Sistema Eletrônico de IoT para a Automação do Controle e Registro de Acesso nos Grupos de Pesquisa da Unipampa

Magno C. Maia, Victor I. A. de Medeiros, Nicolas A. Smaniotto, Adriel Rodrigues, Lucas Compassi-Severo
Grupo de Arquitetura de Computadores e Microeletrônica - GAMA
Universidade Federal do Pampa - Unipampa
Alegrete, RS, Brazil
{magnocostamaia, victormedeiros96, nicolasantoniosmaniotto}@gmail.com, adriel.rodrigues07@hotmail.com,
lucassevero@unipampa.edu.br

***Resumo*—Com o advento da internet das coisas, diversas tecnologias para controle de acesso inteligente vêm sendo estudadas e criadas. O presente trabalho propõe o desenvolvimento de um sistema de IoT para o controle de acesso das salas onde encontram-se os laboratórios de pesquisa da Unipampa. Visando o baixo consumo de potência e baixo custo, o sistema foi implementado com um microcontrolador ESP32, leitor e tags RFID, fechadura eletro-magnética e servidor socket para a gestão do acesso. Uma placa de circuito impresso foi desenvolvida para a validação do protótipo na aplicação desejada.**

***Palavras-chave*—RFID, ESP32, Controle de Acesso, Internet das coisas.**

I. Introdução

A Internet das coisas (IoT, do inglês Internet of Things) é um conceito que é muito discutido nos dias de hoje. Este conceito, entre outras coisas, trata de uma sociedade onde basicamente tudo estará conectado à internet, e a comunicação de máquina para máquina (M2M) terá um aumento considerável. Estas interações entre "máquinas inteligentes" tem por objetivo gerar maior qualidade de vida para toda a sociedade, através da automação e facilidade do gerenciamento e controle de máquinas locais e remotas [1], [2].

As aplicações de IoT são genéricas e podem ser utilizadas em todos os setores da economia. Uma das aplicações mais importantes de IoT é o gerenciamento de pessoas e o controle de acesso em ambientes restritos. Tais sistemas são responsáveis pela identificação de pessoas e liberação de acesso às pessoas autorizadas. Além disso, uma série de registros devem ser adotados para que seja possível realizar um gerenciamento da quantidade de acessos, horários de acesso e atividades realizadas [3].

Na Universidade Federal do Pampa, Campus Alegrete, o gerenciamento de acesso às salas de aulas, laboratórios e escritórios é centralizado em um funcionário terceirizado que exerce a função de porteiro. Este profissional é responsável pela identificação das pessoas, verificação de autorizações de acesso, liberação das chaves e registro escrito das atividades. Tal sistema é trabalhoso e sujeito à inúmeras falhas de segurança, além da dificuldade da verificação manual dos registros quando necessário.

Desta forma, este trabalho tem como principal objetivo o desenvolvimento de um sistema automático para o controle de acesso nas salas do Campus Alegrete da Unipampa, inicialmente focando apenas nos laboratórios onde encontram-se os grupos de pesquisa. O sistema proposto busca contribuir para uma demanda de baixo custo, um sistema de controle de acesso à determinado local, utilizando uma fechadura eletrônica conectada à internet e chaves de acesso com cartões (ou tags) de identificação por rádio frequência (RFID). Com isso, cada usuário do laboratório terá um cartão ou tag RFID que deverá ser aproximada ao leitor próximo à porta de acesso. Se o usuário estiver autorizado, o sistema aciona a fechadura da porta liberando o acesso. Para o gerenciamento e registro do controle de acesso, um servidor foi implementado para verificar a autorização e registrar todas as tentativas de acesso em ambiente remoto.

O presente artigo está organizado da seguinte forma: O sistema proposto é apresentado na Seção II, a Seção III apresenta os resultados experimentais do sistema e a Seção IV conclui o trabalho.

II. Sistema Proposto

O sistema proposto é baseado no diagrama mostrado na Fig. 1 e é comporto por sensores, um controlador, uma fechadura e um servidor. As próximas seções apresentam os detalhes da implementação de cada um dos blocos.

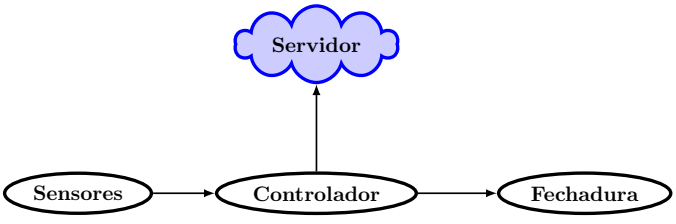
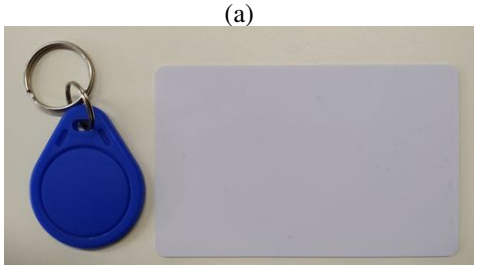


Figura 1. Diagrama de blocos do sistema proposto.

(a)



(b)

Figura 2. Sensor RFID utilizado. (a) Leitor RC522 e (b) cartão e tag RFID.

A. *Sensores*

Como sensor para identificação pessoal, optou-se por utilizar o leitor RFID comercial **MFRC522** que utiliza o padrão MIFARE com tags RFID passivas, conforme mostrado na Fig. 2. Este padrão utiliza uma frequência de 13.56 MHz para o enlace magnético com proximidade sensor-tag de 1 a 3 cm.

B. *Servidor*

O servidor possui as funções de liberação do acesso e o registro de tentativas de acesso. A implementação deste servidor foi baseada no padrão socket simples, utilizando a linguagem de programação Python. A Fig. 3 ilustra o fluxograma presente na lógica de implementação do servidor socket. Neste servidor, a conexão é estabelecida a cada vez que uma tag RFID for aproximada do leitor. Após isso, o sistema compara a ID do cartão com o grupo de usuários autorizados e se autorizado registra no log de autorizações e retorna ao cliente a permissão. Em caso de acesso não autorizado, ocorre o registro de bloqueio e a informação ao cliente sobre a não liberação do acesso.

C. *Fechadura*

Para a liberação mecânica da porta de acesso uma fechadura eletromagnética comercial FE12 AMELCO foi utilizada. Esta fechadura opera com tensão de $12\,V$ e consumo máximo de corrente de $250\,mA$. A Fig. 4 mostra uma imagem da fechadura utilizada.

D. *Controlador*

O controlador é responsável por realizar a interface com o leitor RF, acessar o servidor socket via rede wifi e acionar o

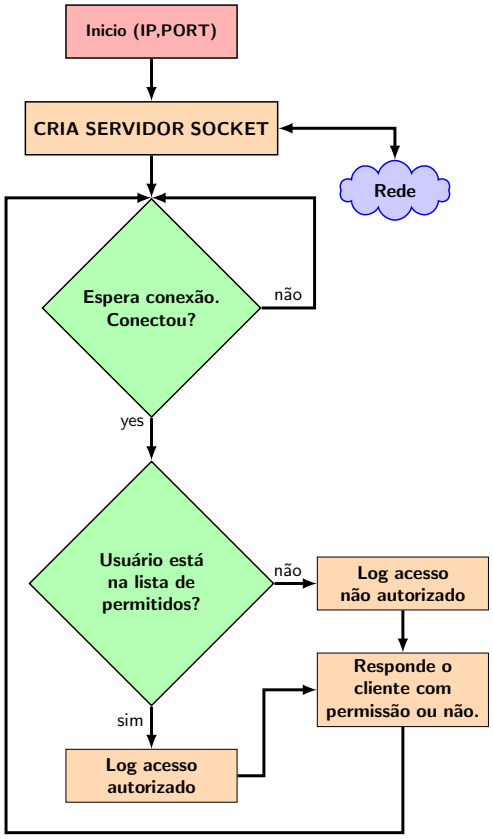


Figura 3. Fluxograma explicativo do servidor socket implementado.

Figura 4. Fechadura eletromagnética utilizada.

relé da fechadura, quando o acesso for liberado. Além disso, este circuito deve prover uma interface visual ao usuário sobre o funcionamento, liberação ou bloqueio de acesso.

Para projetar o controlador foi utilizado o microcontrolador ESP32 da empresa Expressif Systems. Este microcontrolador é um dos circuitos mais utilizados nos dispositivos de IoT, pois apresenta um baixo consumo de potência, baixo custo e possui uma série de funcionalidades integradas ao microcontrolador. Dentre as funcionalidades destacam-se a presença de transceptores de Wifi (IEEE 802.11 b/g/n) e Bluetooth (Clássico e BLE) integrados. Neste projeto o transceptor de wifi foi utilizado para conectar o controlador ao servidor socket.

Para programar o microcontrolador ESP32, utilizou-se o "Arduino SDK", com o software "Arduino IDE". O programa desenvolvido utilizou como base bibliotecas de funções disponíveis na comunidade Arduíno. O código desenvolvido é mostrado abaixo no apêndice deste trabalho.

Para conectar o ESP32 à rede wifi e utilizar o módulo

MFRC522, utilizou-se a bibliotecas “MFRC522.h” e “WiFi.h”. Após isso, definiu-se os pinos utilizados, parâmetros como SSID e senha da rede, e inicialização da biblioteca de RFID. Criou-se um método de validação do cliente (“cliente_valido”), que conecta ao servidor socket, envia a chave que representa o cliente, recebe a resposta do servidor e, se o cliente tem permissão de acesso ou não, retorna 1 caso permita o acesso ou 0 caso o acesso tenha sido negado. No método “setup”, é configurado os pinos de GPIO e conecta-se à rede WiFi. No método “Loop” verifica-se a presença de cartão no leitor, se houver cartão, faz-se a leitura, chama-se o método de validação. Se “cliente_valido” retornar 1, aciona-se o led verde, assim como o relé de liberação da porta. Caso contrário, apenas aciona-se o led vermelho, como resposta visível de não autorização para o cliente.

III. Resultados Experimentais

Para a validação experimental, o sistema completo foi montado em matriz de contatos, conforme mostra a Fig. 5. Para realizar o teste do sistema, três cartões RFID foram utilizados, sendo dois com acesso autorizado e um não autorizado. Os cartões RFID são identificados pelo seu código de identificação única (UID) que pode ser atribuído para representar a identificação do usuário.

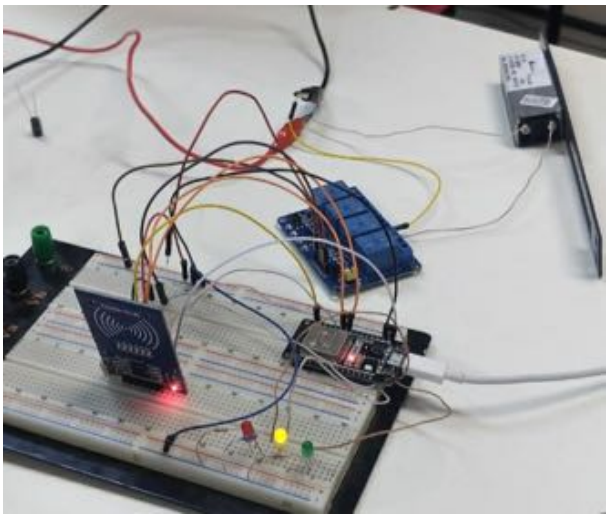


Figura 5. Montagem em matriz de contatos para a validação inicial do sistema.

A Fig. 6 mostra os logs obtidos durante os primeiros testes. No Log de acesso liberado estão salvos todos os acessos dos cartões com UID ”9B-13-24-1B”e ”61-3F-4F-D3”que possuem acesso liberado. Por outro lado, o Log de bloqueio apresenta as tentativas de acesso do cartão com UID ”D7-A3-10-1C”que não foi cadastrado como usuário de acesso liberado no servidor.

Após a validação, uma placa de circuito impresso (PCI) foi projetada para o sistema de controle de acesso. A Fig. 7 mostra os lados superior e inferior da placa projetada. Na parte superior encontra-se o leitor RFID e três LEDs nas cores amarela, verde e vermelha. Tais LEDs são adotados para

Log de acesso

```
9B,13,24,1B;2019-08-15 21:04:27.349012;
61,3F,4F,D3;2019-08-15 21:04:31.312783;
9B,13,24,1B;2019-08-15 21:04:38.436929;
61,3F,4F,D3;2019-08-15 21:40:25.475257;
61,3F,4F,D3;2019-08-15 21:40:38.087476;
61,3F,4F,D3;2019-08-15 21:42:20.321935;
61,3F,4F,D3;2019-08-15 21:42:45.062603;
61,3F,4F,D3;2019-08-15 21:43:52.270922;
61,3F,4F,D3;2019-08-15 21:44:26.380288;
9B,13,24,1B;2019-08-15 21:45:21.162493;
```

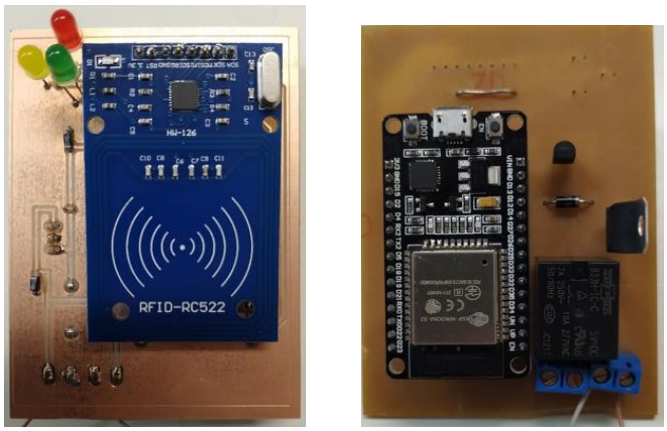
Log de bloqueio

```
D7,A3,10,1C;2019-08-16 16:32:19.543182;
D7,A3,10,1C;2019-08-16 16:37:11.393128;
```

Figura 6. Log do sistema obtidos nos primeiros testes.

sinalizar a operação, a liberação e o bloqueio do sistema de acesso, respectivamente. No Lado inferior da PCI encontra-se o controlador ESP32 e circuitos extras utilizados para a alimentação e regulação de tensão e relé de acionamento da fechadura.

Esta PCI foi projetada para ser integrada em uma caixa plástica onde o sistema pode ser facilmente instalado.



Lado Superior Lado Inferior

Figura 7. Placa e circuito impresso desenvolvida.

IV. Conclusão

Este artigo buscou demonstrar o funcionamento básico de um sistema de controle de acessos digital e de baixo custo, o qual foi testado com sucesso e já se demonstra funcional. Porém, como o intuito do trabalho foi apenas de demonstração e aprendizado, futuramente deseja-se desenvolver uma aplicação mais segura, com criptografia para proteger os dados, e utilizando a memória das “tags”, para garantir melhor veracidade da informação de identificação [4], [5]. Assim como, para os futuros trabalhos, deseja-se utilizar o SDK-IDF, por ser a linguagem padrão oficial, com suporte e documentação [6].

Além disso, como trabalho futuro, espera-se instalar o sistema proposto em um grupo de pesquisa do Campus Alegrete para validação do sistema. Tal instalação servirá de projeto piloto para a expansão nos demais ambientes do Campus.

Referências

[1] A. H. Ngu, M. Gutierrez, V. Metsis, S. Nepal, and Q. Z. Sheng, "Iot middleware: A survey on issues and enabling technologies," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 1–20, Feb 2017.

[2] B. Kang, D. Kim, and H. Choo, "Internet of everything: A large-scale autonomic iot gateway," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 3, no. 3, pp. 206–214, July 2017.

[3] O. Novo, "Blockchain meets iot: An architecture for scalable access management in iot," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1184–1195, April 2018.

[4] M. Frustaci, P. Pace, G. Aloi, and G. Fortino, "Evaluating critical security issues of the iot world: Present and future challenges," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2483–2495, Aug 2018.

[5] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying iot security: An exhaustive survey on iot vulnerabilities and a first empirical look on internet-scale iot exploitations," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2702–2733, thirdquarter 2019.

[6] E. Systems. (2016) ESP32 Series data sheet. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

Apêndice
Código Implementado no Controlador com ESP32

```
#include <MFRC522.h>
#include <WiFi.h>
#define SDA 21
#define RST 22
#define gLed 12
#define rLed 32
#define yLed 14
#define Rele 27
const char* ssid = "NomeDaRede";
const char* password = "senha";
const uint16_t port = 8000;
const char * host = "192.168.137.1";
MFRC522::MIFARE_Key key;
MFRC522::StatusCode status;
MFRC522 mfrc522(SDA,RST);
char buffer[11];
int cliente_valido(char *buffer)
{
  WiFiClient client;
  if (!client.connect(host, port)) {
    Serial.println(F("Connection to host failed"));
    delay(1000);
    return 0;
  }
  client.print(buffer);
  String receive_data = client.readStringUntil('\n')
      ;
  client.stop();
  if (receive_data == "1")
  {
    return 1;
  }
  else {
    return 0;
  }
}
void setup() {
  Serial.begin(9600);
  SPI.begin(); // Init SPI bus
  pinMode(gLed, OUTPUT);
  pinMode(rLed, OUTPUT);
  pinMode(yLed, OUTPUT);
  pinMode(Rele, OUTPUT);
  digitalWrite(Rele, HIGH);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
  }
  mfrc522.PCD_Init();
}
void loop()
{
  if ( ! mfrc522.PICC_IsNewCardPresent())
  {
    digitalWrite(yLed, HIGH);
    return;
  }
  if ( ! mfrc522.PICC_ReadCardSerial())
  {
    return;
  }
  digitalWrite(yLed, LOW);
  sprintf(buffer, "%X,%X,%X,%X", mfrc522.uid.uidByte
      [0], mfrc522.uid.uidByte[1], mfrc522.uid.
      uidByte[2], mfrc522.uid.uidByte[3]);
  if (cliente_valido(buffer)) {
    for (int i=0;i<10;i++){
      digitalWrite(gLed, HIGH);
      digitalWrite(Rele, LOW);
      digitalWrite(yLed, LOW);
      delay(100);
      digitalWrite(gLed, LOW);
      digitalWrite(Rele, HIGH);
    }
  }
  else {
    digitalWrite(rLed, HIGH);
    digitalWrite(yLed, LOW);
    delay(2000);
    digitalWrite(rLed, LOW);
  }
  mfrc522.PICC_HaltA();
  mfrc522.PCD_StopCrypto1();
}
```