



## AUTOMAÇÃO DE PORTÕES VIA NUVEM COM ESP32

KAUAN ROCHA DA SILVA

LETÍCIA MOCCI DEZANETE

RAFAEL MORAES SANTOS MUNIZ GARCIA

YURI DOMINICI PROVEDELLI

### Resumo:

Este trabalho de conclusão de curso apresenta o desenvolvimento de um sistema de automação de portões utilizando o microcontrolador ESP32 e tecnologia de computação em nuvem. O objetivo principal é proporcionar uma solução segura, eficiente e de baixo custo para o controle remoto de portões residenciais e comerciais, permitindo a abertura e fechamento deles através de dispositivos móveis conectados à internet.

A automação de portões tradicionalmente utiliza controles remotos RF (Radiofrequência), que podem apresentar limitações em termos de alcance e segurança. Com a implementação de um sistema baseado na nuvem, é possível superar essas limitações, proporcionando um controle mais amplo e seguro, além de possibilitar o monitoramento e operação do portão de qualquer lugar com acesso à internet.

### Palavras-chave:

automação de portões; esp32; computação em nuvem; IoT.

\*Técnico em Eletrônica, Etec Philadelpho Gouvêa Netto – kauanrochadasilva06@gmail.com

\*\*Técnico em Eletrônica, Etec Philadelpho Gouvêa Netto – Imoccidezanete@gmail.com

\*\*\*Técnico em Eletrônica, Etec Philadelpho Gouvêa Netto – Rafadaplay@gmail.com

\*\*\*\*Técnico em Eletrônica, Etec Philadelpho Gouvêa Netto – Yuridprovedelli@gmail.com

## **I. INTRODUÇÃO**

A automação residencial tem ganhado destaque nos últimos anos, impulsionada pelos avanços na tecnologia da Internet das Coisas (IoT). Dentro desse contexto, a automação de portões é uma aplicação prática que oferece conveniência e segurança aos usuários, permitindo o controle remoto de acessos residenciais e comerciais. Tradicionalmente, os sistemas de abertura de portões utilizam controles remotos RF (radiofrequência), que possuem limitações em termos de alcance e segurança. Com o desenvolvimento de tecnologias de computação em nuvem, surgem novas oportunidades para melhorar esses sistemas, tornando-os mais acessíveis e eficientes.

Este trabalho de conclusão de curso propõe o desenvolvimento de um sistema de automação de portões utilizando o microcontrolador ESP32 e uma infraestrutura de computação em nuvem. O ESP32 é um microcontrolador versátil com conectividade Wi-Fi integrada, ideal para aplicações IoT. A integração com a nuvem permite que o portão seja controlado remotamente através de dispositivos móveis, proporcionando maior flexibilidade e conveniência aos usuários. Além disso, a utilização da nuvem melhora a segurança, uma vez que as comunicações podem ser criptografadas e a autenticação gerenciada de forma centralizada.

### **Objetivo geral**

O sistema proposto neste trabalho visa não apenas modernizar o controle de acesso de portões, mas também explorar as potencialidades da IoT e da computação em nuvem para criar uma solução que seja ao mesmo tempo eficiente, segura e de baixo custo. A escolha do ESP32 se deve à sua popularidade, baixo custo e funcionalidades robustas, incluindo conectividade Wi-Fi e capacidade de processamento adequada para aplicações de automação residencial. A adoção de soluções baseadas em IoT, utilizando dispositivos como o ESP32, permite a superação dessas limitações, oferecendo um controle mais seguro e acessível de qualquer lugar com acesso à internet.

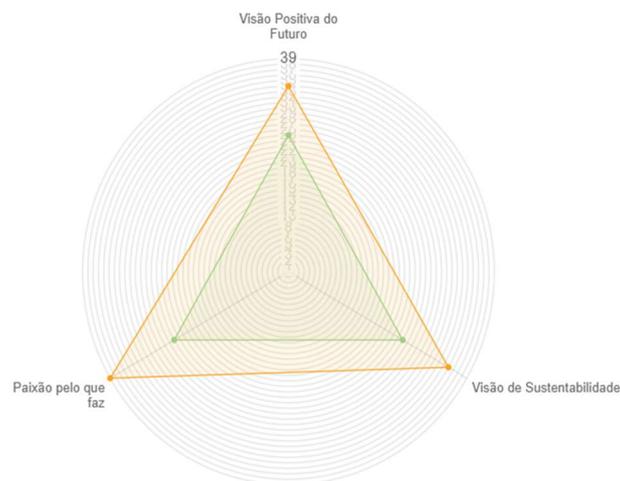
## Objetivos específicos

Desenvolver um sistema de automação para proteção utilizando o microcontrolador ESP32, capaz de controlar a abertura e fechamento do portão externamente via nuvem. O sistema deve ser integrado a sensores de posição para monitorar o estado do portão (aberto, fechado ou em movimento) e uma ponte H para gerenciar o funcionamento do motor de forma bidirecional, garantindo segurança, eficiência e confiabilidade no controle remoto. Além disso, o projeto visa utilizar plataformas de IoT gratuitamente, como Blynk, para monitorar.

## II. DESENVOLVIMENTO

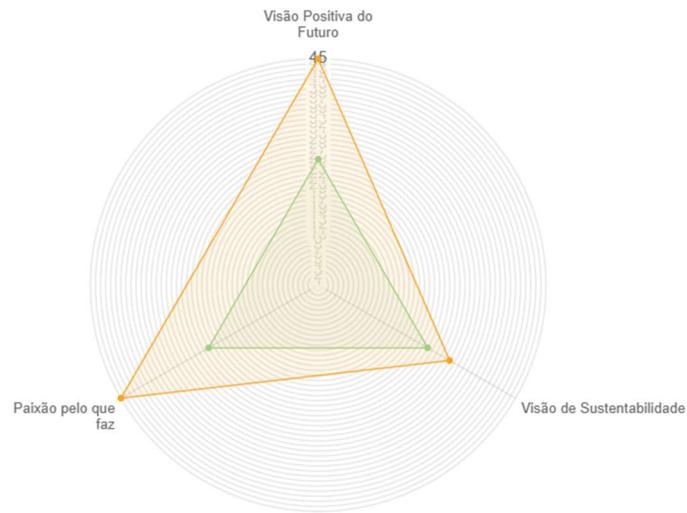
- **Desenvolvimento pessoal e profissional**

*Figura 1: Resultado da autoavaliação de Yuri D. Provedelli*



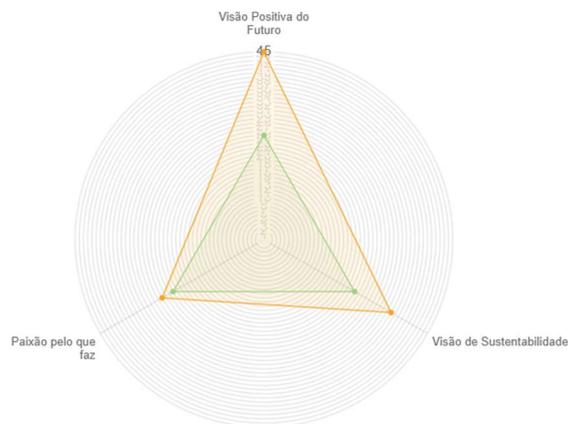
*Fonte: eZAPe!, 2024*

Figura 2: Resultado da autoavaliação de Kauan Rocha da Silva



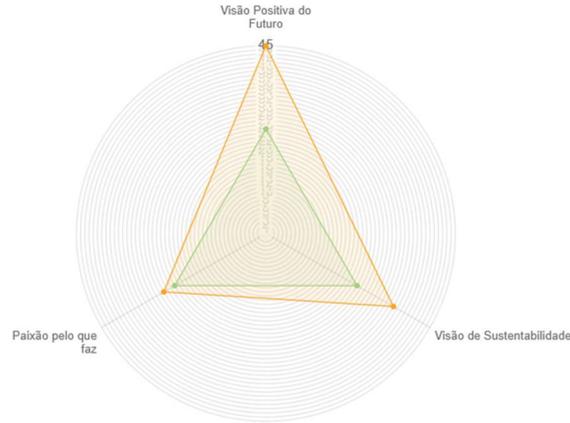
Fonte: eZAPe!, 2024

Figura 3: Resultado da autoavaliação de Leticia Mocci Deanete



Fonte: eZAPe!, 2024

Figura 4: Resultado da autoavaliação de Rafael Moraes Santos Muniz Garcia



Fonte: eZAPe!, 2024

- **Empreendedorismo**

Parceiros-Chave	Atividades-Chave	Proposta de Valor	Relações com Clientes	Clientes
Professores.	Desenvolvimento de automação residencial para portões	Desenvolvimento de um sistema de automação de portões utilizando o microcontrolador ESP32 e tecnologia de computação em nuvem utilizando app Blynk IoT.	Residências e redes sociais.	Domiciliados
	<b>Recursos-Chave</b>		<b>Canais</b>	
	ESP32 Blynk IoT Motor DC Portões		Utilizar o marketing digital por meio das redes sócias (Instagram, Tik Tok)	
<b>Despesas</b>		<b>Receitas</b>		
Compra dos equipamentos para fazer o protótipo.		Renda própria		

Fonte: eZAPe!, 2024

- **Metodologia do projeto**

A metodologia utilizada no desenvolvimento deste projeto segue um conjunto de etapas práticas e teóricas, organizadas para garantir o sucesso na implementação do sistema de automação de travamentos com controle via nuvem utilizando o ESP32.

## Fundamentação teórica

## **1. Automação e IoT**

A automação residencial, também conhecida como domótica, busca integrar dispositivos para proporcionar conforto, segurança e eficiência energética. No contexto da IoT, os dispositivos são conectados à internet, permitindo controle remoto e coleta de dados em tempo real. Conforme [Silva et al. (2021)], a aplicação de IoT na automação residencial melhora a experiência do usuário e aumenta a eficiência do sistema.

## **2. Principais Componentes Eletrônicos**

### **2.1. ESP32**

O ESP32 é um microcontrolador altamente integrado com capacidade de comunicação Wi-Fi e Bluetooth, amplamente utilizado em projetos de IoT. Segundo [Liu et al. (2020)], o ESP32 destaca-se por seu baixo custo, eficiência energética e suporte a múltiplos protocolos de comunicação. No projeto, o ESP32 é o núcleo do sistema, controlando o motor e interagindo com a nuvem.

### **2.2. Ponte H**

A ponte H é um circuito eletrônico utilizado para controle de motores DC, permitindo a inversão do sentido de rotação. Conforme [Rodrigues et al. (2019)], a ponte H é essencial em aplicações que exigem controle de direção e velocidade de motores, como em sistemas de automação de trânsito. Neste projeto, ela será controlada por GPIOs do ESP32.

### **2.3. Sensores de Posição**

Os sensores de posição detectam o estado do portão (aberto ou fechado). Sua funcionalidade garante a segurança do sistema, impedindo o funcionamento do motor em condições ambientais. Segundo [Souza e Oliveira (2020)], sensores simples podem ser usados para reduzir custos, sem comprometer a eficiência.

## **3. Comunicação IoT e Protocolo MQTT**

O MQTT (Message Queuing Telemetry Transport) é um protocolo leve, projetado para dispositivos com recursos limitados. Ele facilita a comunicação entre o ESP32 e a

nuvem, permitindo o envio de comandos e o monitoramento do estado do portão. Segundo [Pereira et al. (2021)], o MQTT é ideal para automação residencial devido à sua eficiência e simplicidade.

#### **4. Motor DC**

Um motor DC (corrente contínua) é um dispositivo eletromecânico que converte energia elétrica em energia mecânica através do movimento rotacional. Ele opera com corrente contínua, o que significa que a eletricidade flui em uma única direção através do motor. O movimento do motor é gerado quando uma corrente elétrica passa por suas bobinas, criando um campo magnético que interage com os ímãs internos, fazendo com que o rotor gire.

- **Função no projeto:**

1. Realiza o movimento de abertura e fechamento do portão.
2. Permite o controle da direção e velocidade usando uma ponte H.

#### **5. Caixa de Redução**

A caixa de redução é um sistema mecânico que reduz a velocidade do motor enquanto aumenta o torque. Isso é essencial para o movimento eficiente de portões pesados.

- **Aplicação no Projeto:**

1. Acoplada ao motor DC, garante que o portão se mova suavemente e com força suficiente para superar resistência mecânica.

#### **6. Protoboard e Jumpers**

A protoboard, também conhecida como placa de ensaio, é uma base reutilizável para a construção temporária de circuitos eletrônicos. Ela permite a conexão de componentes sem a necessidade de solda, facilitando a montagem, modificação e testes de circuitos. Jumpers são fios ou cabos que conectam diferentes pontos em uma protoboard ou entre uma protoboard e outros componentes. Eles são usados para estabelecer conexões elétricas temporárias e flexíveis.

"Utilizar protoboards e jumpers é uma prática comum na eletrônica experimental, proporcionando uma plataforma conveniente para testes e ajustes." (Monk, 2013).

## 7. Tipos de portões

Os modelos mais comuns de portões são três: os deslizantes, os basculantes e pivotantes. O propósito desse trabalho é atender os três modelos de portões. Cada modelo tem suas características:

1. **Deslizantes:** Os portões deslizantes são um dos modelos mais utilizados no Brasil, também chamados de portão de correr, devido por sua principal característica ser deslizar para o lado para se abrir.
2. **Basculantes:** O portão basculante é um tipo de portão que sua principal característica é o seu movimento de abertura e fechamento vertical. Ele é composto por uma única folha que, ao ser acionada, se levanta para permitir a passagem, geralmente girando em torno de um eixo horizontal localizado na parte superior ou inferior do portão.
3. **Pivotantes:** Portão pivotante é um tipo de portão que utiliza um sistema de pivôs para sua abertura e fechamento, em vez de dobradiças tradicionais. Esse sistema permite que o portão gire em torno de um eixo vertical, que geralmente é deslocado do centro para criar um efeito de abertura diferenciado.

## 8. Fonte de alimentação externa

A fonte de alimentação externa desempenha um papel crucial ao fornecer energia suficiente para operar todos os componentes do sistema, especialmente aqueles com altas demandas de corrente, como o motor DC.

## 9. Rotinas de Programação Relevantes

### 9.1 Controle do Motor

O controle do motor é implementado por meio de GPIOs do ESP32, que ativam os pinos da ponte H para determinar o sentido de rotação.

## 9.2 Comunicação com a Nuvem

O código do ESP32 é configurado para enviar e receber dados de uma plataforma IoT (Blynk), utilizando a biblioteca MQTT.

- **Protótipo, montagem e testes**

### 1. Elaboração do Protótipo

#### 1.1. Planejamento

##### Definição do Objetivo

- Criar um sistema para controle de motor DC, utilizando um ESP32 conectado ao aplicativo Blynk.
- Implementar sensores de fim de curso para controlar os limites de movimento.
- Garantir que o sistema seja alimentado e funcione de forma segura e eficiente.

#### 1.2. Escolha dos Componentes

- **Microcontrolador:** ESP32 (com suporte a Wi-Fi para integração ao Blynk).
- **Driver do Motor:** L298N (capaz de controlar motores DC e fornecer suporte a PWM).
- **Motor DC:** Capaz de atender à carga necessária (como movimentar um portão).
- **Sensores de Fim de Curso:** Micro interruptores para detectar o limite de movimento.
- **Fonte de Alimentação:** Separada para ESP32 (5V) e motor (12V, dependendo do motor).
- **Protoboard e Fios Jumper:** Para prototipagem e conexões temporárias.

## 2. Desenvolvimento do Circuito

### 2.1. Esquema Elétrico

O circuito foi projetado para interligar os componentes básicos:

O ESP32 controla os pinos de entrada (IN1, IN2, ENA) do driver L298N.

O L298N regula a corrente para o motor e o controla conforme os comandos recebidos. Os sensores de fim de curso são conectados ao ESP32 para enviar sinais quando os limites de movimento são alcançados. Fontes separadas garantem que o ESP32 e o motor operem de forma independente e segura.

### 3. Montagem Física

#### 3.1. Organização no Protoboard

1. O ESP32 foi colocado no centro para facilitar as conexões.
2. O driver L298N foi conectado próximo ao ESP32, com saídas ligadas ao motor e entradas ao ESP32.
3. Fios jumper foram utilizados para conectar os sensores de fim de curso aos pinos digitais do ESP32.
4. A fonte de alimentação do motor foi ligada diretamente ao L298N (nos pinos VMS e GND).

### 4. Desenvolvimento do Software

#### 4.1. Integração com o Blynk

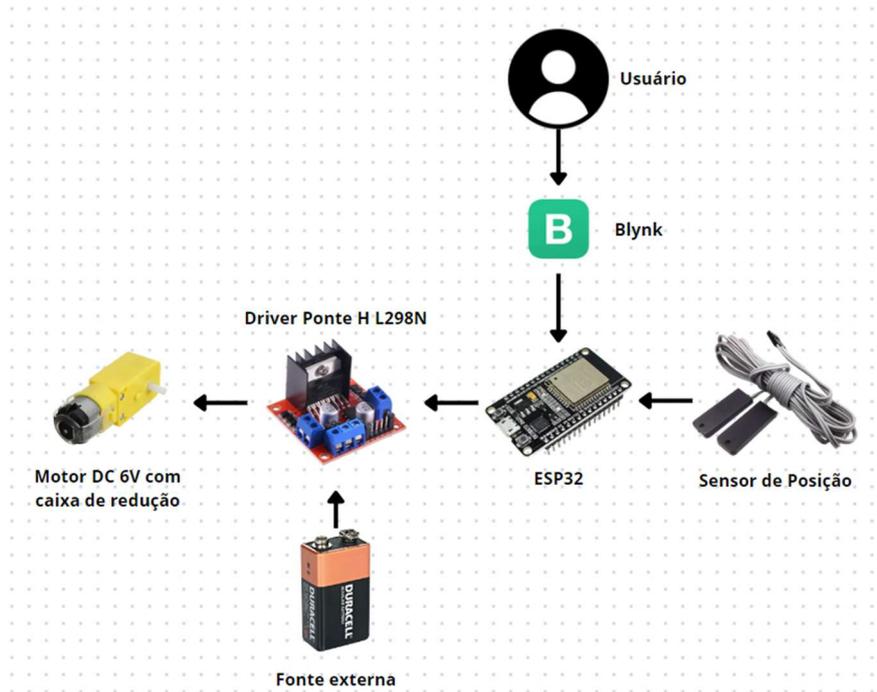
1. Criamos um projeto no aplicativo Blynk e obtivemos o **Token de Autenticação**.
2. Programamos o ESP32 para conectar ao Wi-Fi e interagir com o Blynk.
3. Comandos do aplicativo foram configurados para controlar o motor (via GPIOs).

### 5. Elaboração da Programação - Passos em Tópicos

- **Configuração Inicial**
  - Configuração da IDE Arduino para ESP32.
  - Inclusão das bibliotecas: **WiFi** e **Blynk**.
- **Definição de Pinos**
  - Mapeamento dos pinos GPIO do ESP32:
    - Pinos para controle do L298N (IN1, IN2, ENA).
    - Pinos para leitura dos sensores de fim de curso.

- **Conexão Wi-Fi e Blynk**
  - Configuração do Wi-Fi no ESP32.
  - Integração do token do projeto Blynk para receber comandos do aplicativo.
- **Controle do Motor**
  - Lógica para movimento do motor em ambos os sentidos (abrir/fechar).
  - Controle de velocidade via PWM (ENA).
- **Monitoramento dos Sensores**
  - Leitura dos sensores de fim de curso conectados aos pinos digitais.
  - Interrupção do motor ao detectar os limites de movimento.
- **Interação com o App Blynk**
  - Configuração de botões no aplicativo para enviar comandos ao ESP32.
  - Resposta em tempo real do motor aos comandos recebidos.
- **Teste e Ajustes**
  - Verificação do funcionamento do motor e sensores.
  - Ajustes no código para maior confiabilidade e segurança.
- **Finalização**
  - Integração de todas as funções em um único programa.
  - Organização e comentários no código para facilitar futuras alterações.
  
- **Visão geral da ideia**
  1. **Usuário:** Interage com o sistema por meio do aplicativo Blynk, que permite controlar o hardware remotamente.
  2. **Blynk:** É uma plataforma IoT que permite a comunicação entre o usuário e o hardware. O aplicativo envia comandos e recebe feedback por meio da conexão com o ESP32.
  3. **ESP32:** Atua como o microcontrolador principal, recebendo os comandos do Blynk e controlando os demais componentes eletrônicos.
  4. **Driver Ponte H L298N:** Permite o controle do motor DC. Esse componente é usado para regular a direção e a velocidade do motor, recebendo sinais do ESP32.
  5. **Motor DC 6V com caixa de redução:** Responsável pelo movimento do sistema. A caixa de redução ajusta a velocidade e aumenta o torque.
  6. **Fonte de alimentação externa:** Fornece energia ao sistema, garantindo o funcionamento dos componentes.

Figura 1 – Visão geral do Projeto

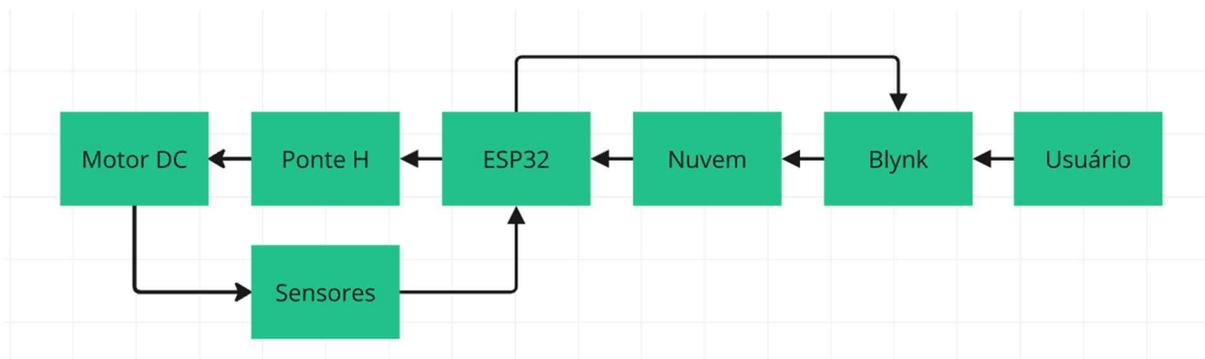


Fonte: De própria autoria, 2024

- **Diagrama em blocos**

O usuário envia comandos via aplicativo Blynk. Esses comandos são processados pelo ESP32, que controla o driver do motor (L298N). O usuário, por sua vez, aciona o motor DC para realizar os movimentos necessários. Uma fonte de alimentação externa fornece energia ao sistema.

Figura 2 – Diagrama em blocos do projeto

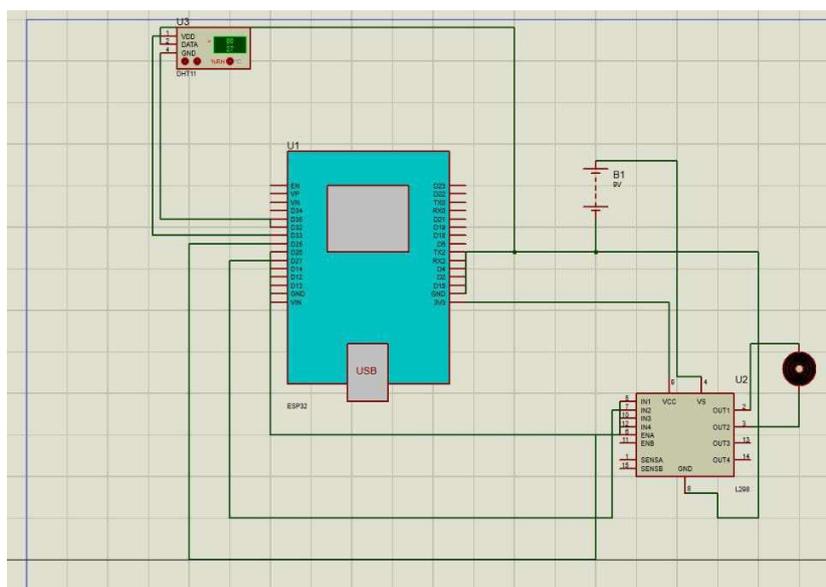


Fonte: De própria autoria, 2024

## Desenho do circuito eletrônico de cada uma das partes

O circuito elétrico do projeto integra componentes para controlar o movimento de um portão automático utilizando o ESP32, um driver de motor L298N, e sensores de fim de curso.

Figura 2 – Circuito elétrico do projeto



Fonte: De própria autoria, 2024

## Fluxograma

### 1. Descrição do Fluxograma

#### • Início

O sistema inicia com o ESP32 energizado e conectado à rede Wi-Fi, aguardando comandos de controle.

#### • Conexão com o Servidor MQTT

O ESP32 tenta estabelecer uma conexão com o broker MQTT. Caso a conexão falhe, ele tentará novamente até obter sucesso.

#### • Recebimento de Comandos

O ESP32 fica monitorando tópicos MQTT para receber mensagens do usuário, enviadas por meio de um aplicativo (Blynk ou ThingSpeak) ou dashboard.

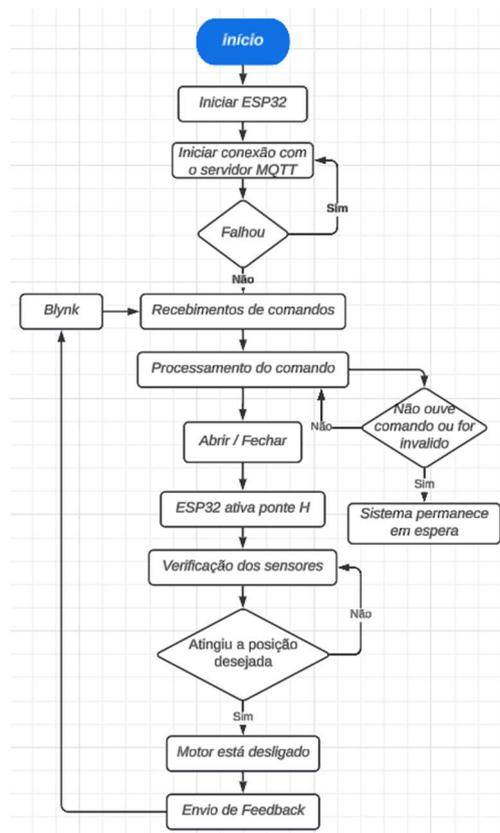
#### • Processamento do Comando

Se o comando para "Abrir", o ESP32 ativa a ponte H para movimentar o motor DC no sentido de abertura.

- Se for "Fechar", o motor é acionado no sentido oposto.

- Se não houver comando ou for inválido, o sistema permanece em espera.
- **Verificação de Sensores**  
Sensores de posição verificam se o portão está totalmente aberto ou fechado.
  - Se atingir a posição desejada, o motor está desligado.
  - Caso contrário, o sistema continua monitorando o movimento.
- **Envio de Feedback**  
O estado atual do portão

Figura 4 – Fluxograma do projeto



Fonte: De própria autoria, 2024

- **Tabela de custos**

Representação dos materiais e preços de aquisição de cada uma das etapas e componentes.

Figura 5 – Tabela de Custos do Projeto

ITEM	PREÇO
ESP32	R\$ 65,00
Driver Ponte H L298N	R\$ 26,00
Motor DC 6V com caixa de redução	R\$ 10,00
Sensor de posição	R\$ 20,00
Protoboard e Jumpers	R\$ 25,00
Fonte externa	R\$ 13,00
<b>TOTAL</b>	R\$ 159,00

Fonte: De própria autoria, 2024

- **Fluxograma da lógica de programação.**

```
#define BLYNK_TEMPLATE_ID "SEU_TEMPLATE_ID"  
#define BLYNK_DEVICE_NAME "SEU_DEVICE_NAME"  
#define BLYNK_AUTH_TOKEN "SEU_AUTH_TOKEN"
```

```
#include <WiFi.h>  
#include <BlynkSimpleEsp32.h>
```

```
// Configurações de Wi-Fi  
char ssid[] = "SEU_SSID";  
char pass[] = "SUA_SENHA";
```

```
// Pinos do ESP32  
#define MOTOR_IN1 26  
#define MOTOR_IN2 27  
#define MOTOR_PWM 25  
#define SENSOR_ABERTO 32  
#define SENSOR_FECHADO 33
```

```
// Variáveis
```

```
bool portaAberto = false; // Estado do portão
```

```
// Função para mover o motor
```

```
void moverMotor(bool abrir) {
```

```
    if (abrir) {
```

```
        digitalWrite(MOTOR_IN1, HIGH);
```

```
        digitalWrite(MOTOR_IN2, LOW);
```

```
        analogWrite(MOTOR_PWM, 200); // Velocidade do motor (0-255)
```

```
    } else {
```

```
        digitalWrite(MOTOR_IN1, LOW);
```

```
        digitalWrite(MOTOR_IN2, HIGH);
```

```
        analogWrite(MOTOR_PWM, 200);
```

```
    }
```

```
}
```

```
// Função para parar o motor
```

```
void pararMotor( ) {
```

```
    digitalWrite(MOTOR_IN1, LOW);
```

```
    digitalWrite(MOTOR_IN2, LOW);
```

```
    analogWrite(MOTOR_PWM, 0);
```

```
}
```

```
// Botão no Blynk para controle manual
```

```
BLYNK_WRITE(V1) {
```

```
    int comando = param.asInt();
```

```
    if (comando == 1 && !digitalRead(SENSOR_ABERTO)) { // abrir
```

```
        moverMotor(true);
```

```
    } else if (comando == 0 && !digitalRead(SENSOR_FECHADO)) { // fechar
```

```
        moverMotor(false);
```

```
    }
```

```
}
```

```
// Função de inicialização
void setup( ) {
    // Inicializa o Serial e Wi-Fi
    Serial.begin(115200);
    Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);

    // Configuração dos pinos
    pinMode(MOTOR_IN1, OUTPUT);
    pinMode(MOTOR_IN2, OUTPUT);
    pinMode(MOTOR_PWM, OUTPUT);
    pinMode(SENSOR_ABERTO, INPUT_PULLUP);
    pinMode(SENSOR_FECHADO, INPUT_PULLUP);

    // Inicia com o motor parado
    pararMotor();
}

// Loop principal
void loop() {
    Blynk.run();
    // Verifica os sensores de fim de curso
    if (digitalRead(SENSOR_ABERTO)) {
        pararMotor();
        portaoAberto = true;
    } else if (digitalRead(SENSOR_FECHADO)) {
        pararMotor();
        portaoAberto = false;
    }
}
```

## **Resultados e discussões**

### **Funcionamento do Sistema**

- a. O ESP32 foi capaz de se conectar à rede Wi-Fi e responder aos comandos enviados pelo aplicativo Blynk.
- b. O motor DC não conseguiu ser acionado para abrir e fechar o portão, devido à problemas ainda não solucionados, testes e estudos estão sendo feitos para solucionar o problema.

### **Detecção dos Limites**

- c. Os sensores de fim de curso não funcionaram devido ao problema ocorrido com o motor DC, entretanto seu objetivo era interromper o motor ao atingir os limites do portão, prevenindo danos mecânicos.

### **Interface com o Usuário**

- d. O aplicativo Blynk proporcionou uma interface intuitiva e responsiva para controlar o sistema, com feedback em tempo real sobre o status do portão.

### **Estabilidade do Sistema**

- e. A separação das fontes de alimentação para o motor e o ESP32 garantiu estabilidade, evitando reinicializações causadas por picos de corrente.

### **Discussões**

- a. Soluções para os devidos problemas encontrados.

## **III. CONSIDERAÇÕES FINAIS**

### **Conclusão**

O desenvolvimento do sistema automatizado para controle de portões utilizando ESP32, Blynk, motor DC e sensores de fim de curso não foi bem-sucedido, não atingindo os objetivos principais de automação, controle remoto e segurança até dado o momento.

## **Impacto do Projeto**

- Este projeto apresenta uma solução de automação de baixo custo, que pode ser implementada tanto em residências quanto em pequenas empresas.
- Ao aliar funcionalidade, conectividade e segurança, o sistema demonstra o potencial das tecnologias IoT no cotidiano.

## **Desafios Enfrentados**

- **Configuração do Blynk**
  - A integração inicial com o aplicativo exigiu ajustes para estabilizar a conexão e garantir respostas em tempo real.
- **Alimentação do Sistema**
  - Foi necessário separar as fontes de energia para o motor e o ESP32 para evitar flutuações de corrente que pudessem interferir no microcontrolador.
- **Condições Ambientais**
  - Embora o sistema tenha sido criado originalmente para funcionar em ambiente controlado, proteções adicionais seriam necessárias para operação em condições externas adversas.

## **Abstract:**

### **Gate automation with Cloud and ESP32**

This course completion work presents development a gate automation system using the ESP32 microcontroller and cloud computing technology. The main objective is to provide a safe, efficient and low-cost solution for remote control of residential and commercial gates, allowing them to be opened and closed using mobile devices connected to the internet.

Gate automation traditionally uses RF (Radio Frequency) remote controls, which may present limitations in terms of range and safety. With the implementation of a cloud-based system, it is possible to overcome these limitations, providing broader and safer control, in addition to enabling the monitoring and operation of the gate from anywhere with internet access.

## Keywords:

Gate automation; esp32; cloud computing; IoT.

## REFERÊNCIAS BIBLIOGRÁFICAS

**AOSONG ELECTRONICS CO.** DHT22 DATASHEET. **SparkFun**. Disponível em: <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>. Acesso em: 19 jun. 2024.

**ARDUINO.** DOCUMENTAÇÃO – REFERÊNCIA. **Arduino.cc**. Disponível em: <https://www.arduino.cc/reference/en/>. Acesso em: 12 jun. 2024.

**CORPO DE BOMBEIROS DO ESTADO DO CEARÁ.** A prevenção contra incêndios no Brasil. **Governo do Estado do Ceará**. 2023. Disponível em: <https://www.bombeiros.ce.gov.br/2023/07/15/a-prevencao-contra-incendios-no-brasil/>. Acesso em: 14 nov. 2024.

**ESPRESSIF.** ESP32 DATASHEET. **Espressif**. Disponível em: [https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf). Acesso em: 12 jun. 2024.

**FOLHA BV.** Loja no Cinturão Verde tem princípio de incêndio em quadro de energia; veja vídeo. **Folha BV**. 2023. Disponível em: <https://www.folhabv.com.br/policia/loja-no-cinturao-verde-tem-principio-de-incendio-em-quadro-de-energia-veja-video/>. Acesso em: 14 nov. 2024.

**HANWEI ELECTRONICS CO.** MQ-9 DATASHEET. **Electronicos Caldas**. Disponível em: [https://www.electronicoscaldas.com/datasheet/MQ-9\\_Hanwei.pdf](https://www.electronicoscaldas.com/datasheet/MQ-9_Hanwei.pdf). Acesso em: 19 jun. 2024.

**INSTITUTO SPRINKLER BRASIL.** NOTÍCIA – REFERÊNCIA. **Instituto Sprinkler Brasil**. Disponível em: <https://sprinklerbrasil.org.br/imprensa/principio-de-incendio-atinge-upa-e-pacientes-sao-transferidos-na-pb/>. Acesso em: 29 ago. 2024.

**MUNDODAELETRICA.** HISTÓRIA DOS DISJUNTORES - Disponível em: <https://www.mundodaeletrica.com.br/conheca-a-historia-dos-disjuntores/> Acesso em: 26/11/2024

**RIOMAFRAMIX.** Vazamento em botijão de gás causa princípio de incêndio e assusta moradores. **Riomaframix**. 2023. Disponível em:

<https://www.riomaframix.com.br/policial/vazamento-em-botijao-de-gas-causa-principio-de-incendio-e-assusta-moradores/>. Acesso em: 14 nov. 2024.

**SIEMENS.** Quadro de distribuição: o que é, como funciona e como instalar.

**Siemens Brasil.** 2023. Disponível em:

<https://www.siemens.com/br/pt/empresa/stories/energia/quadro-de-distribuicao-o-que-e-como-funciona-e-como-instalar.html>. Acesso em: 14 nov. 2024.