



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SÃO PAULO
CAMPUS BIRIGUI**

Aline Bertolazo dos Santos
Isadora Disposti Bueno dos Santos

Porta com senha - Controle de acesso aos Laboratórios

**BIRIGUI- SP
2022**



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SÃO PAULO
CAMPUS BIRIGUI**

Requisitos Funcionais:

Considerando o ambiente onde deseja-se controlar o acesso de pessoas aos laboratórios do campus Birigui, busca-se resolver este problema utilizando uma porta com verificação de senha. Devido a dificuldade de acesso por meio dos funcionários e professores, cada um teria seu próprio cadastro por meio da Matrícula e uma Senha. Utilizando um teclado 4x3 contendo os numéricos (TECLADO), botão utilizado para apagar algum dado digitado incorretamente (DEL), um para realizar a verificação de dados da senha (ENTER) e caso nada seja pressionado estará no modo inativo (INATIVO). Após isso será ativado um solenóide (SOL) por meio de um pulso e a fechadura se abrirá. As informações serão exibidas para o usuário através de um display lcd.

Requisitos de Comunicação:

Para estabelecer a senha a ser usada para abrir a trava eletrônica a mesma será definida previamente através de um sistema front-end, o mesmo ocorrerá com a matrícula, sendo registradas em um sistema a parte e enviadas ao sistema da trava quando requisitado por este. Após receber a matrícula informada, chama-se a comunicação com o servidor que envia a senha por meio de callback e receberá a senha digitada. Após isso no programa é realizada a verificação se os dados são válidos.

Requisitos de usuário:

O usuário será capaz de informar a matrícula e a senha, sendo visualizado pelo monitor serial. Após a matrícula ser enviada para um banco de dados (provisoriamente o Broker), o microcontrolador receberá a senha condizente com a matrícula informada e será pedido a senha do usuário, tendo o mesmo 20 segundos para informá-la, do contrário o microcontrolador irá cancelar a operação e voltará a pedir pela matrícula. Caso a matrícula e a senha estejam corretas a porta será aberta. Mesmo que a matrícula informada não esteja cadastrada no banco de dados, será pedido uma senha e só após isso é informado que os dados estão incorretos, por motivos de segurança.

Requisitos adiados:

- Implementação de um lcd funcional para a melhor visualização do usuário;
- Implementação de um banco de dados.

Definição dos I/Os e níveis lógicos:

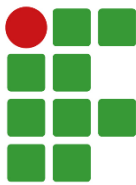
Como será utilizado um teclado matricial cada linha e cada coluna tem um TECLADO específico que é enviado no barramento de 7 pinos.

Os dados informados no TECLADO podem ser {0,1,2,3,4,5,6,7,8,9,ENTER (#) ou DEL(*)}, quando TECLADO não está pressionado é igual a INATIVO.

SOL - solenóide (Output digital)

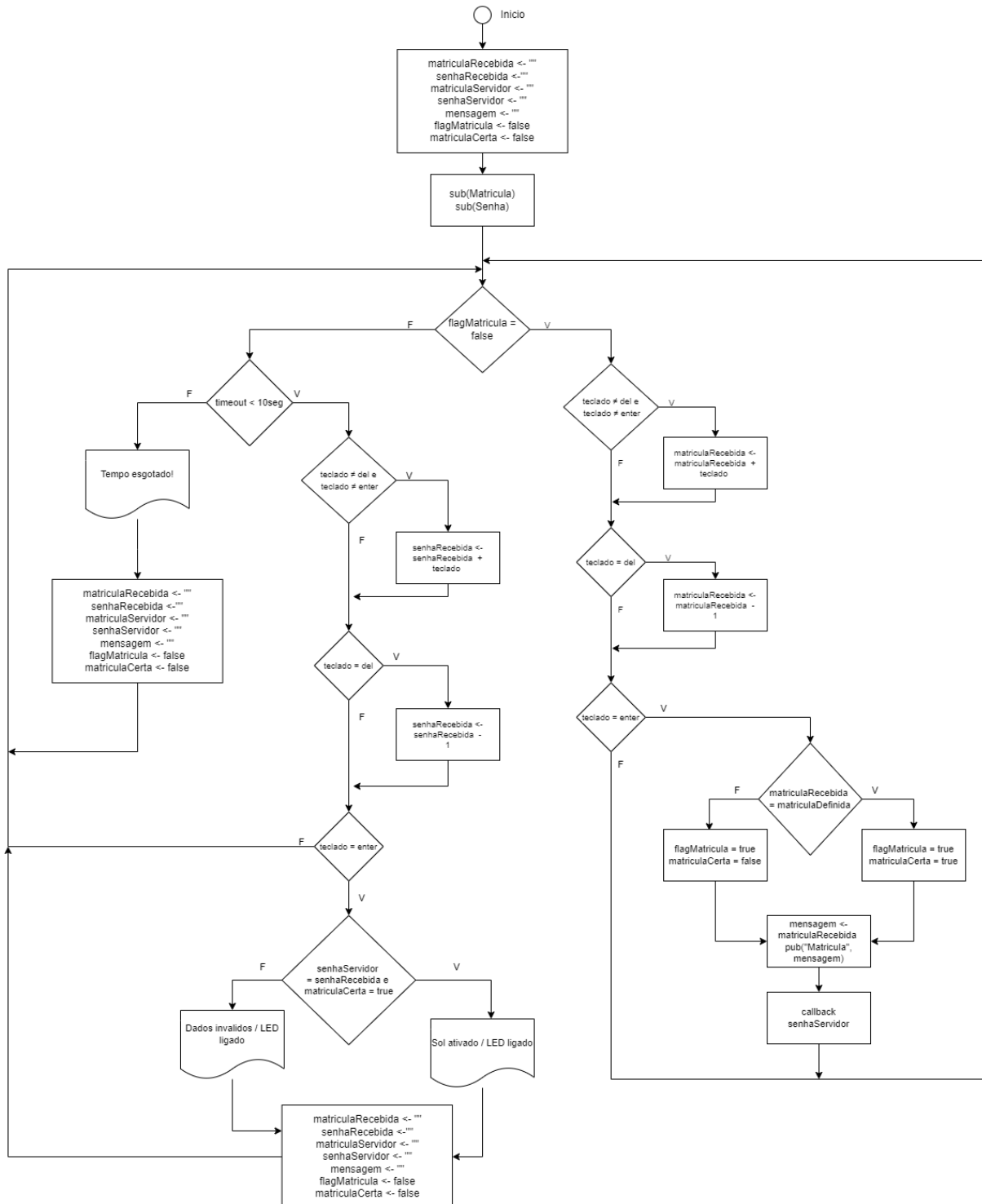
pulso 1 - 300 milissegundos (Senha correta)

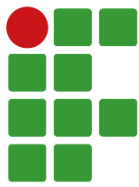
0 - não-ativo (Senha incorreta)



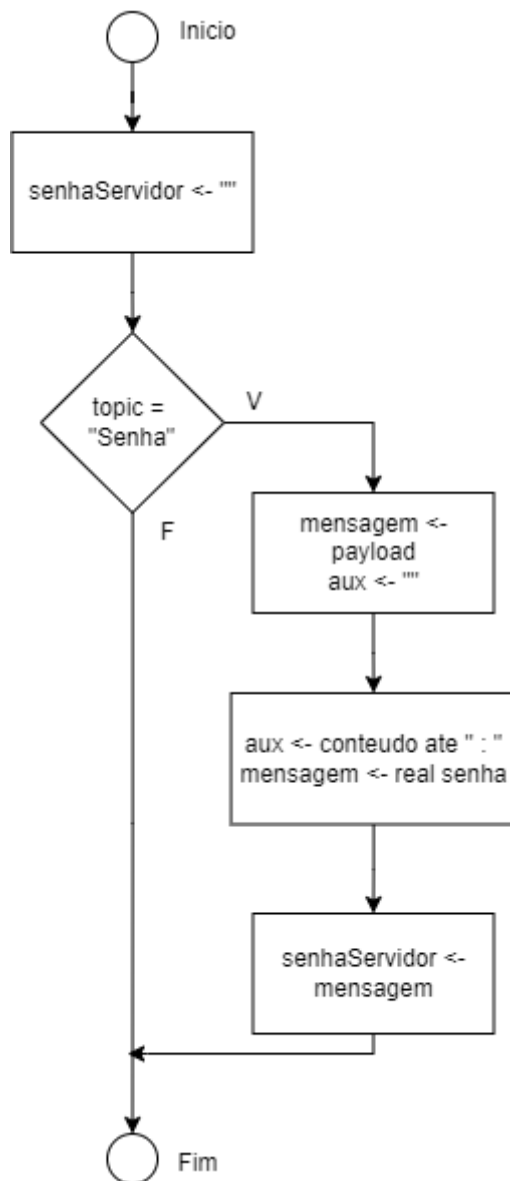
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SÃO PAULO
CAMPUS BIRIGUI

Modelo lógico



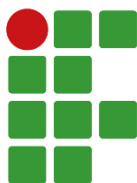


Modelo lógico Callback



Plataforma alvo

NodeMCU ESP32



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SÃO PAULO
CAMPUS BIRIGUI**

Mapeamento físico (NodeMCU ESP32)

Descrição	Pino	I/O
Coluna 1	GPIO5	Entrada Digital
Coluna 2	GPIO4	Entrada Digital
Coluna 3	GPIO2	Entrada Digital
Linha 1	GPIO27	Saída Digital
Linha 2	GPIO14	Saída Digital
Linha 3	GPIO12	Saída Digital
Linha 4	GPIO13	Entrada Digital
SOL	GPIO15	Saída Digital

Definição dos níveis lógicos do teclado:

Descrição	Pino	Ativo	Não Ativo
Coluna 1	GPIO5	1	0
Coluna 2	GPIO4	1	0
Coluna 3	GPIO2	1	0
Linha 1	GPIO27	1	0
Linha 2	GPIO14	1	0
Linha 3	GPIO12	1	0
Linha 4	GPIO13	1	0
SOL	GPIO15	1	0

Funcionamento: A partir das definições de colunas e linhas, o firmware deve por meio de uma matriz de valores, mapear as teclas do teclado matricial.



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SÃO PAULO
CAMPUS BIRIGUI**

Código:

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <LiquidCrystal_I2C.h>

// Definir a saída Solenoide
#define SOL 15

// Definir a quantidade de linhas e colunas
#define Linhas 4
#define Colunas 3

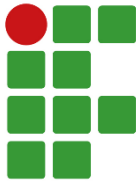
// Definição dos parâmetros de rede
#define SSID "Fit Leon"
#define PASS "123456789"

//Variáveis para o display
int lcdColumns = 16;
int lcdRows = 2;

LiquidCrystal_I2C lcd(0x27, lcdColumns, lcdRows);
// Definir os caracteres referentes ao teclado --
/* - representa DEL          //# - representa ENTER
char mapaTeclas[Linhas][Colunas] = {
    { '1', '2', '3' },
    { '4', '5', '6' },
    { '7', '8', '9' },
    { '*', '0', '#' }
};

//Definir Variáveis que serão utilizados
String matriculaRecebida; //Senha que recebe o conteúdo do teclado
String senhaRecebida;     //Matricula recebida do teclado
String senhaServidor;     //Senha devolvida pelo servidor
String mensagem;          //Variável que tratará o conteúdo recebido
do servidor

//Matricula definida para comparação
String matriculaDefinida = "587";
```



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SÃO PAULO
CAMPUS BIRIGUI**

```
// Flags para garantir o envio correto para o servidor
bool flagMatricula = false;
bool matriculaCerta = false;
bool flagTimeOut = false;

//Variaveis para conexão com o servidor
WiFiClient wifiCliente;
PubSubClient cliente;

//Variavel para timeOUT

int t1;
int t2;
String senhaCripto;

// Definir os pinos das linhas e colunas
byte pinos_linha[Linhas] = { 27, 14, 12, 13 };
byte pinos_coluna[Colunas] = { 5, 4, 2 };

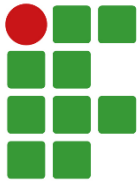
// TODO: Funcao de tratamento dos dados recebidos via mqtt
void callback(const char *topic, byte *payload, unsigned int length) {

    senhaServidor = "";
    if ((String)topic == "Senha") {

        for (int i = 0; i < length; i++) {
            mensagem += (char)payload[i];
        }

        // Variaveis auxiliares para o recebimento das informacoes via mqtt
        String aux = "";    // Guarda as substrings
        String param = "";  // Guarda os parametros a serem alterados
        //Colocar a mensagem na senha do servidor

        aux = mensagem.substring(0, mensagem.indexOf(":"));
        mensagem = mensagem.substring(mensagem.indexOf(":") + 1, length);
        senhaServidor = mensagem;
        Serial.println("Senha: " + senhaServidor);
    }
}
```



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SÃO PAULO
CAMPUS BIRIGUI**

```
void setup() {

    Serial.begin(9600);

    // Iniciar o LCD
    lcd.init();
    lcd.begin(16, 2);
    lcd.backlight();

    while (!Serial)
        ;
    // Definir os pinos da linha como output
    for (auto pin : pinos_linha)
        pinMode(pin, OUTPUT);

    // Definir os pinos da culuna como input
    for (auto pin : pinos_coluna)
        pinMode(pin, INPUT);

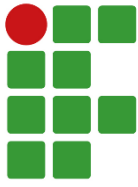
    t1 = millis();

    // Definir o Solenoid como output e setar como low
    pinMode(SOL, OUTPUT);
    digitalWrite(SOL, LOW);

    // Iniciar conexao wifi
    Serial.println();
    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(SSID);
    WiFi.begin(SSID, PASS);

    // Verificar conexao wifi
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    // Exibir mensagem de sucesso
```

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SÃO PAULO
CAMPUS BIRIGUI**

```
Serial.println("");
Serial.println("WiFi connected.");

// Conectar ao HiveMQ
cliente.setClient(wifiCliente);
cliente.setServer("broker.hivemq.com", 1883);
cliente.connect("Esp32Master", "Usuario", "Senha");
cliente.subscribe("Senha");
cliente.subscribe("Matricula");
cliente.setCallback(callback);

matriculaRecebida = "";
senhaRecebida = "";
senhaCripto = "";
}

void loop() {

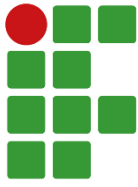
    lcd.setCursor(0, 0);
    lcd.print("                ");
    t2 = millis();

    //Iniciar Variaveis
    mensagem = "";

    cliente.loop();

    // para da linha
    for (int i = 0; i < 4; i++) {
        for (int j = 0; j < 4; j++) {
            digitalWrite(pinos_linha[j], LOW);
        }
        digitalWrite(pinos_linha[i], HIGH);
        // para cada coluna
        for (int j = 0; j < 3; j++) {
            if (digitalRead(pinos_coluna[j]) == HIGH) {
                const char teclado = mapaTeclas[i][j];

                // Serial.print("Teclado: ");
                //Serial.println(teclado);
            }
        }
    }
}
```



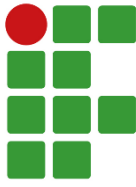
**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SÃO PAULO
CAMPUS BIRIGUI**

```
        if (!flagMatricula) { //Conferir se a matricula não foi
informada

            if ((teclado != '*' ) && (teclado != '#')) { // Se a tecla
for precionada mas não for DEL ou ENTER
                matriculaRecebida += teclado; // Matricula
recebe o conteudo do teclado
            }
            if (teclado == '*') {
// Se teclado for igual a DEL
                matriculaRecebida.remove(matriculaRecebida.length() - 1,
1); // Matricula perde o ultimo caracter adicionado
            }
            if (teclado == '#') { // Se o ENTER for precionado
                mensagem = ("Matricula: " + String(matriculaRecebida));
                t1 = millis();
                //Comparação da matricula recebida com a definida
                if (matriculaRecebida.compareTo(matriculaDefinida) == 0) {
                    cliente.publish("Matricula", mensagem.c_str());
//Publicar a matricula
                    flagMatricula = true;
//Confirmar que a matricula foi recebida
                    matriculaCerta = true;
                } else {
                    cliente.publish("Matricula", mensagem.c_str());
//Publicar a matricula
                    flagMatricula = true;
//Confirmar que a matricula foi recebida
                    matriculaCerta = false;
                }
            }
        } else { // Conteudo da Senha
            //TimeOut de 20 segundos para o recebimento da senha
            if ((t2 - t1) <= 20000) {

                if ((teclado != '*' ) && (teclado != '#')) {
                    //Serial.println(senhaRecebida);
                    senhaRecebida += teclado; // senha recebe o conteudo do
teclado

                    senhaCripto += "*";
                }
            }
        }
    }
}
```



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SÃO PAULO
CAMPUS BIRIGUI**

```
// Se teclado for igual a DEL
if (teclado == '*') {
    // Senha perde o ultimo caracter adicionado
    senhaRecebida.remove(senhaRecebida.length() - 1, 1);
    senhaCripto.remove(senhaCripto.length() - 1, 1);
}

if (teclado == '#') {
    mensagem = ("Senha: " + senhaRecebida);
    if ((senhaRecebida.compareTo(senhaServidor) == 0) &&
(flagMatricula == true) && (matriculaCerta == true)) {
        digitalWrite(SOL, HIGH);
        Serial.println("ABERTO ");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("ABERTO!");
        matriculaRecebida = "";
        senhaRecebida = "";
        delay(1000);
        digitalWrite(SOL, LOW);
        t1 = millis();

    } else {
        delay(1000);
        Serial.print("Dados Invalidos!\n");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Dados Invalidos!");
        matriculaRecebida = "";
        senhaRecebida = "";
        flagMatricula = false;
        t1 = millis();
    }
}

} else {

    Serial.println("Tempo Esgotado!");
    matriculaRecebida = "";
    senhaRecebida = "";
}
```



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SÃO PAULO
CAMPUS BIRIGUI**

```
        flagMatricula = false;
        flagTimeOut = false;
        t1 = millis();
    }
}
//lcd.clear();
lcd.setBacklight(HIGH);
lcd.setCursor(0, 0);
lcd.print("Matrícula: " + matriculaRecebida);
lcd.setCursor(0, 1);
//lcd.print("Senha: " + senhaRecebida);
Serial.print("Matricula: ");
Serial.println(matriculaRecebida);
Serial.print("Senha: ");
Serial.println(senhaCripto);
while (digitalRead(pinos_coluna[j]) == HIGH)
    ;
}
}
}
}
```