

**UNIVERSIDADE DO ESTADO DO RIO DE JANEIRO**  
**INSTITUTO POLITÉCNICO**  
**ENGENHARIA DE COMPUTAÇÃO**

**PROJETO DE AUTOMAÇÃO COM A UTILIZAÇÃO DE RFID PARA CONTROLE DE  
PRESENÇA E DE ALMOXARIFADO**

Aluno: Vicente Brito Soares

Graduando do curso de Engenharia de Computação do IPRJ/UERJ

Nova Friburgo, 11 de janeiro de 2024

## SUMÁRIO

<b>ENGENHARIA DE COMPUTAÇÃO.....</b>	<b>1</b>
<b>1.INTRODUÇÃO.....</b>	<b>10</b>
<b>2. METODOLOGIA.....</b>	<b>11</b>
<b>2.1 ARDUINO .....</b>	<b>11</b>
<b>2.2 MÓDULO RELÉ.....</b>	<b>11</b>
<b>2.3 ESP 32.....</b>	<b>11</b>
<b>2.4 MFRC522 .....</b>	<b>12</b>
<b>2.5 TELA LCD 16X2 I2c.....</b>	<b>12</b>
<b>2.6 PROJETO .....</b>	<b>13</b>
<b>2.7 CUSTO.....</b>	<b>13</b>
<b>3.0 INSTALAÇÃO.....</b>	<b>13</b>
<b>4. APENDICE .....</b>	<b>15</b>
<b>4.1 CIRCUITO LÓGICO EASYEDA .....</b>	<b>15</b>
<b>4.2 CÓDIGO.....</b>	<b>17</b>

## 1.INTRODUÇÃO

Uma das tecnologias de controle e segurança mais difundidas nos dias de hoje é o sistema de Identificação por Radiofrequência (RFID). Apesar de muitas vezes passar despercebido, o RFID está presente em diversas situações do cotidiano, como em cartões de transporte público, crachás de ponto em ambientes de trabalho, e dispositivos de segurança que encontramos em lojas de roupas. O motivo de sua popularidade reside na combinação de eficiência, praticidade e baixo custo, características que tornam essa tecnologia ideal para uma ampla gama de aplicações.

Diante dessa realidade, foi desenvolvido um dispositivo inovador que busca integrar todas essas funcionalidades em um único sistema, de maneira eficiente e acessível. O projeto tem como objetivo centralizar o controle e a segurança dentro de uma empresa, utilizando a tecnologia RFID. Com essa solução, cada funcionário poderá, por exemplo, registrar sua entrada e saída, acessar materiais de trabalho e transitar livremente pelos diferentes setores da empresa, tudo isso sem a necessidade de chaves físicas ou múltiplos dispositivos de acesso.

O projeto é composto por componentes eletrônicos cuidadosamente selecionados para garantir o desempenho e a funcionalidade desejados. A base do sistema é um microcontrolador ESP32, que oferece capacidade de processamento e conectividade. O módulo sensor RFID MFRC522 é responsável pela leitura dos cartões RFID, enquanto um módulo relé permite a ativação de dispositivos externos, como fechaduras eletrônicas. Além disso, uma tela LCD I2C 16x2 fornece uma interface visual simples e eficaz para o usuário.

A integração com um banco de dados é feita por meio da tecnologia WebSocket, permitindo uma comunicação rápida e segura entre o dispositivo e o servidor. Isso facilita o gerenciamento centralizado das informações e o monitoramento em tempo real das atividades, proporcionando um controle ainda mais preciso e seguro sobre o acesso e a movimentação dos funcionários dentro da empresa.

## 2. METODOLOGIA

### 2.1 ARDUINO

O Arduino é uma placa de prototipagem eletrônica que permite o desenvolvimento de projetos de automação. Ela conta com a tecnologia *Open Source*, ou seja, tem seu código aberto, o que permite o acesso por qualquer pessoa. Ele também conta com uma IDE (*Integrated Development Environment*) para o desenvolvimento do software que será inserido na placa para realizar as atividades programadas.

### 2.2 MÓDULO RELÉ

O módulo Relé consiste em um interruptor eletromagnético que é usado para ligar e desligar um circuito por um sinal elétrico de baixa potência, ou onde vários circuitos devem ser controlados por um único sinal. Os relés são nomeados com designação, para o caso do presente trabalho usa-se o relé *Single Double Pole Throw* (SPDT). Esse possui um total de cinco terminais, destes dois são os terminais da bobina e um terminal comum também está incluído, que se conecta a qualquer um dos outros dois.

### 2.3 ESP 32

A ESP32 é um microcontrolador de baixo custo e alto desempenho desenvolvido pela Espressif Systems. Ele combina um processador dual-core Tensilica Xtensa LX6, conectividade Wi-Fi e Bluetooth integradas, além de uma ampla gama de periféricos e interfaces, como GPIOs, UART, SPI, I2C e ADC. A ESP32 é amplamente utilizada em projetos IoT devido à sua capacidade de conectar dispositivos à internet de maneira eficiente e econômica. Possui suporte robusto à programação em ambiente Arduino IDE, MicroPython e outras linguagens, facilitando o desenvolvimento de aplicações para diferentes necessidades. Sua flexibilidade, baixo consumo de energia e preço acessível a tornam uma escolha popular para projetos desde simples sensores até soluções complexas de automação residencial e industrial.



Imagem: Esp 32

## 2.4 MFRC522

O módulo sensor RFID MFRC522 operando na frequência de 13,56 MHz é amplamente utilizado para leitura e gravação de tags RFID (Radio Frequency Identification). Ele é baseado no chip MFRC522 da NXP, oferecendo uma interface SPI para comunicação com microcontroladores como Arduino e Raspberry Pi. O MFRC522 suporta diferentes modos de operação RFID, incluindo leitura e escrita de tags passivas, o que o torna ideal para aplicações de controle de acesso, inventário e identificação. Compatível com uma variedade de bibliotecas e plataformas de desenvolvimento, o MFRC522 simplifica a integração de funcionalidades RFID em projetos IoT e sistemas de segurança. Sua simplicidade de uso, baixo consumo de energia e custo acessível o tornam popular em diversos cenários de automação e monitoramento.

## 2.5 TELA LCD 16X2 I2C

A tela LCD 16x2 com interface I2C é um componente popular em projetos de eletrônica e microcontroladores. Ela consiste em um display de cristal líquido capaz de mostrar duas linhas com até 16 caracteres cada, facilitando a exibição de informações de forma clara e legível. A interface I2C permite uma conexão simplificada com microcontroladores como Arduino, ESP32 e Raspberry Pi, utilizando apenas dois pinos para comunicação (SDA e SCL). Isso reduz a quantidade de fios necessários e simplifica a integração em projetos compactos. A tela LCD 16x2 I2C é amplamente utilizada em aplicações que exigem a exibição de dados como sensores,

status do sistema, mensagens de usuário e muito mais, tornando-se uma escolha popular devido à sua simplicidade de uso e eficiência.

## **2.6 PROJETO**

O projeto consiste em um microcontrolador ESP32, ligado a um módulo MFRC522, um módulo Relé e uma tela LCD 16X2 com interface I2c, a caixa do projeto foi toda modelada no SketchUp e feita em uma impressora 3d utilizando PLA como filamento. Sua funcionalidade envolve conectar em uma rede de internet selecionada no código, depois coletar a RFID posicionada no local indicado, os bytes recebidos pela TAG, será enviado para um banco de dados, esse banco de dados irá procurar em seu sistema se aquela TAG está cadastrada e se ela é autorizada pelo sistema e irá devolver essas informações ao aparelho, que irá mostrar na tela LCD as informações recebidas e caso aquela TAG for autorizada, ela acionará o Relé que estará conectada a uma catraca ou portão.

## **2.7 CUSTO**

O custo foi um dos principais fatores para criação desse projeto, os sensores utilizados em sua montagem possuem uma boa eficiência e um baixo custo se comparado aos presentes no mercado, sendo o microcontrolador na faixa de 24 reais, o sensor RFID por 14, além da caixa ser feita em uma impressora 3d utilizando PLA como filamento. Ao todo o projeto fica na faixa de 100/110 para ser construído.

## **3 INSTALAÇÃO**

Para fazer a instalação, apenas algumas mudanças são necessárias, a primeira é alterar no código qual o nome e a senha da rede de internet que será utilizada, após isso ele já estará em operação, para conectar a catraca, basta que pegue os fios na parte direita da caixa e conectá-lo na catraca, respeitando os fios de energia e o aterramento. Todo o sistema é modular, caso aconteça de algum componente falhar seja por tempo ou por uso, sua troca é fácil, basta removê-lo dos pinos e colocar um novo de mesmo modelo nos mesmos pinos e montar o aparelho de volta.

## 4 REQUISITOS PARA FUNCIONAMENTO

### 4.1 REQUISITOS ESP32

#### Requisitos de Alimentação

- **Tensão de Operação (Vin)**
  - **Mínima:** 2.7V
  - **Recomendada:** 3.3V
  - **Máxima:** 3.6V
- **Corrente de Operação**
  - **Modo de Transmissão Wi-Fi:** até 240 mA (dependendo da intensidade do sinal)
  - **Modo de Transmissão Bluetooth:** até 140 mA
  - **Consumo em modo de baixa energia (Deep Sleep):** < 10  $\mu$ A

#### Especificações Ambientais

- **Temperatura de Operação**
  - **Mínima:** -40°C
  - **Máxima:** 85°C

#### Interfaces e Pinos

- **GPIO (General Purpose Input/Output)**
  - Capacidade de corrente: até 12 mA por pino
  - Tensão máxima de entrada nos pinos: 3.6V
- **ADC (Analog to Digital Converter)**
  - Resolução: 12 bits
  - Tensão de referência: 0V a 3.3V
- **DAC (Digital to Analog Converter)**
  - Resolução: 8 bits
- **Comunicação**
  - **Wi-Fi:** IEEE 802.11 b/g/n
  - **Bluetooth:** BLE (Bluetooth Low Energy) e Bluetooth Classic

#### Considerações Especiais

- **Regulador de Tensão:** Se a tensão de alimentação for maior que 3.6V, um regulador de tensão é necessário para baixar a tensão para a faixa operacional segura.
- **Capacitores de Desacoplamento:** Recomenda-se o uso de capacitores de desacoplamento próximos aos pinos de alimentação para estabilizar a voltagem.

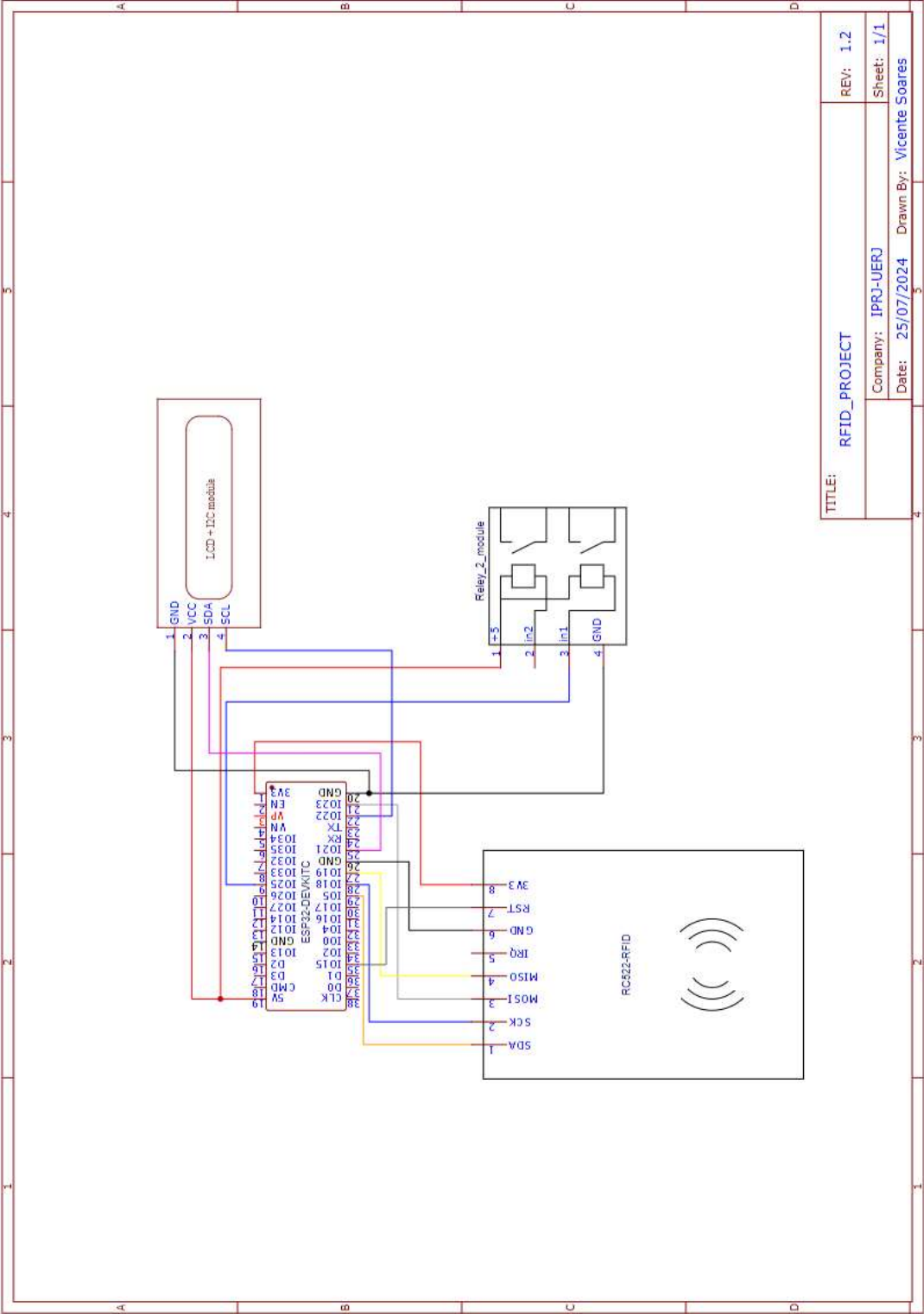
## 4.2 REQUISITOS CATRACA/ RELÉ

- **Valor de saída:** 110/220 V
- Fazer a conversão para a voltagem desejada, normalmente catracas e portões usam 12V como padrão
- **Fio Vermelho:** Vin
- **Fio Azul:** Terra

## 5. APENDICE

### 5.1 CIRCUITO LÓGICO EASYEDA





## 5.2 CÓDIGO

```
#include <SPI.h>

#include <MFRC522.h>

#include <WiFi.h>

#include <Wire.h>

#include <LiquidCrystal_I2C.h>


// Definição dos pinos para o ESP32

#define SS_PIN 5 // Pino do ESP32 conectado ao pino SDA do MFRC522
#define RST_PIN 15 // Pino do ESP32 conectado ao pino RST do MFRC522
MFRC522 mfrc522(SS_PIN, RST_PIN); // Cria o objeto MFRC522


// Configurações de rede WiFi

const char* ssid = "Vicente Wifi";

const char* password = "98719885";

const char* server = "www.geaf-web.com.br"; // Nome do host

const char* resource = "/receber_rfi"; // Caminho do recurso

const int porta = 80; // Porta do servidor HTTP

WiFiClient client;


const int relePin = 25; // Pino do relé
```

```
// Inicialização do display LCD I2C
LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
    Serial.begin(115200);    // Inicializa a comunicação serial
    SPI.begin();            // Inicializa a interface SPI
    mfrc522.PCD_Init();     // Inicializa o leitor RFID
    pinMode(relePin, OUTPUT); // Configura o pino do relé como saída
    // Inicialização do display LCD
    lcd.init();
    lcd.backlight(); // Ativa a luz de fundo do LCD
    lcd.setCursor(0, 0);
    lcd.print("Conectando...");

    WiFi.begin(ssid, password); // Conecta-se à rede WiFi
    while (WiFi.status() != WL_CONNECTED) { // Aguarda a conexão
        delay(500);
        Serial.println("Conectando ao WiFi...");
    }
    Serial.println("Conectado ao WiFi!");
    Serial.print("Endereço IP: ");
    Serial.println(WiFi.localIP());
}

void loop() {
    lcd.setCursor(0, 0);
    lcd.print("Posicione");
```

```

lcd.setCursor(0, 1);

lcd.print("o Cartao");


// Verifica se há novas tags RFID presentes

if (mfrc522.PICC_IsNewCardPresent() && mfrc522.PICC_ReadCardSerial()) {

    String rfidData = ""; // Armazena o código RFID lido


    for (byte i = 0; i < mfrc522.uid.size; i++) {

        rfidData.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? "0" : ""));

        rfidData.concat(String(mfrc522.uid.uidByte[i], HEX));

    }

    Serial.println("RFID lido: " + rfidData);


    // Exibe a mensagem no LCD

    lcd.clear();

    lcd.setCursor(0, 0);

    lcd.print("Consultando...");


    if (client.connect(server, porta)) {

        // Envia os dados RFID e o endereço IP local para o servidor HTTP

        String requestData = String("GET ") + resource + "/" + rfidData + "?ip=" + WiFi.localIP().toString() +
        " HTTP/1.1\r\n" +

            "Host: " + server + "\r\n" +

            "Connection: close\r\n\r\n";

        client.print(requestData);

        Serial.println("RFID e endereço IP enviados para o servidor");


        // Aguarda a resposta do servidor

```

```
String response = "";

while (client.connected() && !client.available()) {

    delay(1); // Aguarda até que os dados estejam disponíveis
}

while (client.connected() || client.available()) {

    if (client.available()) {

        response += client.readStringUntil('\r');

    }

}

// Encontra o índice do início do corpo da resposta
int bodyStartIndex = response.indexOf("\r\n\r\n") + 4;

// Extrai o corpo da resposta
String responseBody = response.substring(bodyStartIndex);

// Verifica se a mensagem recebida contém "AUTORIZADO" ou "NEGADO"
int autorizadoIndex = responseBody.indexOf("AUTORIZADO");
int negadoIndex = responseBody.indexOf("NEGADO");

if (autorizadoIndex != -1) {

    // Extrai o nome e sobrenome da pessoa autorizada

    int nomeEndIndex = responseBody.lastIndexOf(' ', autorizadoIndex - 1);
    int nomeStartIndex = responseBody.lastIndexOf(' ', nomeEndIndex - 1);

    String nomeAutorizado = responseBody.substring(nomeStartIndex + 8, nomeEndIndex);

    String sobrenomeAutorizado = responseBody.substring(nomeEndIndex + 1,
responseBody.indexOf("-", nomeEndIndex + 1));
```

```

// Extrai o texto "ASO VENC 11/05" após a palavra "AUTORIZADO"

int asoIndex = responseBody.indexOf("ASO VENC", autorizadoIndex);

String textoASO = responseBody.substring(asoIndex);

// Exibe o nome da pessoa autorizada e "ASO VENC 11/05"

Serial.println("Acesso autorizado para: " + nomeAutorizado);

lcd.clear();

lcd.setCursor(0, 0);

lcd.print(nomeAutorizado);

lcd.print(" ");

lcd.print(sobrenomeAutorizado);

lcd.setCursor(0, 1);

lcd.print(textoASO);

delay(2000); // Aguarda 2 segundos

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Autorizado");

digitalWrite(relePin, HIGH); // Ativa o relé

delay(2000);           // Mantém o relé ativado por 2 segundos

digitalWrite(relePin, LOW); // Desativa o relé

lcd.clear();

} else if (negadoIndex != -1) {

// Extrai o nome da pessoa antes da palavra "NEGADO"

int nomeStartIndex = responseBody.indexOf("-") + 1;

int nomeEndIndex = responseBody.indexOf("-", nomeStartIndex);

String nomeNegado = responseBody.substring(nomeStartIndex, nomeEndIndex);

```

```
Serial.println("Acesso negado para: " + nomeNegado);

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Acesso Negado:");

lcd.setCursor(0, 1);

lcd.print("Procurar RH");

delay(1000);

lcd.clear();

} else {

    Serial.println("Acesso negado (sem resposta do servidor)");

    lcd.clear();

    lcd.setCursor(0, 0);

    lcd.print("Acesso Negado:");

    lcd.setCursor(0, 1);

    lcd.print("nao encontrada");

    delay(1000);

    lcd.clear();

}

} else {

    Serial.println("Falha ao conectar-se ao servidor");

    lcd.clear();

    lcd.setCursor(0, 0);

    lcd.print("Erro!");

}

delay(1000); // Aguarda um pouco antes de ler o próximo cartão

}

}
```