

Sistema embarcado RFID para controle IoT de identidade em diferentes níveis de acesso

José de Alencar de Sousa Júnior, Sandro César Silveira Jucá

Departamento de Telemática – Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)

CEP 60040-531 – Fortaleza – CE – Brasil

alenior@gmail.com, sandro.juca@gmail.com

17 de fevereiro de 2025

Resumo

Abstract. This article aims to promote access control with face-to-face identity recognition and online activation based on radio frequency identification (RFID) technology, sending information to an administrator user via a messaging application. The entire identification and registration process is done via the Esp32 microcontroller, which, because it has internet communication, updates the information sent online. All information about the tag's actions is displayed on an LCD display connected to the microcontroller board, and each action is signaled by LEDs and a specific sound signal.

Resumo. Este artigo tem o objetivo de promover controle de acesso com distinção de identidade presencial e acionamento online baseado na tecnologia de identificação por radiofrequência (RFID), enviando informações para um usuário administrador por meio de aplicativo de mensagem. Todo o processo de identificação e registro é feito por meio do microcontrolador Esp32 que por ter comunicação com a internet atualiza as informações enviadas de forma online. Todas as informações sobre as ações da tag são exibidas em um display LCD ligado à placa microcontrolada e cada ação é sinalizada por leds e um sinal sonoro específico.

1 Introdução

A fim de almejar melhores resultados nas suas atividades profissionais e particulares, a sociedade humana busca constantemente melhorar suas rotinas, em especial quanto a controles e suas informações associadas. Neste contexto, a Identificação por Radiofrequência (RFID) “é uma daquelas raras tecnologias que ‘mudam o mundo’, que forçarão a uma reconsideração de muitas estratégias na cadeia de valores” [Glover 2015, apud Jucá 2016].

A identificação por radiofrequência já tem aplicações difundidas, como sistema de segurança de lojas e monitoramento logístico de itens a ser transportados. No entanto, muitas outras aplicações podem ser feitas, graças ao seu baixo custo e praticidade de implantação, além de ser aplicável a seres humanos, animais, objetos diversos, além de veículos de toda espécie [UFRJ RFID].

A partir desse cenário, foi desenvolvido um sistema de controle de acesso baseado na tecnologia de radiofrequência, onde cada usuário autorizado possui uma tag para permitir acesso ao ambiente controlado, podendo ainda ser definido um nível adicional de segurança por meio da digitação de senha para autorização do acesso. É amplamente reconhecida a eficiência da identificação por radiofrequência na promoção de maior controle e segurança a determinados ambientes e pessoas.[Nascimento 2015]

Ao identificar a tag, o acesso pode ser feito (com ou sem a digitação adicional de senha, conforme nível de acesso do usuário identificado). Adicionalmente, o administrador é informado sobre as tentativas de acesso e seus sucessos (ou não) por meio de mensagem enviada por aplicativo (WhatsApp), no instante em que ocorre.

2 Materiais e métodos

Serão utilizados neste trabalho os seguintes componentes: microcontrolador Esp32, Fonte de alimentação 5v, 2 placas protoboard de 830 furos, Variedade de cabos Dupont Jumper (Macho

x Macho e Macho x Fêmea), Módulo relé 5v de 1 canal, Display LCD 16x02, Buzzer passivo 5v, Leds verde, amarelo e vermelho, Fonte 12v, Tranca solenóide 12v e Sensor RFID RC522. Suas imagens são apresentadas nas figuras que seguem (Figuras 1 a 9).

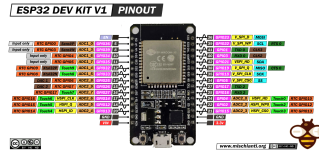


Figura 1: Ilustração e pinagem do Esp32. Disponível em: <https://mischianti.org/doit-esp32-dev-kit-v1-high-resolution-pinout-and-specs/>

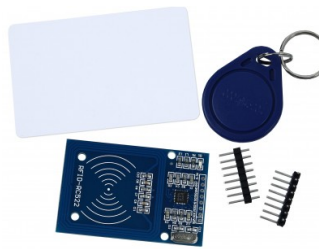


Figura 2: Sensor RFID RC522. Disponível em: <https://www.usinainfo.com.br/rfid-arduino-e-ibutton/kit-rc522-leitor-rfid-tags-chaveiro-cartao-2582.html>



Figura 3: Display LCD 16x02. Disponível em: <https://www.amazon.com.br/Display-M%C3%B3dulo-Soldado-Arduino-raspberry/dp/B0CDHZVV4Z>



Figura 4: Teclado matricial 4x3. Disponível em: <https://www.pontodaeletronica.com.br/teclado-matricial-4x3-para-arduino.html>



Figura 5: Tranca solenóide 12v. Disponível em: <https://www.usinainfo.com.br/mini-fechadura-eletrica-solenoid/fechadura-eletrica-solenoid-12v-nf-fe-91-lingueta-reversivel-3524.html>



Figura 6: Módulo relé 5v de 1 canal. Disponível em: <https://loja.maisrobotic.com.br/shields-e-modulos/64-modulo-rele-1-canal-5v.html>

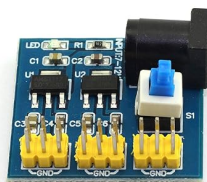


Figura 7: Fonte 12v. Disponível em: <https://www.amazon.com.br/DGZZI-Convertor-alimenta%C3%A7%C3%A3o-step-down-multi-sa%C3%ADda/dp/B07RKMZY8S>



Figura 8: Fonte 5v. Disponível em: <https://www.amazon.com.br/Fonte-Carregador-Celular-Tomada-Bivolt/dp/B0BN2YHWKY>

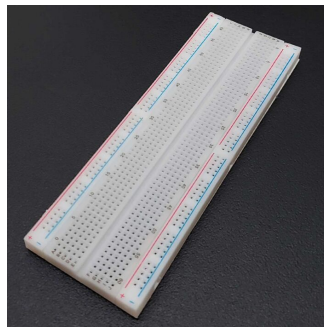


Figura 9: Placas protoboards 830 furos. Disponível em: <https://www.mamuteeletronica.com.br/protoboard-830-pontos-7978>



Figura 10: Cabos dupont jumper. Disponível em: <https://www.autocorerobotica.com.br/cabos-dupont-jumper-fxf-30cm-pacote-com-40pcs>

O sensor **RFID RC522** é uma tecnologia bastante conhecida dentre os projetistas de circuitos eletrônicos, assim como utilizada pelos usuários dos sistemas implementados com o mesmo, podendo ser citados como exemplos: um crachá para acesso a um determinado ambiente restrito, uma etiqueta para acompanhamento de um determinado objeto num processo logístico, ou o monitoramento de animais em ambiente selvagem para estudos sobre o risco de extinção.

Conforme [Araujo 2014, apud Washington 2021], a RFID é um leitor de radiofrequência capaz de realizar a leitura de até 200 etiquetas (TAGs) por segundo, sem a necessidade de estar visivelmente perto, sendo composta por dois elementos complementares: um leitor construído de uma antena (transceptor), que faz a leitura do sinal e transfere a informação para o dispositivo desejado, e uma etiqueta de radiofrequência (RF) que contém a informação a ser transmitida.

2.1 Funcionamento do sistema RFID

Segundo detalhamento do datasheet do sensor RC522 [Alldatasheet RC522], a tecnologia de identificação por radiofrequência consiste em se utilizar um microchip ligado a uma antena,

operando tanto em baixas como altas frequências. Esse microchip consiste num transponder que não necessita de fonte de alimentação, pois o sinal que o excita vem diretamente de um circuito de leitura/gravação. Ao ser excitado, o circuito é alimentado enviando ou recebendo dados que estejam gravados. Um dos princípios de funcionamento da tecnologia RFID é a radiação eletromagnética, que é definida como sendo o transporte de energia por meio de flutuações dos campos elétrico e magnético. A luz, ou radiação eletromagnética, pode ser observada sob diferentes formas ou seja, em diferentes faixas espectrais: visível, infravermelho, ultravioleta, ondas rádio, etc. [Unesp 2025]. Uma parte pequena do campo emissor interage com a bobina da antena do transponder, que está a uma determinada distância da bobina do leitor. Pela indução magnética, uma tensão é gerada na bobina da antena do transponder. Esta tensão é retificada e serve como a fonte de alimentação para o microchip. Um capacitor é conectado paralelamente à bobina da antena do leitor. A capacitância é selecionada de forma a combinar com a indutância da bobina da antena para dar forma a um circuito ressonante paralelo, ou seja, para se obter uma frequência ressonante que corresponda com a frequência da transmissão do leitor, a ilustração é mostrada na Figura X.

2.2 Sistema de ativação online

Para a ativação online do sistema, foi utilizado o serviço do sítio Blynk (<https://blynk.io>), que oferece plataforma intuitiva e amigável para cadastramento de um microcontrolador (neste caso, Esp32; mas podem ser outros, como Esp8266 e Arduino) e aplicativo disponível para Android e Ios que permite a ativação online do sistema, portanto, via smartphone e navegador de internet.

Para tanto, é necessário fazer um cadastro simplificado de usuário, assim como informar os detalhes do sistema a ser criado, como: microcontrolador utilizado, forma de acesso à internet, ide utilizada para desenvolvimento etc. Após esta preparação, um painel é disponibilizado tanto no navegador quanto na aplicação do smartphone, permitindo a integração com o projeto desenvolvido por meio do código lógico implantado no microcontrolador.

2.3 Sistema de envio de mensagens

Para monitoramento administrativo e de segurança do sistema desenvolvido, foi utilizado o serviço do sítio CallMeBot (callmebot.com), que oferece serviço de fácil e prática implementação de envio de mensagens automático, permitindo que se possa acompanhar em tempo real as ações envolvidas ao sistema de controle de acesso desenvolvido.

Para se utilizar do serviço da api, faz-se um cadastramento simplificado utilizado-se um número de celular ativo (também cadastrado e em uso num aplicativo mensageiro, como WhatsApp ou Telegram) que dá acesso a uma "apikey" correspondente. Após esta preparação, é disponibilizado um link personalizado ao número e seu "apikey", permitindo promover envios automáticos de mensagens para o número informado.

3 Contrução do circuito

O sistema completo consiste de um controle de acesso RFID que permite o acesso a um determinado ambiente restrito mediante respectiva identificação e nível de acesso: um leitor de RFID fica posicionado ao lado da porta trancada, assim como um teclado para informar senha, quando aplicável. Sinais sonoros são emitidos para auxiliar o usuário na percepção dos resultados de sua interação com o sistema, assim como sinais visuais emitidos por leds nas cores vermelha, amarela e verde. Adicionalmente, mas não menos importante, um display lcd informa sucintamente instruções diversas para as interações com o sistema, assim como informa consequências resultantes dessas interações.

Caso uma tag seja aproximada, é exigido que se informe a senha de acesso para destrancar a porta. Caso trate-se de uma tag máster, a senha não é exigida para liberar o acesso. Para efetivar o destrancamento da porta protegida, ao ser liberado o acesso o microcontrolador envia um sinal para um relé que atua abrindo a tranca solenóide instalada. No caso de ativamento remoto, o display lcd detalha o mesmo, e também atuam-se os sinais sonoros e visuais instalados.

4 Resultados

O protótipo foi montado com os componentes previstos e funcionou como esperado. A Figura 11 demonstra o aspecto amplo do protótipo montado, pronto para uso/demonstração.

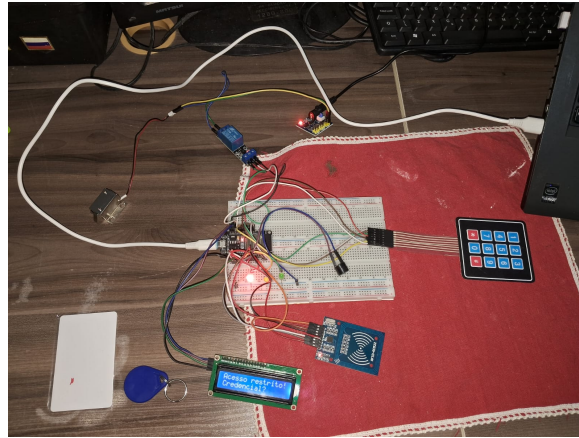


Figura 11: Imagem do protótipo montado.

Baseado no princípio de Internet das coisas (IoT – Internet of Things), o sistema pode ser acessado via internet pela plataforma Blynk no navegador ou via aplicativo de smartphone, conforme as Figuras 12 e 13 a seguir:

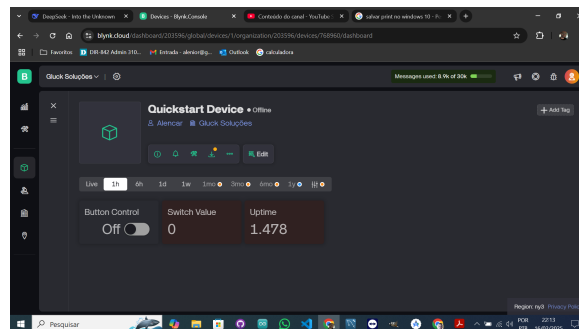


Figura 12: Utilizando a plataforma Blynk via navegador de internet.

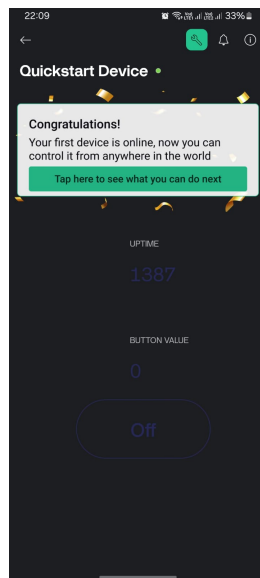


Figura 13: Utilizando a plataforma Blynk via aplicativo em Android.

Foi ainda utilizada a api do site callmebot.com para o recebimento de notificações do sistema (Figura 13).

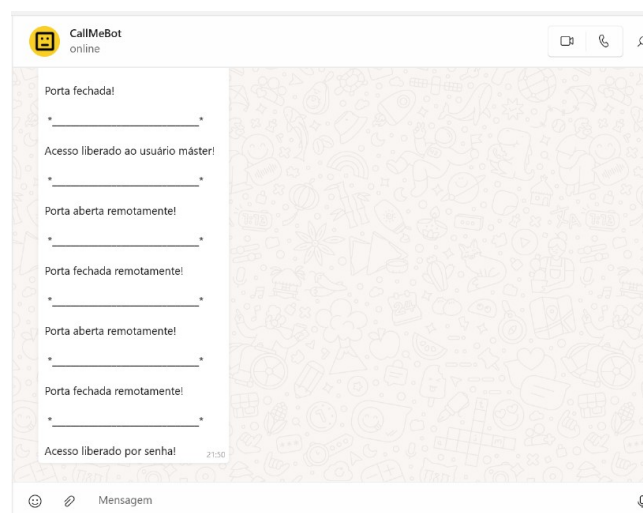
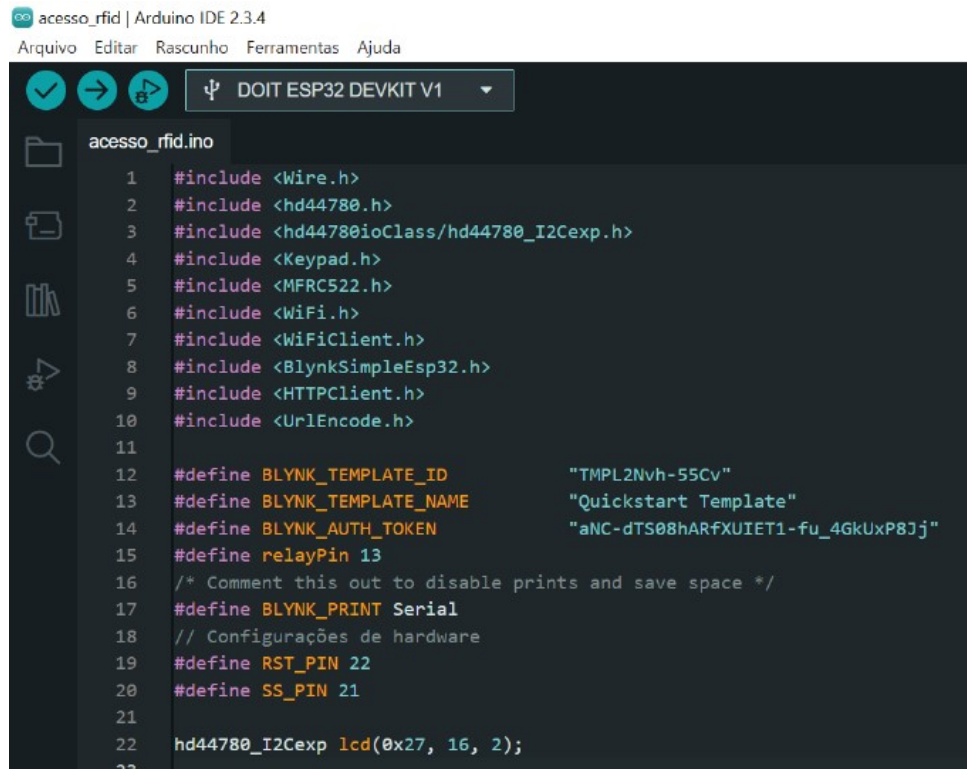


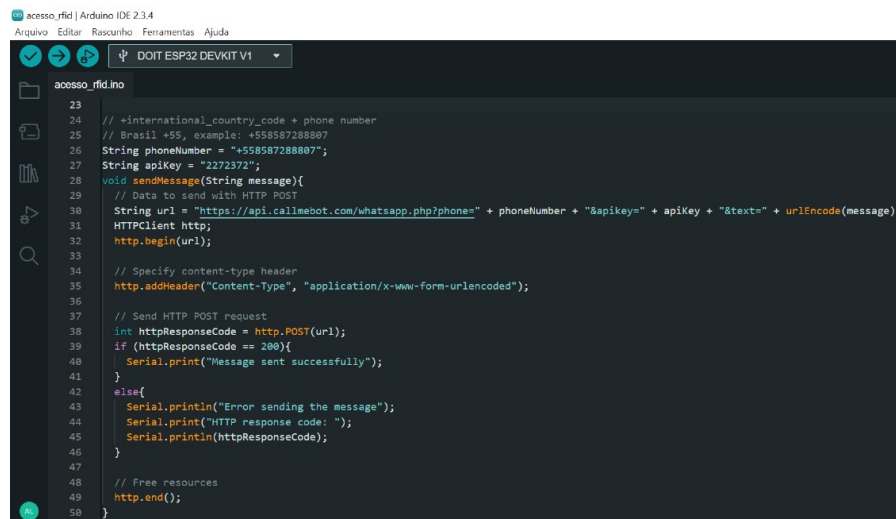
Figura 14: Recebendo notificações do sistema via aplicativo de mensagens WhatsApp.

O código completo do projeto desenvolvido na IDE Arduino ficou como segue (Figuras 15 a 27):



```
acesso_rfid.ino
1  #include <Wire.h>
2  #include <hd44780.h>
3  #include <hd44780ioClass/hd44780_I2Cexp.h>
4  #include <Keypad.h>
5  #include <MFRC522.h>
6  #include <WiFi.h>
7  #include <WiFiClient.h>
8  #include <BlynkSimpleEsp32.h>
9  #include <HTTPClient.h>
10 #include <URLEncoder.h>
11
12 #define BLYNK_TEMPLATE_ID      "TMPL2Nvh-55Cv"
13 #define BLYNK_TEMPLATE_NAME    "Quickstart Template"
14 #define BLYNK_AUTH_TOKEN      "aNC-dTS08hARfXUIET1-fu_4GkUxP8Jj"
15 #define relayPin 13
16 /* Comment this out to disable prints and save space */
17 #define BLYNK_PRINT Serial
18 // Configurações de hardware
19 #define RST_PIN 22
20 #define SS_PIN 21
21
22 hd44780_I2Cexp lcd(0x27, 16, 2);
23
```

Figura 15: Código desenvolvido para o projeto (1/13).



```
acesso_rfid.ino
23
24 // +international_country_code + phone number
25 // Brasil +55, example: +558587288807
26 String phoneNumber = "+558587288807";
27 String apiKey = "2272372";
28 void sendMessage(String message){
29   // Data to send with HTTP POST
30   String url = "https://api.callmebot.com/whatsapp.php?phone=" + phoneNumber + "&apikey=" + apiKey + "&text=" + URLEncoder(message);
31   HTTPClient http;
32   http.begin(url);
33
34   // Specify content-type header
35   http.addHeader("Content-Type", "application/x-www-form-urlencoded");
36
37   // Send HTTP POST request
38   int httpResponseCode = http.POST(url);
39   if (httpResponseCode == 200){
40     Serial.print("Message sent successfully");
41   }
42   else{
43     Serial.println("Error sending the message");
44     Serial.print("HTTP response code: ");
45     Serial.println(httpResponseCode);
46   }
47
48   // Free resources
49   http.end();
50 }
```

Figura 16: Código desenvolvido para o projeto (2/13).


```

acesso_rfido.ino
51 // Your WiFi credentials: Set password to "" for open networks.
52 // char ssid[] = "Alencar's Galaxy M14 5G";
53 char ssid[] = "GCNET-Alencar";
54 char pass[] = "11223344";
55 BlynkTimer timer;
56 // Variáveis globais para controle do temporizador
57 unsigned long tempoAnterior = 0; // Armazena o tempo inicial
58 bool relayAtivado = false; // Indica se o relé foi ativado
59 // This function is called every time the Virtual Pin 0 state changes (DA PLATAFORMA BLYNK PARA A PLACA ESP32)
60 BLYNK_WRITE(V0) {
61 // Verifica se o valor recebido é 1 (relé ativado)
62 if (param.asInt() == 1) {
63     digitalWrite(relayPin, HIGH); // Ativa o relé
64     lcd.clear();
65     lcd.noCursor();
66     lcd.print("Acesso liberado");
67     lcd.setCursor(0, 1);
68     lcd.print("remotamente!");
69     delay(2000);
70     lcd.clear();
71     lcd.print("Acesso restrito!");
72     lcd.setCursor(0, 1);
73     lcd.print("Credencial?");
74     // Send Message to WhatsApp
75     sendMessage("RFID IOT: Porta aberta remotamente!");
76     relayAtivado = true; // Marca que o relé foi ativado
77     tempoAnterior = millis(); // Registra o tempo atual
78     Blynk.virtualWrite(V0, 1); // Atualiza o estado do botão no Blynk
79 }

```

Figura 17: Código desenvolvido para o projeto (3/13).

```

acesso_rfido.ino
79 } else {
80     digitalWrite(relayPin, LOW); // Desativa o relé (caso o botão seja desligado manualmente)
81     lcd.clear();
82     lcd.noCursor();
83     lcd.print("Acesso encerrado");
84     lcd.setCursor(0, 1);
85     lcd.print("remotamente!");
86     delay(2000);
87     lcd.clear();
88     lcd.print("Acesso restrito!");
89     lcd.setCursor(0, 1);
90     lcd.print("Credencial?");
91     // Send Message to WhatsApp
92     sendMessage("RFID IOT: Porta fechada remotamente!");
93     relayAtivado = false; // Reseta o estado do relé
94     Blynk.virtualWrite(V0, 0); // Atualiza o estado do botão no Blynk
95 }
96 }
97 // This function is called every time the device is connected to the Blynk.Cloud
98 BLYNK_CONNECTED()
99 {
100 // Change Web Link Button message to "Congratulations!"
101 Blynk.setProperty(V3, "offImageUrl", "https://static-image.nyc3.cdn.digitaloceanspaces.com/general/fte/congratulations.png");
102 Blynk.setProperty(V3, "onImageUrl", "https://static-image.nyc3.cdn.digitaloceanspaces.com/general/fte/congratulations_pressed.png");
103 Blynk.setProperty(V3, "url", "https://docs.blynk.io/en/getting-started/what-do-i-need-to-blynk/how-quickstart-device-was-made");
104 }

```

Figura 18: Código desenvolvido para o projeto (4/13).

```
acesso_rfid | Arduino IDE 2.3.4
Arquivo Editar Rascunho Ferramentas Ajuda

DOIT ESP32 DEVKIT V1

acesso_rfid.ino
105 // This function sends Arduino's uptime every second to Virtual Pin 2. (DA PLACA ESP32 PARA A PLATAFORMA BLYNK)
106 void myTimerEvent()
107 {
108     // You can send any value at any time.
109     // Please don't send more than 10 values per second.
110     Blynk.virtualWrite(V2, millis() / 1000);
111 }
112 MFRC522 rfid(SS_PIN, RST_PIN);
113 const byte ROWS = 4;
114 const byte COLS = 3;
115 char keys[ROWS][COLS] = {
116     {'1', '2', '3'},
117     {'4', '5', '6'},
118     {'7', '8', '9'},
119     {'*', '0', '#'}
120 };
121 byte rowPins[ROWS] = {32, 33, 25, 26};
122 byte colPins[COLS] = {27, 14, 12};
123 Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);
124 const int ledRed = 2;
125 const int ledYellow = 5;
126 const int ledGreen = 16;
127 const int buzzer = 17;
128 bool senhaDigitada = false;
129 // Senha padrão
130 String senha = "123456";
131 String entradaSenha = "";
```

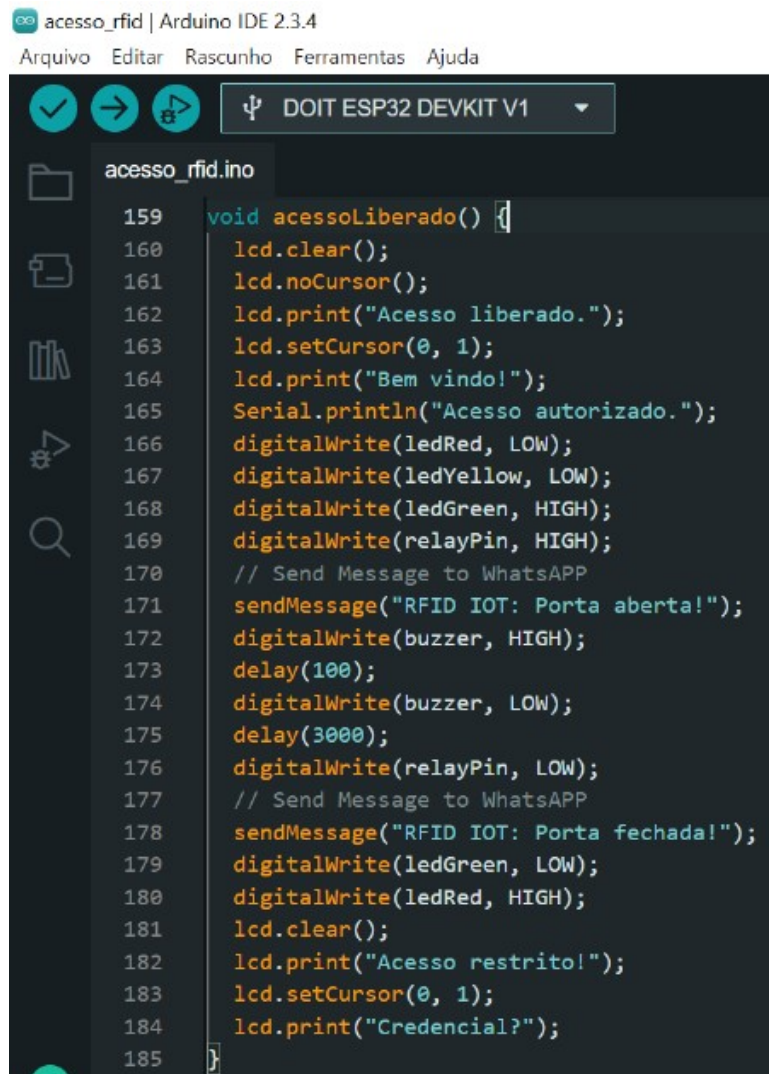
Figura 19: Código desenvolvido para o projeto (5/13).

```
acesso_rfid | Arduino IDE 2.3.4
Arquivo Editar Rascunho Ferramentas Ajuda

DOIT ESP32 DEVKIT V1

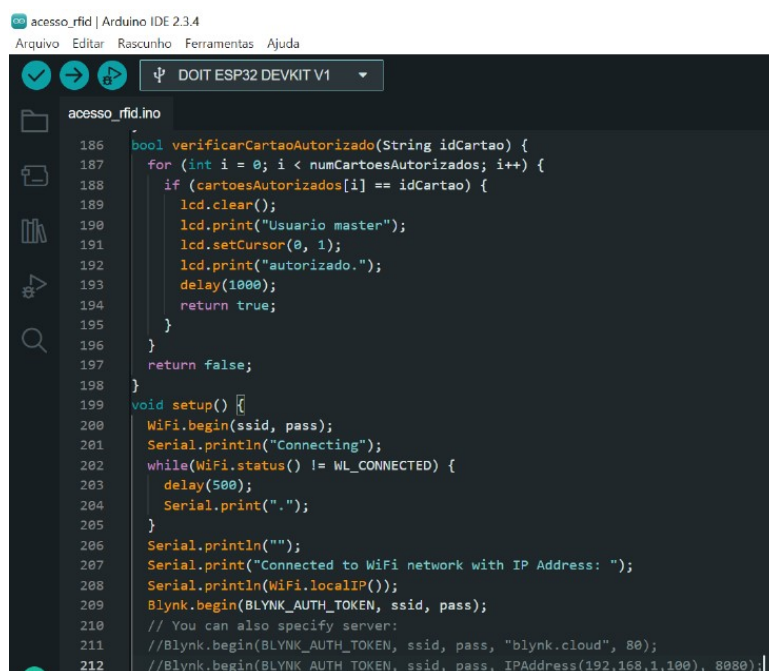
acesso_rfid.ino
132 // Lista de IDs de cartões autorizados
133 const int numCartoesAutorizados = 2;
134 String cartoesAutorizados[numCartoesAutorizados] = {
135     "52 7D 83 54", // ID do cartão
136     "9F 18 32 28" // ID da tag
137 };
138 // Funções auxiliares
139 void acessoNegado() {
140     lcd.clear();
141     lcd.noCursor();
142     lcd.print("Senha errada!");
143     lcd.setCursor(0, 1);
144     lcd.print("Acesso negado!");
145     digitalWrite(buzzer, HIGH);
146     delay(500);
147     digitalWrite(buzzer, LOW);
148     delay(250);
149     digitalWrite(buzzer, HIGH);
150     delay(500);
151     digitalWrite(buzzer, LOW);
152     Serial.println("Senha errada! Acesso não autorizado!");
153     delay(3000);
154     lcd.clear();
155     lcd.print("Acesso restrito!");
156     lcd.setCursor(0, 1);
157     lcd.print("Credencial?");
158 }
```

Figura 20: Código desenvolvido para o projeto (6/13).



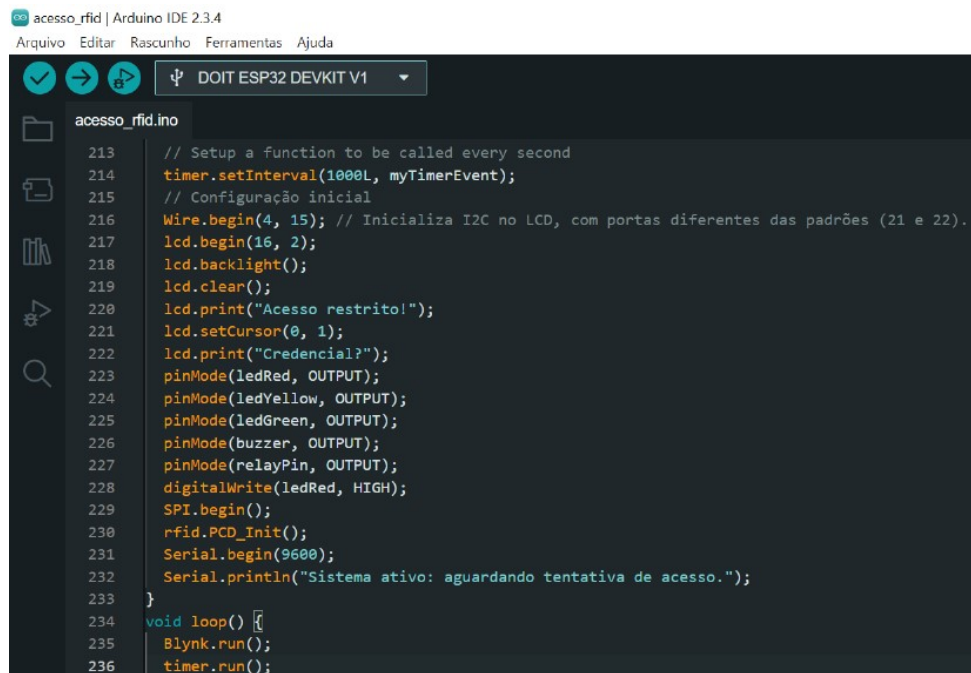
```
acesso_rfid.ino
159 void acessoLiberado() {
160     lcd.clear();
161     lcd.noCursor();
162     lcd.print("Acesso liberado.");
163     lcd.setCursor(0, 1);
164     lcd.print("Bem vindo!");
165     Serial.println("Acesso autorizado.");
166     digitalWrite(ledRed, LOW);
167     digitalWrite(ledYellow, LOW);
168     digitalWrite(ledGreen, HIGH);
169     digitalWrite(relayPin, HIGH);
170     // Send Message to WhatsApp
171     sendMessage("RFID IOT: Porta aberta!");
172     digitalWrite(buzzer, HIGH);
173     delay(100);
174     digitalWrite(buzzer, LOW);
175     delay(3000);
176     digitalWrite(relayPin, LOW);
177     // Send Message to WhatsApp
178     sendMessage("RFID IOT: Porta fechada!");
179     digitalWrite(ledGreen, LOW);
180     digitalWrite(ledRed, HIGH);
181     lcd.clear();
182     lcd.print("Acesso restrito!");
183     lcd.setCursor(0, 1);
184     lcd.print("Credencial?");
185 }
```

Figura 21: Código desenvolvido para o projeto (7/13).



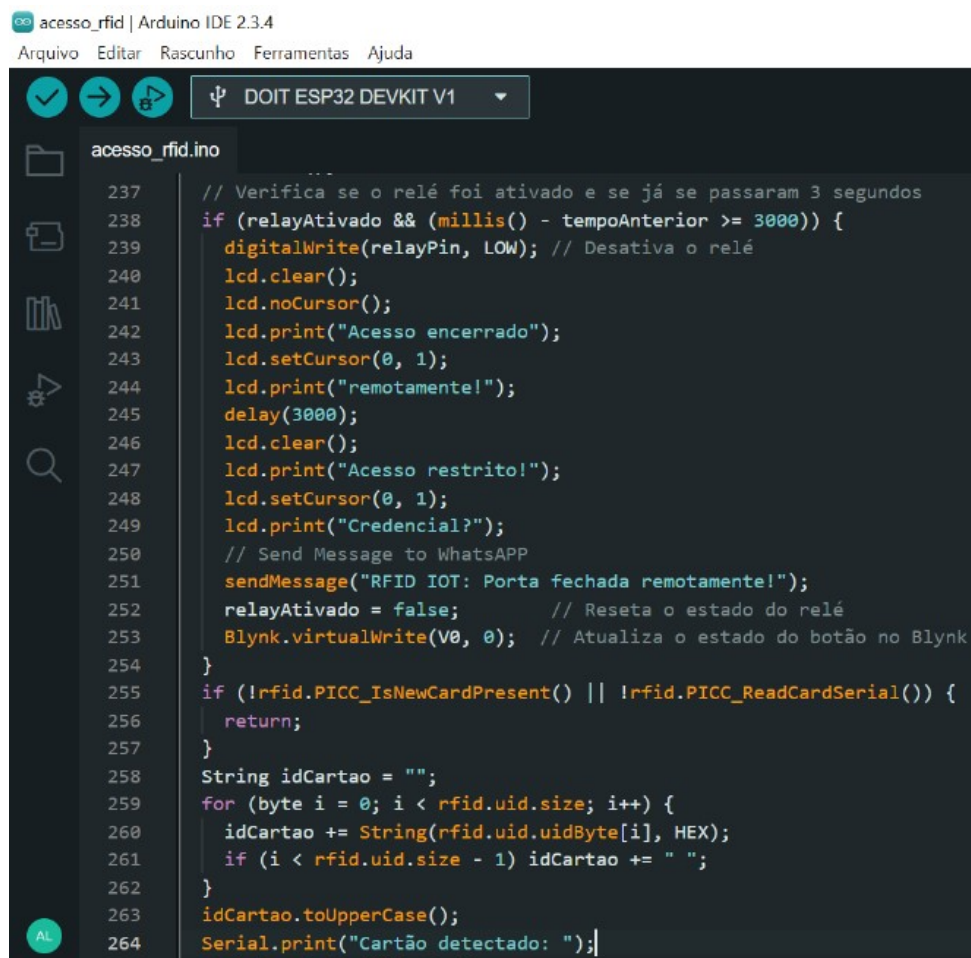
```
acesso_rfid.ino
186 bool verificarCartaoAutorizado(String idCartao) {
187     for (int i = 0; i < numCartoesAutorizados; i++) {
188         if (cartoesAutorizados[i] == idCartao) {
189             lcd.clear();
190             lcd.print("Usuario master");
191             lcd.setCursor(0, 1);
192             lcd.print("autorizado.");
193             delay(1000);
194             return true;
195         }
196     }
197     return false;
198 }
199 void setup() {
200     WiFi.begin(ssid, pass);
201     Serial.println("Connecting");
202     while(WiFi.status() != WL_CONNECTED) {
203         delay(500);
204         Serial.print(".");
205     }
206     Serial.println("");
207     Serial.print("Connected to WiFi network with IP Address: ");
208     Serial.println(WiFi.localIP());
209     Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
210     // You can also specify server:
211     //Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass, "blynk.cloud", 80);
212     //Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass, IPAddress(192,168,1,100), 8080);
```

Figura 22: Código desenvolvido para o projeto (8/13).



```
213 // Setup a function to be called every second
214 timer.setInterval(1000L, myTimerEvent);
215 // Configuração inicial
216 Wire.begin(4, 15); // Inicializa I2C no LCD, com portas diferentes das padrões (21 e 22).
217 lcd.begin(16, 2);
218 lcd.backlight();
219 lcd.clear();
220 lcd.print("Acesso restrito!");
221 lcd.setCursor(0, 1);
222 lcd.print("Credencial?");
223 pinMode(ledRed, OUTPUT);
224 pinMode(ledYellow, OUTPUT);
225 pinMode(ledGreen, OUTPUT);
226 pinMode(buzzer, OUTPUT);
227 pinMode(relayPin, OUTPUT);
228 digitalWrite(ledRed, HIGH);
229 SPI.begin();
230 rfid.PCD_Init();
231 Serial.begin(9600);
232 Serial.println("Sistema ativo: aguardando tentativa de acesso.");
233 }
234 void loop() {
235   Blynk.run();
236   timer.run();
```

Figura 23: Código desenvolvido para o projeto (9/13).



```
237 // Verifica se o relé foi ativado e se já se passaram 3 segundos
238 if (relayAtivado && (millis() - tempoAnterior >= 3000)) {
239   digitalWrite(relayPin, LOW); // Desativa o relé
240   lcd.clear();
241   lcd.noCursor();
242   lcd.print("Acesso encerrado");
243   lcd.setCursor(0, 1);
244   lcd.print("remotamente!");
245   delay(3000);
246   lcd.clear();
247   lcd.print("Acesso restrito!");
248   lcd.setCursor(0, 1);
249   lcd.print("Credencial?");
250   // Send Message to WhatsApp
251   sendMessage("RFID IOT: Porta fechada remotamente!");
252   relayAtivado = false; // Reseta o estado do relé
253   Blynk.virtualWrite(V0, 0); // Atualiza o estado do botão no Blynk
254 }
255 if (!rfid.PICC_IsNewCardPresent() || !rfid.PICC_ReadCardSerial()) {
256   return;
257 }
258 String idCartao = "";
259 for (byte i = 0; i < rfid.uid.size; i++) {
260   idCartao += String(rfid.uid.uidByte[i], HEX);
261   if (i < rfid.uid.size - 1) idCartao += " ";
262 }
263 idCartao.toUpperCase();
264 Serial.print("Cartão detectado: ");
```

Figura 24: Código desenvolvido para o projeto (10/13).


```

acesso_rfid.ino
265 Serial.println(idCartao);
266 if (verificarCartaoAutorizado(idCartao)) {
267   Serial.println("Usuario máster identificado.");
268   acessoLiberado(); // Libera acesso sem senha para cartões autorizados
269   // Send Message to WhatsApp
270   sendMessage("RFID IOT: Acesso liberado ao usuário máster!");
271   Serial.println("Sistema ativo: aguardando tentativa de acesso.");
272 } else {
273   lcd.clear();
274   lcd.print("Tag detectada.");
275   lcd.setCursor(0, 1);
276   lcd.print("Exige senha!");
277   digitalWrite(ledYellow, HIGH);
278   delay(2000);
279   entradaSenha = "";
280   lcd.clear();
281   lcd.print("Informe senha:");
282   lcd.setCursor(0, 1);
283   while (true) {
284     lcd.cursor();
285     char key = keypad.getKey();
286     if (key) {
287       if (key == '*') {
288         Serial.println("");
289         if (entradaSenha == senha) {
290           acessoLiberado();
291           // Send Message to WhatsApp
292           sendMessage("RFID IOT: Acesso liberado por senha!");

```

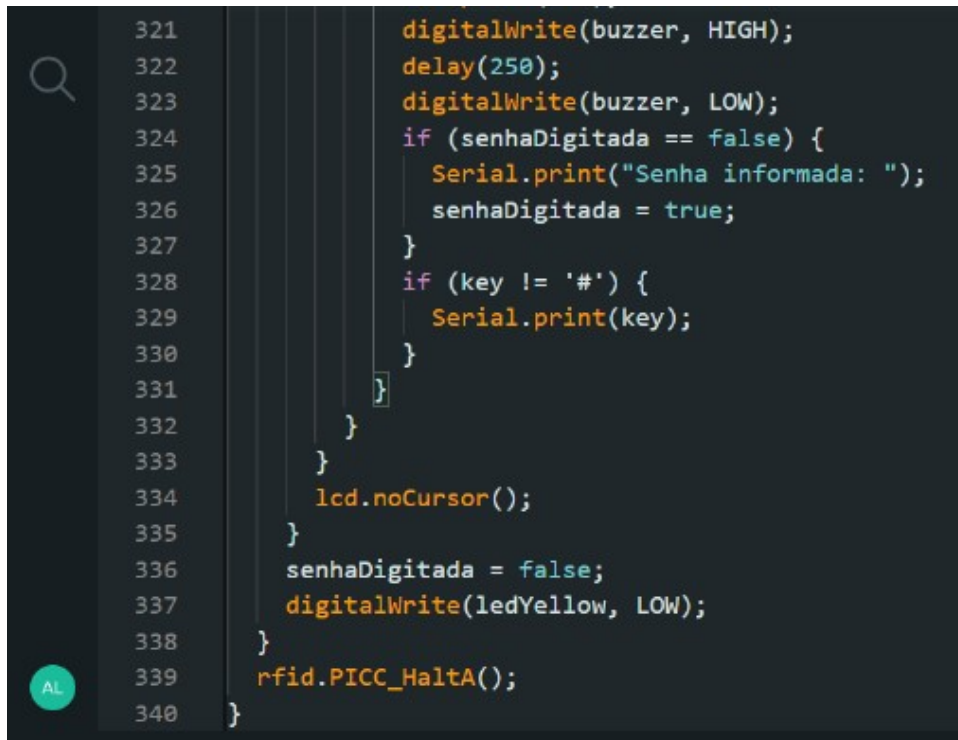
Figura 25: Código desenvolvido para o projeto (11/13).

```

acesso_rfid.ino
293   } else {
294     acessoNegado();
295     // Send Message to WhatsApp
296     sendMessage("RFID IOT: Tentativa de acesso negada: senha errada!");
297   }
298   Serial.println("Sistema ativo: aguardando tentativa de acesso.");
299   break;
300 } else if (key == '#') {
301   if (entradaSenha.length() > 0) {
302     entradaSenha.remove(entradaSenha.length() - 1);
303     digitalWrite(buzzer, HIGH);
304     delay(250);
305     digitalWrite(buzzer, LOW);
306     lcd.setCursor(entradaSenha.length(), 1);
307     lcd.print(" ");
308     lcd.setCursor(entradaSenha.length(), 1);
309     if (senhaDigitada == false) {
310       Serial.print("Senha informada: ");
311       senhaDigitada = true;
312     }
313     if (key != '#') {
314       Serial.print(key);
315     }
316   }
317 } else {
318   if (entradaSenha.length() < 6) {
319     entradaSenha += key;
320     lcd.print("*");

```

Figura 26: Código desenvolvido para o projeto (12/13).



```

321     digitalWrite(buzzer, HIGH);
322     delay(250);
323     digitalWrite(buzzer, LOW);
324     if (senhaDigitada == false) {
325         Serial.print("Senha informada: ");
326         senhaDigitada = true;
327     }
328     if (key != '#') {
329         Serial.print(key);
330     }
331 }
332 }
333 }
334     lcd.noCursor();
335 }
336     senhaDigitada = false;
337     digitalWrite(ledYellow, LOW);
338 }
339     rfid.PICC_HaltA();
340 }

```

Figura 27: Código desenvolvido para o projeto (13/13).

A demonstração completa do protótipo pode ser verificada em:

<https://youtube.com/shorts/Cfg2asOm7Lw?feature=share>

5 Considerações finais

De acordo com os testes realizados, o sistema mostrou-se funcional quanto ao proposto inicialmente. Há de se considerar as ressalvas quanto à alimentação própria do sistema, que pode agregar independência ao mesmo, assim como observar eventuais intermitências quanto ao acesso à internet. Estes pontos parecem ser o ponto de partida para posterior melhoramento do protótipo, além de outras necessidades que porventura ocorram.

6 Referências

Whashington J. F. Resende, Lúcio R. Júnior, Guilherme H. Alves, Marcelo C. Dias, Antonio M. B. da Silva (2021) "Desenvolvimento de um sistema para monitoramento e controle patrimonial, utilizando Rfid e dispositivos Iot", Brazilian Journal of Development, Curitiba, Brasil, Julho.

Pedro H. M. Araujo, Renan P. Figueiredo, Douglas L. Dias, Sandro C. S. Jucá (2016) "Controle de acesso RFID utilizando o princípio de Internet das Coisas", Escola Regional de Informática do Piauí, Teresina, Brasil

Alldatasheet, "RC522 Datasheet",

<https://www.alldatasheet.com/datasheet-pdf/pdf/346109/NXP/RC522.html>, Fevereiro

Unesp Sorocaba, "Radiação eletromagnética",

<https://www.sorocaba.unesp.br/Home/Graduacao/EngenhariaAmbiental/robertowlourenco1502/rem.pdf>, Fevereiro

UFRJ, "RFID", <https://www.gta.ufrj.br/grad/121/rfid/links/aplicacoes.html>, Fevereiro

Cryslaine C. C. Nascimento, Kality D. de Freitas, Joao V. N. Lopes, Thiago R. Fernandes, Sonagno de P. Oliveira "APLICAÇÃO DO SISTEMA RFID NO CONTROLE DE ACESSO DE VEÍCULOS EM CONDOMÍNIOS", ENEGE 2015, Fortaleza, Brasil, Outubro