

Faculdade de Tecnologia Professor Francisco de Moura – FATEC Jacareí

Disciplina Internet das Coisas e Aplicações

Prof. Henrique Louro

RELATÓRIO ATIVIDADE PRÁTICA P2

1º semestre 2025

Aluno: André Flávio de Oliveira

Aluno: Mauro do Prado Santos

Introdução:

O presente relatório documenta o desenvolvimento de um sistema de controle de acesso baseado em tecnologia RFID, utilizando uma placa Arduino Uno, um módulo RFID RC522, um servo motor SG90, um módulo sensor touch capacitivo TTP224 4 vias e integração com um banco de dados PostgreSQL gerenciado por um script Python. O objetivo do projeto foi criar um protótipo funcional capaz de autenticar usuários por meio de cartões RFID ou entrada de senha via sensores touch, simular o controle de acesso físico (abertura e fechamento de uma porta) e registrar os eventos de acesso em um banco de dados para monitoramento.

A placa Arduino é uma plataforma de prototipagem eletrônica de código aberto, amplamente utilizada em projetos educacionais e industriais devido à sua flexibilidade e facilidade de programação. O módulo RFID RC522 permite a leitura de identificadores únicos (UIDs) de cartões ou tags, sendo ideal para aplicações de autenticação. O módulo TTP224 permite a entrada de senhas por meio de toques em sensores capacitivos, oferecendo uma alternativa à autenticação RFID. O servo motor SG90 simula a abertura e fechamento de uma porta, e a integração com o banco de dados PostgreSQL, por meio de um script Python, possibilita o armazenamento e análise dos registros de acesso.

Este sistema tem aplicações práticas em cenários como controle de entrada em escritórios, escolas, residências ou áreas restritas, além de sistemas de ponto eletrônico e monitoramento de segurança.

Metodologia:

O desenvolvimento do protótipo foi planejado e executado em etapas sistemáticas, descritas a seguir:

Planejamento do Sistema

- Seleção dos componentes: Arduino Uno, módulo RFID RC522, servo motor SG90, módulo TTP224 4 vias, protoboard e cabos jumpers.

Faculdade de Tecnologia Professor Francisco de Moura – FATEC Jacareí

Disciplina Internet das Coisas e Aplicações

Prof. Henrique Louro

- Definição do PostgreSQL como banco de dados para armazenamento dos acessos.
- Planejamento da integração entre o hardware (Arduino) e o software (Python) via comunicação serial.

Montagem do Hardware

Os componentes foram conectados conforme o esquema de ligações:

- *Módulo RFID RC522*: SDA (D10), SCK (D13), MOSI (D11), MISO (D12), RST (D9), alimentação 3.3V e GND.
- *Servo Motor SG90*: Sinal (D3), VCC (5V), GND.
- *Módulo TTP224 4 Vias*: OUT1 (D4), OUT2 (D5), OUT3 (D6), OUT4 (D7), VCC (5V), GND.

A alimentação foi fornecida pela porta USB do Arduino, com 5V para o servo e o TTP224, e 3.3V para o RFID, utilizando uma protoboard para organizar as conexões.

Desenvolvimento do Software

- *Código Arduino*: Desenvolvido na Arduino IDE, o código realiza a leitura do UID do cartão RFID, validação contra uma lista de UIDs autorizados, leitura de senhas via TTP224, controle do servo motor e envio do UID via porta serial.
- *Código Python*: Um script Python foi criado para ler os dados seriais enviados pelo Arduino e inserir os UIDs no banco de dados PostgreSQL.
- *Configuração do Banco de Dados*: Criado um banco acesso_rfid no PostgreSQL com a tabela acessos, contendo os campos id (SERIAL, chave primária), uid (VARCHAR) e data_hora (TIMESTAMP com valor padrão CURRENT_TIMESTAMP).

Testes e Validação

- Testes foram realizados para verificar a leitura correta dos cartões RFID, a validação de UIDs, a entrada de senhas via TTP224 e o funcionamento do servo motor.
- A comunicação serial entre o Arduino e o Python foi testada para garantir o envio correto dos UIDs.
- Os registros no banco de dados foram monitorados em tempo real utilizando o pgAdmin.

Resultados:

O protótipo foi implementado com sucesso, atendendo aos objetivos estabelecidos. Os principais resultados observados foram:

Faculdade de Tecnologia Professor Francisco de Moura – FATEC Jacareí

Disciplina Internet das Coisas e Aplicações

Prof. Henrique Louro

Autenticação de Usuários

- Cartões RFID com UUIDs autorizados (ex.: "32 5B 03 10" e "E9 1B 71 8B") foram reconhecidos corretamente, resultando na abertura da porta (simulada pelo servo motor).
- Cartões não autorizados não permitiram o acesso, e os eventos foram registrados.
- A entrada de senha via TTP224 foi testada com a sequência correta (1, 3, 2, 4), resultando na abertura da porta. Sequências incorretas foram rejeitadas.

Controle Físico

- O servo motor SG90 moveu-se de 70° (porta fechada) para 160° (porta aberta) ao detectar um UUID autorizado ou uma senha correta, retornando à posição inicial após 3 segundos.

Registro de Acessos

- Os UUIDs lidos foram enviados via comunicação serial para o script Python, que os inseriu na tabela acessos do banco de dados acesso_rfid.
- Exemplo de dados registrados (visualizados no pgAdmin):

id	uid	data_hora
1	32 5B 03 10	2025-04-23 10:15:23
2	E9 1B 71 8B	2025-04-23 10:16:45
3	12 34 56 78	2025-04-23 10:17:10

Faculdade de Tecnologia Professor Francisco de Moura – FATEC Jacareí

Disciplina Internet das Coisas e Aplicações

Prof. Henrique Louro

- Observação: UIDs não autorizados (como "12 34 56 78") também foram registrados, permitindo o rastreamento de tentativas de acesso.

Feedback

- O sistema forneceu feedback via Serial Monitor, exibindo mensagens como "Cartão autorizado!", "Senha correta!" e "Senha incorreta!".

Problema Encontrado

Ao realizar os testes, nos deparamos com o seguinte problema: o professor nos disponibilizou 4 tags RFID e 1 cartão RFID. Com os testes, vimos que 3 das tags funcionam na frequência de 125kHz, assim o sensor não lê. A outra tag e o cartão funcionam em 13,56MHz, o que funciona perfeitamente.

Os testes confirmaram a integração eficiente entre hardware e software, com o sistema funcionando de forma estável e confiável.

Conclusão:

A experiência de desenvolver o sistema de controle de acesso foi enriquecedora, pois envolveu a integração de hardware (Arduino, sensores e atuadores) e software (Python e PostgreSQL), reforçando conceitos de eletrônica, programação e gerenciamento de dados. O protótipo demonstrou ser funcional, com autenticação precisa, controle físico adequado e registro confiável dos acessos.

Reflexão

- O projeto destacou a importância de testes iterativos para identificar e corrigir problemas, como conexões soltas ou configuração incorreta da porta serial.
- A integração entre o Arduino e o Python foi um desafio inicial, mas a solução encontrada (comunicação serial) provou ser eficaz.

Sugestões de Continuidade e Melhorias

- Implementar uma interface web para visualização dos registros de acesso em tempo real.
- Adicionar um buzzer para feedback sonoro em casos de acesso autorizado ou negado.
- Associar nomes de usuários aos UIDs no banco de dados para maior rastreabilidade.

Faculdade de Tecnologia Professor Francisco de Moura – FATEC Jacareí

Disciplina Internet das Coisas e Aplicações

Prof. Henrique Louro

- Incluir um sistema de log para tentativas de acesso não autorizadas, com alertas em tempo real.
- Substituir o servo motor por um mecanismo de trava real para aplicações práticas.
- Adicionar um mecanismo para alterar a senha do TTP224 dinamicamente.

O projeto serviu como uma base sólida para futuros desenvolvimentos em sistemas de controle de acesso, com potencial para escalabilidade e adaptação a diferentes contextos.

Anexos:

Anexo 1: Esquema de Conexões

Alimentação:

5V do Arduino | Fileira positiva da protoboard

GND do Arduino | Fileira negativa da protoboard

Módulo RFID RC522:

Pino RFID | Arduino

SDA | D10

SCK | D13

MOSI | D11

MISO | D12

RST | D9

3.3V | 3.3V (Arduino)

GND | GND (protoboard)

Servo Motor SG90:

Fio Servo | Arduino / Protoboard

Laranja | D3 (PWM)

Vermelho | 5V (protoboard)

Marrom | GND (protoboard)

Faculdade de Tecnologia Professor Francisco de Moura – FATEC Jacareí
Disciplina Internet das Coisas e Aplicações
Prof. Henrique Louro

Módulo TTP224 4 Vias:

Pino TTP224 | Arduino / Protoboard

OUT1 | D4

OUT2 | D5

OUT3 | D6

OUT4 | D7

VCC | 5V (protoboard)

GND | GND (protoboard)

Anexo 2: Código Arduino (rfid_access.ino)

```
#include <Servo.h> #include <SPI.h> #include <MFRC522.h>
```

```
// Defining pins #define SS_PIN 10 #define RST_PIN 9 #define SERVO_PIN 3 #define  
TOUCH_1 4 #define TOUCH_2 5 #define TOUCH_3 6 #define TOUCH_4 7
```

```
// Creating objects MFRC522 rfid(SS_PIN, RST_PIN); Servo servo;
```

```
// Defining valid UUIDs and password String validUUIDs[] = { "32 5B 03 10", "E9 1B 71 8B" };  
const int senhaCorreta[4] = {1, 3, 2, 4}; int entradaUsuario[4]; int posicao = 0;
```

```
// Setup function void setup() { Serial.begin(9600); SPI.begin(); rfid.PCD_Init();  
servo.attach(SERVO_PIN); servo.write(70); pinMode(TOUCH_1, INPUT);  
pinMode(TOUCH_2, INPUT); pinMode(TOUCH_3, INPUT); pinMode(TOUCH_4, INPUT);  
Serial.println("Sistema iniciado. Use cartao RFID ou senha."); }
```

```
// Main loop void loop() { verificarCartao(); verificarTeclado(); delay(100); }
```

```
// Function to verify RFID card void verificarCartao() { if (!rfid.PICC_IsNewCardPresent() ||  
!rfid.PICC_ReadCardSerial()) return; String uidLido = ""; for (byte i = 0; i < rfid.uid.size; i++)  
{ if (rfid.uid.uidByte[i] < 0x10) uidLido += "0"; uidLido += String(rfid.uid.uidByte[i], HEX); if (i  
< rfid.uid.size - 1) uidLido += " "; } uidLido.toUpperCase(); Serial.print("Cartao detectado:  
"); Serial.println(uidLido); for (String uid : validUUIDs) { if (uidLido == uid) {
```

Faculdade de Tecnologia Professor Francisco de Moura – FATEC Jacareí
Disciplina Internet das Coisas e Aplicações
Prof. Henrique Louro

```
Serial.println("Cartao autorizado! Abrindo porta..."); abrirPorta(); break; } }  
rfid.PICC_HaltA(); rfid.PCD_StopCrypto1(); }  
  
// Function to verify password via TTP224 void verificarTeclado() { if  
(digitalRead(TOUCH_1) == HIGH) registrarToque(1, TOUCH_1); if  
(digitalRead(TOUCH_2) == HIGH) registrarToque(2, TOUCH_2); if  
(digitalRead(TOUCH_3) == HIGH) registrarToque(3, TOUCH_3); if  
(digitalRead(TOUCH_4) == HIGH) registrarToque(4, TOUCH_4); }  
  
// Function to register a touch void registrarToque(int valor, int pino) { while  
(digitalRead(pino) == HIGH); entradaUsuario[posicao] = valor; Serial.print("Botao  
pressionado: "); Serial.println(valor); posicao++; if (posicao == 4) { verificarSenha();  
posicao = 0; } }  
  
// Function to verify password void verificarSenha() { Serial.print("Voce digitou: "); for (int i  
= 0; i < 4; i++) { Serial.print(entradaUsuario[i]); Serial.print(" "); } Serial.println(); bool  
correta = true; for (int i = 0; i < 4; i++) { if (entradaUsuario[i] != senhaCorreta[i]) { correta =  
false; break; } } if (correta) { Serial.println("Senha correta! Abrindo porta..."); abrirPorta(); }  
else { Serial.println("Senha incorreta!"); } }  
  
// Function to open the door void abrirPorta() { servo.write(160); delay(3000);  
servo.write(70); Serial.println("Porta trancada."); }
```

Anexo 3: Código Python (log_access.py)

```
import serial  
import time  
import psycopg2
```

Setting up serial connection

```
ser = serial.Serial('COM6', 9600)  
time.sleep(2)
```

Connecting to PostgreSQL database

Faculdade de Tecnologia Professor Francisco de Moura – FATEC Jacaré
Disciplina Internet das Coisas e Aplicações
Prof. Henrique Louro

```
conexao = psycopg2.connect( host="localhost", user="postgres",  
password="sua_senha_aqui", dbname="acesso_rfid" ) cursor = conexao.cursor()
```

Main loop to read serial data and insert into database

```
while True: if ser.in_waiting: uid = ser.readline().decode().strip() print(f"UID recebido: {uid}")  
cursor.execute("INSERT INTO acessos (uid) VALUES (%s)", (uid,)) conexao.commit()  
print(f"Dados inseridos para o UID: {uid}")
```

Notas:

- Substitua COM6 pela porta serial correta.
- Substitua password="sua_senha_aqui" pela senha real do PostgreSQL.

Data:
02 de maio de 2025