

Integração de Sistemas e Tecnologias

Interoperabilidade de Sistemas e Tecnologias

Tutorial 10 – WebServices e Arduinos.

Escola Superior de Tecnologia e Gestão
Instituto Politécnico de Viana do Castelo
Braga, Portugal

Jorge Ribeiro
jribeiro@estg.ipvc.pt

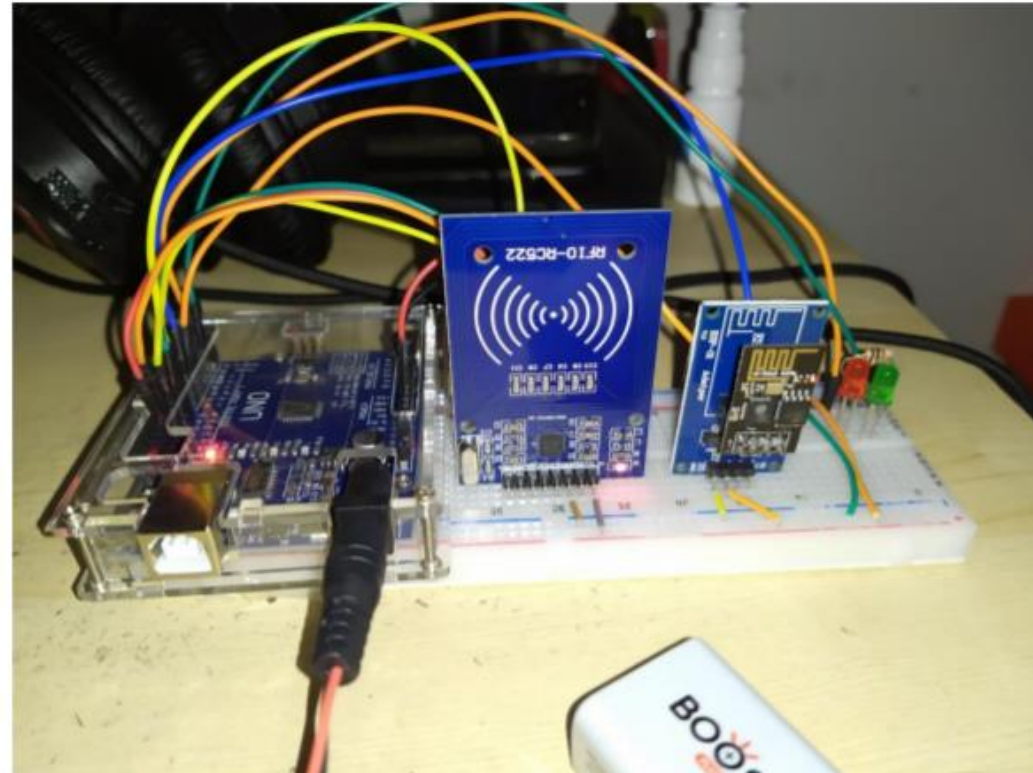
Arduino a ler RFID e a colocar o NUID da tag RFID numa base de dados

De modo a testar a integração de sistemas através de web services, com diferentes tipos de sistema, montei um Arduino com um leitor RFID (RFID RC522) e com um módulo WIFI (ESP8266 ESP-01).

Este pequeno exemplo lê uma tag RFID (com frequência nos 13.56 MHz) e, através de um Web Service desenvolvido em PHP, coloca o UID (Unique identifier) da tag numa base de dados MySQL.

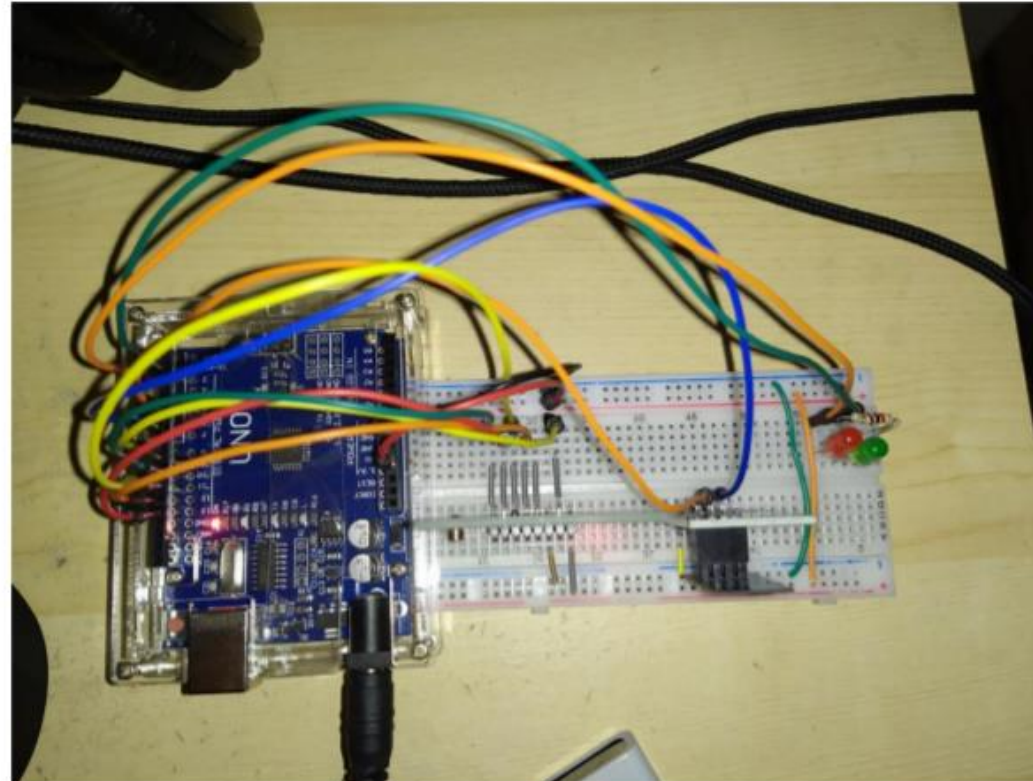
Arduino a ler RFID e a colocar o NUID da tag RFID numa base de dados

O Circuito:



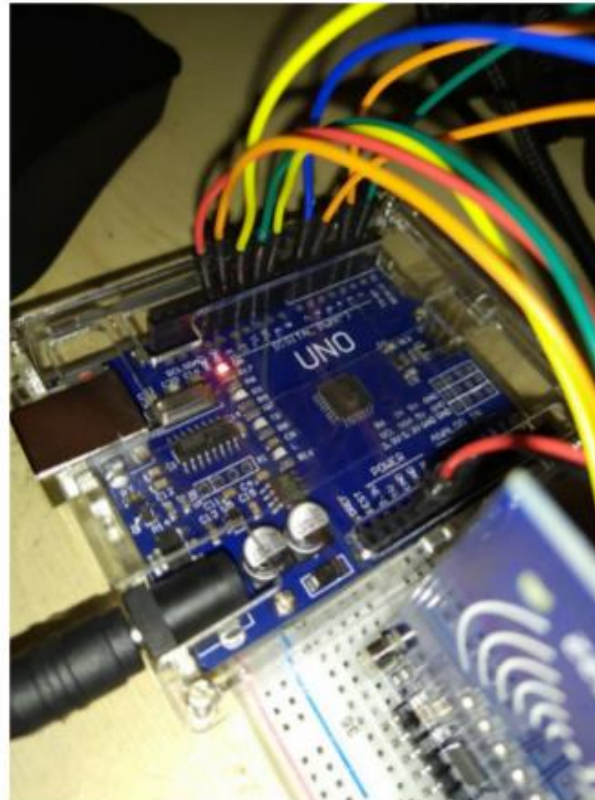
Arduino a ler RFID e a colocar o NUID da tag RFID numa base de dados

O Circuito:



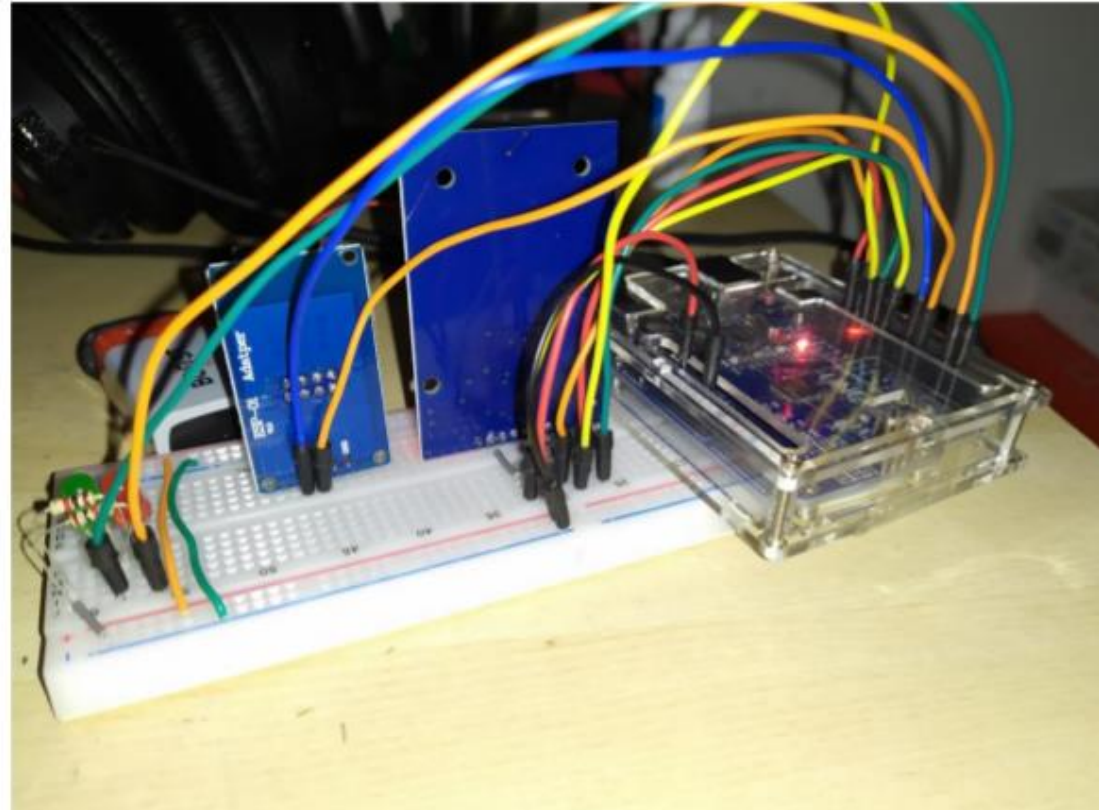
Arduino a ler RFID e a colocar o NUID da tag RFID numa base de dados

O Circuito:



Arduino a ler RFID e a colocar o NUID da tag RFID numa base de dados

O Circuito:



Arduino a ler RFID e a colocar o NUID da tag RFID numa base de dados

Código do WS – Recebe o UID por parâmetro GET

```
1  <?php
2  if(!empty($_GET['NUID'])) {
3      $conn = new mysqli("localhost", "root", "", "teste_arduino");
4      if ($conn->connect_error) {
5          die("Connection failed: " . $conn->connect_error);
6      }
7
8      $sql = "INSERT INTO passagens (UDI) VALUES ('".$_GET['NUID']."'");
9
10     if ($conn->query($sql) === TRUE) {
11         echo "RegistoInserido";
12     } else {
13         echo "Error: " . $sql . "<br>" . $conn->error;
14     }
15
16     $conn->close();
17 }
18 echo "A tentar...";
19 ?>
```

Arduino a ler RFID e a colocar o NUID da tag RFID numa base de dados

TESTE: Tabela na Base de Dados antes de Passar Cartão

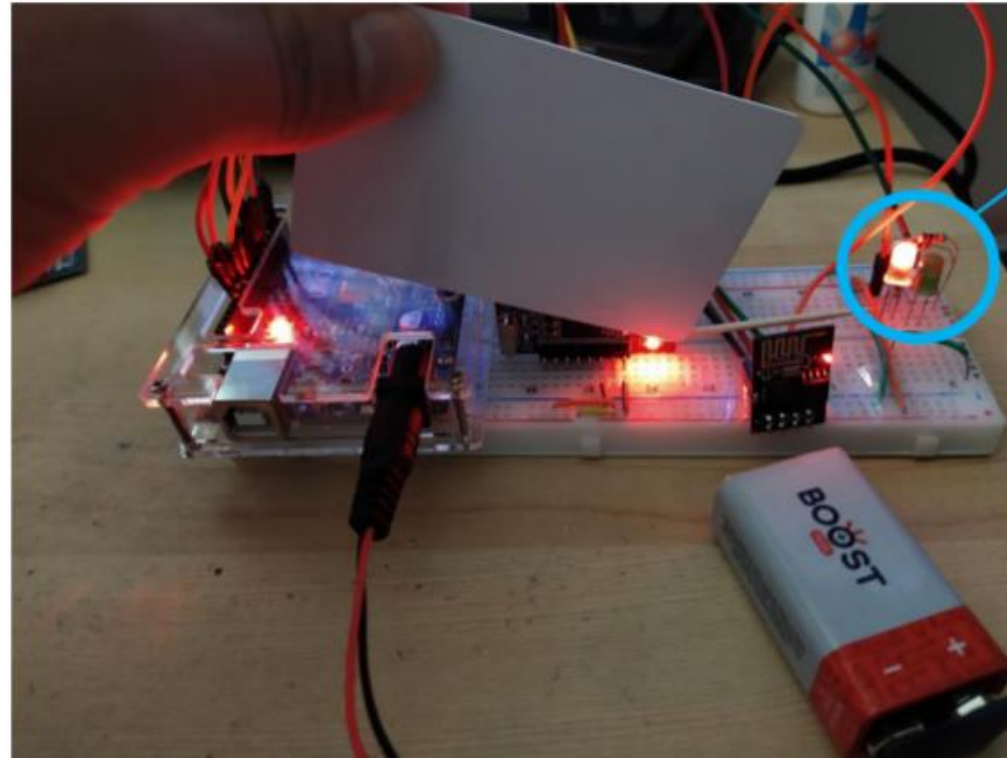
				id	datahora	UDI
<input type="checkbox"/>	 Editar	 Copiar	 Apagar	27	2019-11-19 14:25:14	E7:ED:D6:2B
<input type="checkbox"/>	 Editar	 Copiar	 Apagar	26	2019-11-19 14:25:07	E7:ED:D6:2B
<input type="checkbox"/>	 Editar	 Copiar	 Apagar	25	2019-11-19 14:24:43	79:71:BF:99
<input type="checkbox"/>	 Editar	 Copiar	 Apagar	24	2019-11-16 20:38:41	79:71:BF:99
<input type="checkbox"/>	 Editar	 Copiar	 Apagar	23	2019-11-16 20:38:37	79:71:BF:99
<input type="checkbox"/>	 Editar	 Copiar	 Apagar	22	2019-11-16 20:38:34	79:71:BF:99
<input type="checkbox"/>	 Editar	 Copiar	 Apagar	21	2019-11-16 20:38:23	79:71:BF:99

Apenas 7 Registos

Outras integrações desenvolvidas

Arduino a ler RFID e a colocar o NUID da tag RFID numa base de dados

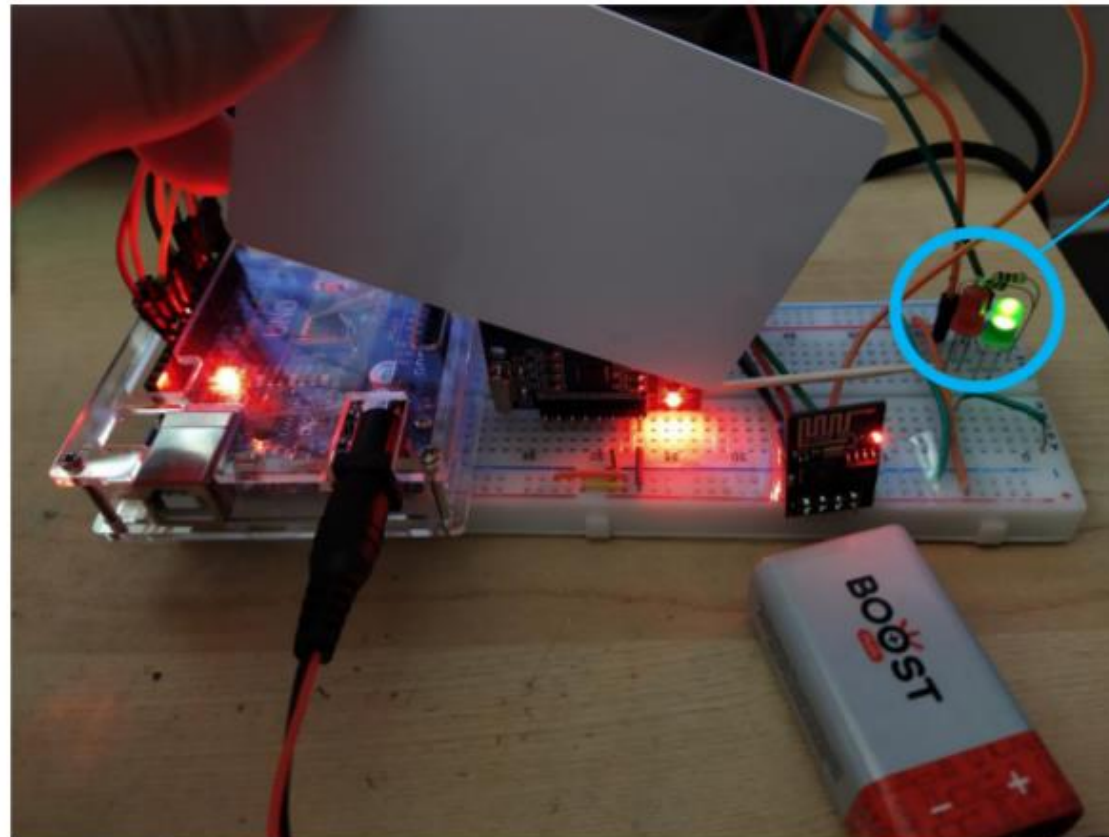
TESTE: Passagem Cartão



Led Vermelho
liga em caso
de erro

Arduino a ler RFID e a colocar o NUID da tag RFID numa base de dados

TESTE: Passagem Cartão



Led Verde liga
em caso de
Sucesso

Arduino a ler RFID e a colocar o NUID da tag RFID numa base de dados

TESTE: Passagem Cartão

Após obter o LED Verde ligado, que indica sucesso, verifiquei a base de dados de modo a provar o sucesso.

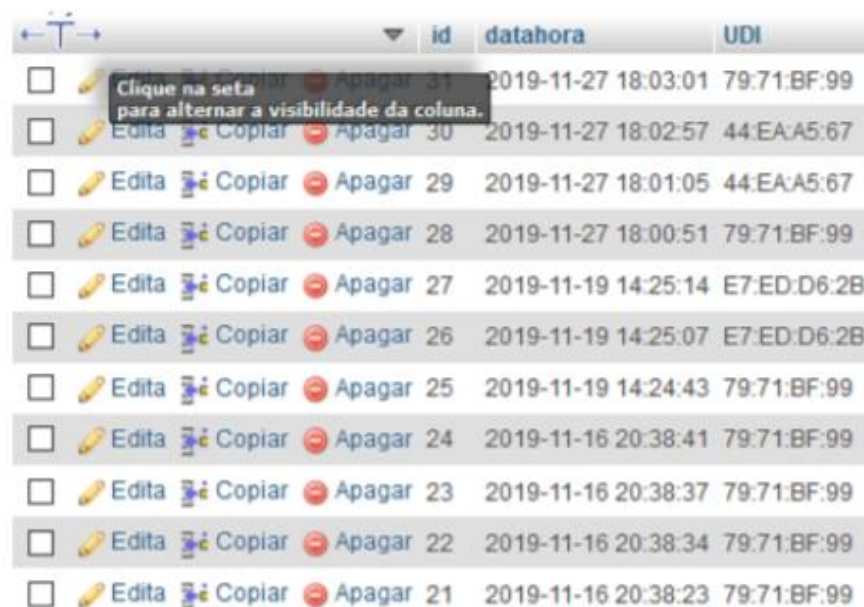
			id	datahora	UDI
<input type="checkbox"/>	✎ Edita	📋 Copiar	🚫 Apagar	28	2019-11-27 18:00:51 79:71:BF:99
<input type="checkbox"/>	✎ Edita	📋 Copiar	🚫 Apagar	27	2019-11-19 14:25:14 E7:ED:D6:2B
<input type="checkbox"/>	✎ Edita	📋 Copiar	🚫 Apagar	26	2019-11-19 14:25:07 E7:ED:D6:2B
<input type="checkbox"/>	✎ Edita	📋 Copiar	🚫 Apagar	25	2019-11-19 14:24:43 79:71:BF:99
<input type="checkbox"/>	✎ Edita	📋 Copiar	🚫 Apagar	24	2019-11-16 20:38:41 79:71:BF:99
<input type="checkbox"/>	✎ Edita	📋 Copiar	🚫 Apagar	23	2019-11-16 20:38:37 79:71:BF:99
<input type="checkbox"/>	✎ Edita	📋 Copiar	🚫 Apagar	22	2019-11-16 20:38:34 79:71:BF:99
<input type="checkbox"/>	✎ Edita	📋 Copiar	🚫 Apagar	21	2019-11-16 20:38:23 79:71:BF:99


































Verifica-se que existe um novo registo

Arduino a ler RFID e a colocar o NUID da tag RFID numa base de dados

TESTE: Passagem Cartão

Base de dados após passar cartões mais algumas vezes



				id	datahora	UDI
<input type="checkbox"/>				31	2019-11-27 18:03:01	79:71:BF:99
<input type="checkbox"/>				30	2019-11-27 18:02:57	44:EA:A5:67
<input type="checkbox"/>				29	2019-11-27 18:01:05	44:EA:A5:67
<input type="checkbox"/>				28	2019-11-27 18:00:51	79:71:BF:99
<input type="checkbox"/>				27	2019-11-19 14:25:14	E7:ED:D6:2B
<input type="checkbox"/>				26	2019-11-19 14:25:07	E7:ED:D6:2B
<input type="checkbox"/>				25	2019-11-19 14:24:43	79:71:BF:99
<input type="checkbox"/>				24	2019-11-16 20:38:41	79:71:BF:99
<input type="checkbox"/>				23	2019-11-16 20:38:37	79:71:BF:99
<input type="checkbox"/>				22	2019-11-16 20:38:34	79:71:BF:99
<input type="checkbox"/>				21	2019-11-16 20:38:23	79:71:BF:99



leRFIDparaMysqlESP01.ino

```
1 #include <SPI.h>
2 #include <MFRC522.h>
3 #include <SoftwareSerial.h>
4
5 #define PIN_ENVIO_SUCESSO 2
6 #define PIN_ENVIO_ERRO 3
7
8 // RFID RC552
9 #define SS_PIN 10
10 #define RST_PIN 9
11
12 MFRC522 rfid(SS_PIN, RST_PIN); // Instance of the class
13 MFRC522::MIFARE_Key key;
14 byte nuidPICC[4]; // inicializar array que guarda o NUID do último cartão passado
15 char strAEnviarParaBD[32] = "";
16
17 // ESP8266-ESP01
18 const byte rxPin = 6; ///TX do esp
19 const byte txPin = 7; // RX do esp
20 String ssid = "NetGratis_MasNaoParaTi";
21 String password = "252682350sapo";
22 SoftwareSerial esp8266 (rxPin, txPin);
23 String path = "/WS_arduino_inserePassagem/inserePassagemCartao.php";
24 String server = "192.168.1.67";
25 String getRequest = "GET " + path + " HTTP/1.1\r\n" + "Host: " + server + "\r\n" + "Connection: keep-alive\r\n\r\n";
26 String getRequestLength = String(getRequest.length());
27 String response="";
28
29 void setup() {
30     // GERAL
31     Serial.begin(9600);
32     pinMode(PIN_ENVIO_SUCESSO, OUTPUT);
33     digitalWrite(PIN_ENVIO_SUCESSO, LOW);
34     pinMode(PIN_ENVIO_ERRO, OUTPUT);
35     digitalWrite(PIN_ENVIO_ERRO, LOW);
36 }
```



```
29 void setup() {
30     // GERAL
31     Serial.begin(9600);
32     pinMode(PIN_ENVIO_SUCESSO, OUTPUT);
33     digitalWrite(PIN_ENVIO_SUCESSO, LOW);
34     pinMode(PIN_ENVIO_ERRO, OUTPUT);
35     digitalWrite(PIN_ENVIO_ERRO, LOW);
36
37     // RDIF SETUP
38     SPI.begin(); // Init SPI bus
39     rfid.PCD_Init(); // Init MFRC522
40
41     // Inicializa password cartão RFID
42     for (byte i = 0; i < 6; i++) {
43         key.keyByte[i] = 0xFF;
44     }
45
46     Serial.println(F("A aguardar passagem de cartão RFID"));
47     Serial.print(F("A usar a seguinte chave:"));
48     printHex(key.keyByte, MFRC522::MF_KEY_SIZE, strAEnviarParaBD);
49
50
51     // ESP8622 ESP01 SETUP
52     esp8266.begin(9600);
53     reset();
54     connectWifi();
55 }
56
```

```
57 void loop() {
58     delay(500);
59     // Verifica se há cartão no leitor
60     if (!rfid.PICC_IsNewCardPresent())
61         return;
62
63     if(!rfid.PICC_ReadCardSerial())
64         return;
65
66     Serial.print(F("PICC type: "));
67     MFRC522::PICC_Type piccType = rfid.PICC_GetType(rfid.uid.sak);
68     Serial.println(rfid.PICC_GetTypeName(piccType));
69
70     // Check is the PICC of Classic MIFARE type
71     if (piccType != MFRC522::PICC_TYPE_MIFARE_MINI && piccType != MFRC522::PICC_TYPE_MIFARE_1K && piccType != MFRC522::PICC_TYPE_MIFARE_4K) {
72         Serial.println(F("A Tag rfid não é MIFARE Classic."));
73         return;
74     }
75
76     Serial.println(F("Cartão Detetado! A Ler..."));
77
78     // Store NUID into nuidPICC array
79     for (byte i = 0; i < 4; i++) {
80         nuidPICC[i] = rfid.uid.uidByte[i];
81     }
82 }
```

```
83 Serial.println(F("NUID:"));
84 Serial.print(F("-- Hexadecimal: "));
85 for( int i = 0; i < sizeof(strAEnviarParaBD); ++i )
86     strAEnviarParaBD[i] = (char)0;
87 printHex(rfid.uid.uidByte, rfid.uid.size, strAEnviarParaBD);
88 Serial.println();
89 Serial.println(strAEnviarParaBD);
90
91 Serial.println();
92 // -- CHAMA API PARA INSERIR PASSAGEM
93 Serial.println(F("A tentar escrever na base de dados..."));
94 escreveBD(strAEnviarParaBD);
95
96 // Halt PICC
97 rfid.PICC_HaltA();
98
99 // Stop encryption on PCD
100 rfid.PCD_StopCryptol();
101
102 }
103
```

```
104 void escreveBD(String string){
105     int inserido = 0;
106     string = string.substring(0, string.length() - 1);
107     //string.replace(":", "%3A");
108     //string.replace("?", "%3F");
109     String pathusar = path;
110     pathusar.concat("?NUID=");
111     pathusar.concat(string);
112
113     Serial.println(pathusar);
114     getRequest = "GET " + pathusar + " HTTP/1.1\r\n" + "Host: " + server + "\r\n" + "Connection: keep-alive\r\n\r\n";
115     getRequestLength = String(getRequest.length());
116     Serial.println(getRequest);
117
118     esp8266.println("AT+CIPSTART=\"TCP\", \"" + server + "\", 80");
119     if(esp8266.find("OK"))
120         Serial.println("TCP Connection Ready");
121     esp8266.println("AT+CIPSEND=" + getRequestLength);
122     if(esp8266.find(">")) {
123         Serial.println("Sending Request...");
124         esp8266.print(getRequest);
125     } else {
126         Serial.println("Aconteceu algo...");
127         digitalWrite(PIN_ENVIO_ERRO, HIGH);
128         delay(1000);
129         digitalWrite(PIN_ENVIO_ERRO, LOW);
130         return;
131     }
132     if(esp8266.find("SEND OK"))
133         Serial.println("Request Sent");
134     while(!esp8266.available()) {};
135
136 }
```

```
136
137 if(esp8266.find("RegistoInserido")) {
138     inserido = 1;
139 }
140
141 espRead();
142 esp8266.println("AT+CIPCLOSE");
143 if(esp8266.find("OK")) Serial.println("TCP Connection Closed");
144 if(inserido == 1){
145     digitalWrite(PIN_ENVIO_SUCESSO, HIGH);
146     delay(1000);
147     digitalWrite(PIN_ENVIO_SUCESSO, LOW);
148 } else {
149     digitalWrite(PIN_ENVIO_ERRO, HIGH);
150     delay(1000);
151     digitalWrite(PIN_ENVIO_ERRO, LOW);
152 }
153 delay(300);
154 }
155
156 void espRead() {
157     String c;
158     while(esp8266.available()) {
159         c = esp8266.readString();
160         Serial.print(c);
161     }
162 }
163
164 void espClear() {
165     while(esp8266.available()) {
166         esp8266.read();
167     }
168 }
```



```
169
170 // PRINT array byte em HEXADECIMAL.
171 void printHex(byte *buffer, byte bufferSize, char* stringBD) {
172     for (byte i = 0; i < bufferSize; i++) {
173         Serial.print(buffer[i] < 0x10 ? " 0" : " ");
174         Serial.print(buffer[i], HEX);
175
176         byte nib1 = (buffer[i] >> 4) & 0x0F;
177         byte nib2 = (buffer[i] >> 0) & 0x0F;
178         stringBD[i*3+0] = nib1 < 0xA ? '0' + nib1 : 'A' + nib1 - 0xA;
179         stringBD[i*3+1] = nib2 < 0xA ? '0' + nib2 : 'A' + nib2 - 0xA;
180         if(i < 4)
181             stringBD[i*3+2] = ':';
182     }
183 }
184
185 // REINICIA ESP8266
186 void reset() {
187     Serial.println("Resetting WiFi");
188     esp8266.println("AT+RST");
189     delay(1000);
190     if(esp8266.find("OK"))
191         Serial.println("Reset!");
192 }
```

```
193
194 // LIGA ESP8266 AO WIFI
195 void connectWifi() {
196     espClear();
197     Serial.println("Connecting...");
198     String CMD = "AT+CWJAP=\"" + ssid + "\",\"" + password + "\"";
199     esp8266.println(CMD);
200     while(!esp8266.available()) {};
201     if(esp8266.find("OK"))
202         Serial.println("Connected");
203     else
204         Serial.println("Couldn't connect to WiFi");
205 }
206
207
208 // DEFINE MODOD DO ESP8266
209 void setMode(String mode) {
210     Serial.println("Setting Mode = " + mode);
211     esp8266.println("AT+CWMODE=" + mode);
212     delay(1000);
213     espRead();
214 }
215
```