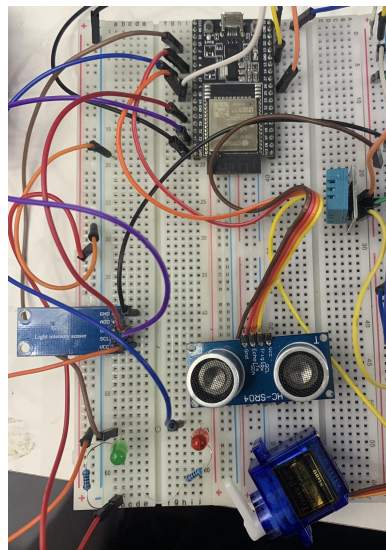


Figura 4.1: Protótipo implementado na placa de ensaio.

Após a definição de todas as funções e a elaboração do código de operação geral, a Figura 4.2 ilustra as interconexões dos nós no Node-RED, enquanto a Figura 4.3 exibe a interface do usuário. Esta interface inclui um centro de controle que permite a modificação do estado do LED, visualização dos valores de humidade e temperatura, bem como o controle da janela.

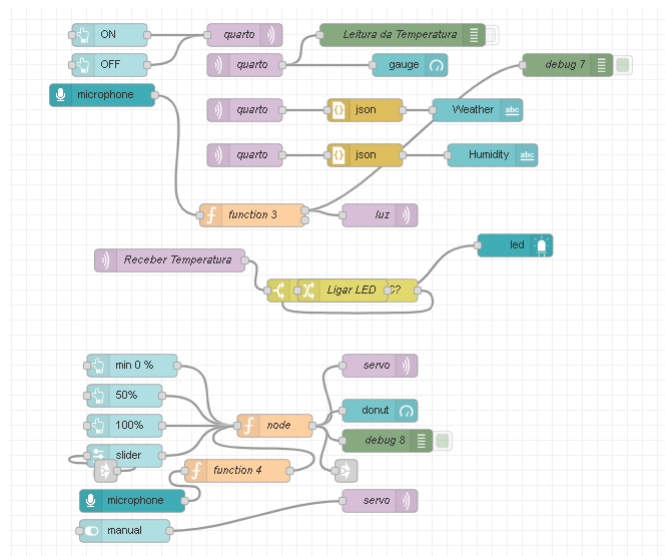


Figura 4.2: Fluxo geral implementado no Node-RED.

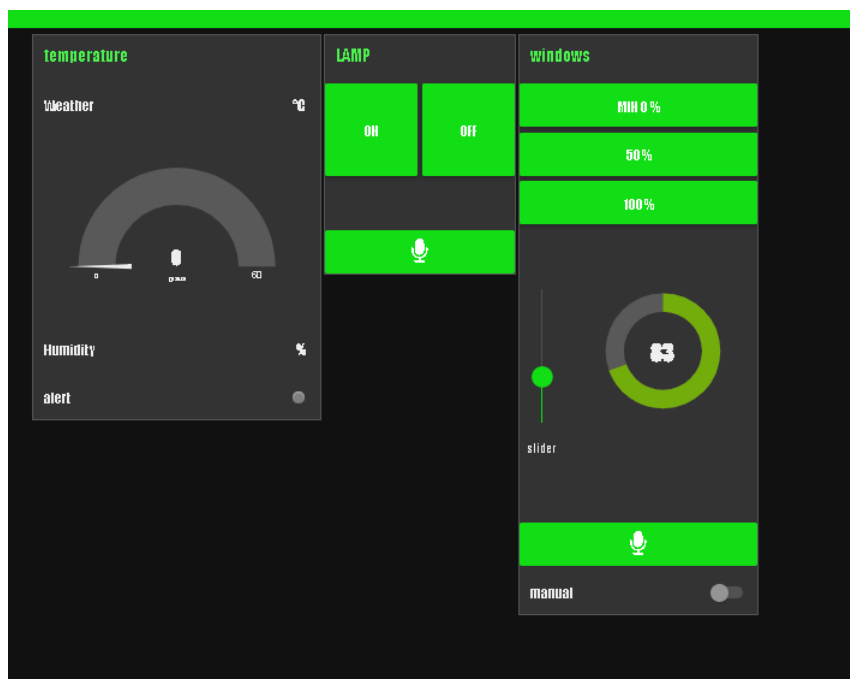


Figura 4.3: Dashboard gerada para a interação com o usuário.

O sistema implementado com o uso do ESP32, Node-RED e MQTT demonstrou um funcionamento eficaz e confiável, sendo possível controlar a lâmpada, as janelas e monitorar a temperatura e humidade do ambiente de forma remota, proporcionando comodidade e segurança ao usuários. A comunicação entre o microcontrolador ESP32 e os dispositivos foi bem sucedida transmitindo informações de maneira consistente. Algumas das principais vantagens do sistema :

- Eficiência na transmissão de dados;
- Facilidade de configuração na plataforma Node-RED;
- O uso do protocolo MQTT contribuindo para uma comunicação eficiente entre os dispositivos;
- Adaptabilidade e expansibilidade;
- Monitoramento remoto e controle efetivo.

Capítulo 5

Conclusões e Trabalhos Futuros

Em um mundo tecnológico em constante evolução, a automação residencial se destaca como uma conquista significativa. A capacidade de controlar dispositivos e realizar ações com um simples toque ou comando de voz oferece conveniência, segurança e acessibilidade a todos os usuários.

A facilidade e o conforto que a tecnologia proporciona ao permitir o controle de dispositivos são de grande importância. A plataforma Node-RED desempenha um papel fundamental na criação desse projeto, permitindo a criação de um painel de controle para gerenciar dispositivos de forma eficaz. O microcontrolador ESP32 atua como o centro de comunicação, e o protocolo MQTT é escolhido para a comunicação entre dispositivos devido à sua eficiência. Além disso, a segurança é uma prioridade, com a opção de criar código de acesso para o painel de controle, garantindo que apenas usuários autorizados tenham acesso. Em resumo, a automação residencial não apenas simplifica a vida cotidiana, mas também aumenta a eficiência e a segurança em nossas casas, demonstrando o impacto positivo da tecnologia na melhoria da qualidade de vida.

Ao fazer este projeto vimos o quanto de coisas que podemos ter o controle facilitando o uso, poupando tempo, energia e ajudando os que mas necessitam, pensando na vida de um idoso ou pessoas com dificuldade. Com o desenvolver do trabalho surgiram varias ideias como:

- Adicionar mais dispositivos controlados e monitorados, como sistemas de segurança, fechaduras inteligentes, sistemas de áudio/vídeo, entre outros.
- Explorar a IA e o aprendizado de máquina para criar um sistemas inteligentes.
- Desenvolver rotinas de automação que facilitem as atividades diárias
- Ampliar as capacidades de acesso e controle remoto, permitindo o acesso em qualquer lugar.

Embora a automação residencial tenha alcançado avanços notáveis, desafios persistem. Questões de segurança e privacidade são considerações críticas, especialmente quando se trata de dispositivos conectados. No entanto, com a evolução contínua das tecnologias e a conscientização crescente, espera-se que esses desafios sejam endereçados. O futuro da automação residencial promete uma integração ainda mais intuitiva, onde a inteligência artificial e a aprendizagem de máquina podem antecipar as necessidades dos moradores.

Bibliografia

- [1] T. Kramp, R. van Kranenburg e S. Lange, “Introduction to the Internet of Things”, em *Enabling Things to Talk*. Springer Berlin Heidelberg, 2013, pp. 1–10, ISBN: 9783642404030. DOI: 10.1007/978-3-642-40403-0_1.
- [2] *Enchanted Objects: Design, Human desire and the Internet of Things*, <https://responsiveobjects.files.wordpress.com/2016/01/enchanted-objects-design-human-desire-and-the-internet-of-things-rose-david.pdf>, Acesso: 2023-12-06.
- [3] *MQTT*, <http://mqtt.org/faq>, Acesso: 2023-06-08.
- [4] *NODE-RED*, <https://nodered.org/>, Acesso: 2023-10-18.
- [5] *ESP32*, <https://blog.eletrogate.com/conhecendo-o-esp32-introducao-1/>, Acesso: 2023-09-23.
- [6] *DHT11*, <https://www.makerhero.com/produto/sensor-de-umidade-e-temperatura-dht11/>, Acesso: 2023-09-23.
- [7] *Sensor de distancia*, <https://www.electrofun.pt/sensores-arduino/sensor-distancia-hc-sr04>, Acesso: 2023-09-23.
- [8] *Sensor de intensidade da luz*, <https://www.casadarobotica.com/sensores-modulos/sensores/luminosidade/modulo-gy-30-sensor-de-luminosidade-luz-bh1750fvi>, Acesso: 2023-09-20.
- [9] *Servo motor*, <https://www.electrofun.pt/robotica/servo-motor-sg90-9g-tower-pro-360>, Acesso: 2023-09-23.

- [10] *Arduino IDE*, <https://revistaft.com.br/iot-para-controle-e-gerenciamento-residencial/>, Acesso: 2023-09-23.
- [11] *Wokwi*, <https://wokwi.com/>, Acesso: 2023-12-06.

Apêndice A

Pratica

Toda a parte do código feito no arduino e os flows no Node-RED poderão ser acessados a partir do link (GitHub) disponível no projeto. As aplicações serão apresentadas em um curto vídeo explicando o funcionamento geral.