

# Representação de Automação Residencial

Alunos: Elildes e Magdiel

# Tópicos

1. Introdução;
2. Arquitetura IoT;
3. Circuito Eletrônico;
4. Software Embarcado;
5. Montagem;
6. Resultados;
7. Conclusões.

### Problemática

- Identificar vazamento de gás;
- Saber se uma porta está aberta;
- Receber dados de temperatura e umidade do ambiente;
- Ligar lâmpadas à distância;
- Abrir e fechar o portão de uma garagem automaticamente.

### Solução Proposta

- Usar sensores que possam captar dados do ambiente e atuadores que respondam a determinadas informações;
- Poder visualizar determinadas informações por um aplicativo através de um smartphone.

### Breve introdução do projeto

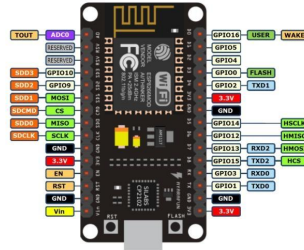
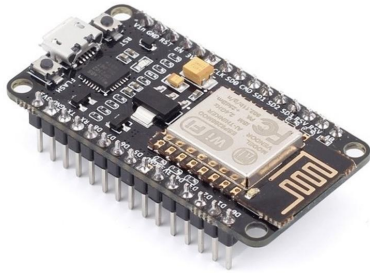
Para o projeto, foi usado microcontroladores (ESP8266 e Arduino Nano) para receber informações dos sensores e responder segundo esses dados. Também foi criado um aplicativo para visualizar essas informações e, além disso, ligar e desligar leds.

Com microcontrolador ESP8266 foram conectados o sensor de gás, de umidade e temperatura, o módulo reed switch, leds e buzzer. Esses, através da internet, farão comunicação com o aplicativo.

Usando Arduino Nano, podemos fazer uma simulação de levantar e abaixar uma cancela (uso do servo motor, PIR e leds).

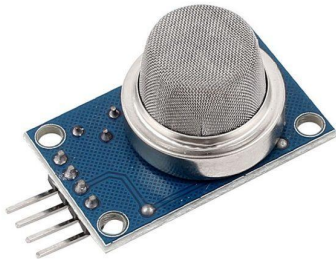
# Arquitetura IoT

## - ESP8266



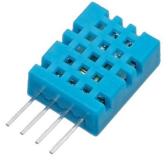
ESP8266 é um Circuito Integrado (CI) caracterizado como um microcontrolador que conecta-se à internet pelo WiFi e ele está acoplado à placa NodeMCU.

## - MQ-2



O sensor MQ-2 é responsável por identificar concentrações de certos gases/vapores de GLP (gás de cozinha), Metano, Propano, Butano, Hidrogênio, Álcool e fumaça de cigarro.

- DHT 11



Com ele, pode-se medir temperaturas entre 0 a 50° Celsius com uma precisão de 2 graus e umidade entre 20% a 90% com precisão de 5%.

- PIR



Esse sensor capta radiação infravermelha com comprimento de onda em torno de 10 micrômetros, que equivale aproximadamente à temperatura corporal de animais de sangue quente em geral.

- Buzzer



É um dispositivo eletrônico que gera frequências sonoras, usado em alarmes, por exemplo. Ele gera frequências que estão na faixa de 1 a 7 khz.

- Micro Servo Motor



É um dispositivo eletromecânico usado para movimentar, com precisão, um objeto, permitindo-o girar em ângulos ou distâncias específicas, com garantia de posicionamento e garantia de velocidade.

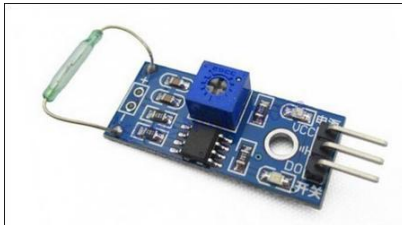


- LED



LED ou é um diodo semicondutor que quando energizado emite luz. A cor da luz do LED depende do cristal e da impureza de dopagem com que o componente é fabricado.

- Módulo Reed Switch



Reed-Switch é composto de uma cápsula de vidro e de duas lâminas de um material ferromagnético. As duas lâminas são colocadas muito próximas, sem que haja contato entre elas. Para haver contato elétrico entre as lâminas, é necessário induzir a magnetização delas, fazendo com que elas se atraiam magneticamente.

- Jumper



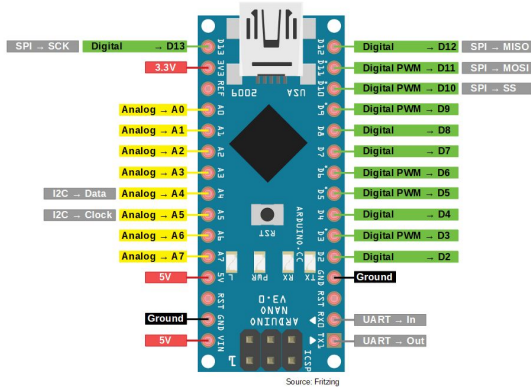
Faz as conexões entre microcontrolador e os componentes eletrônicos.

- Protoboard



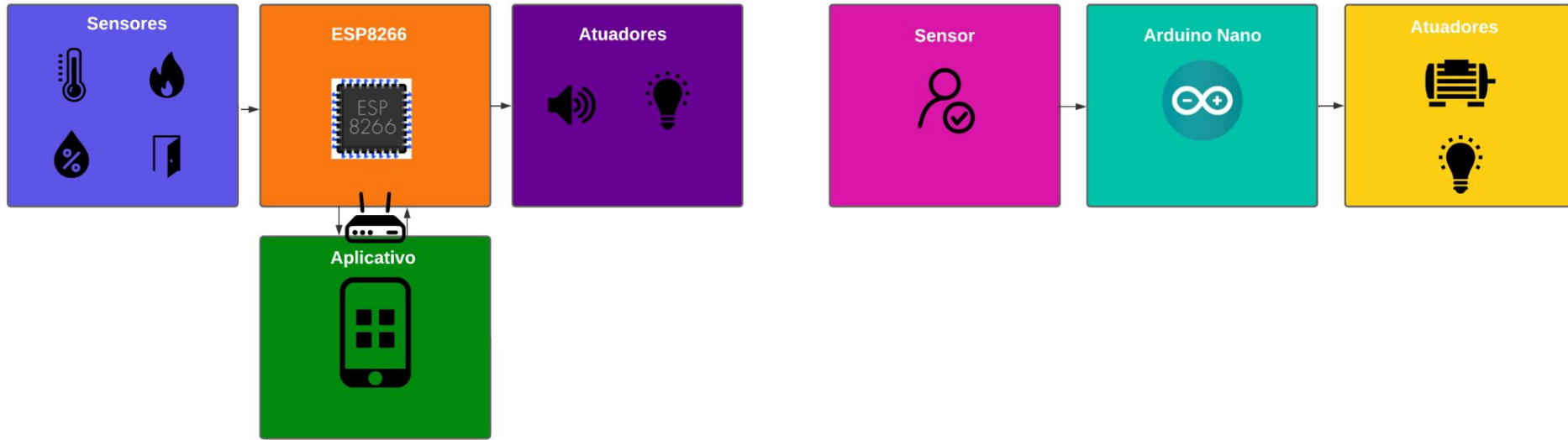
Serve como um instrumento de teste: antes de soldar na placa, você testa no protótipo, e, se tudo der certo, parte para o projeto final.

## - Arduino Nano



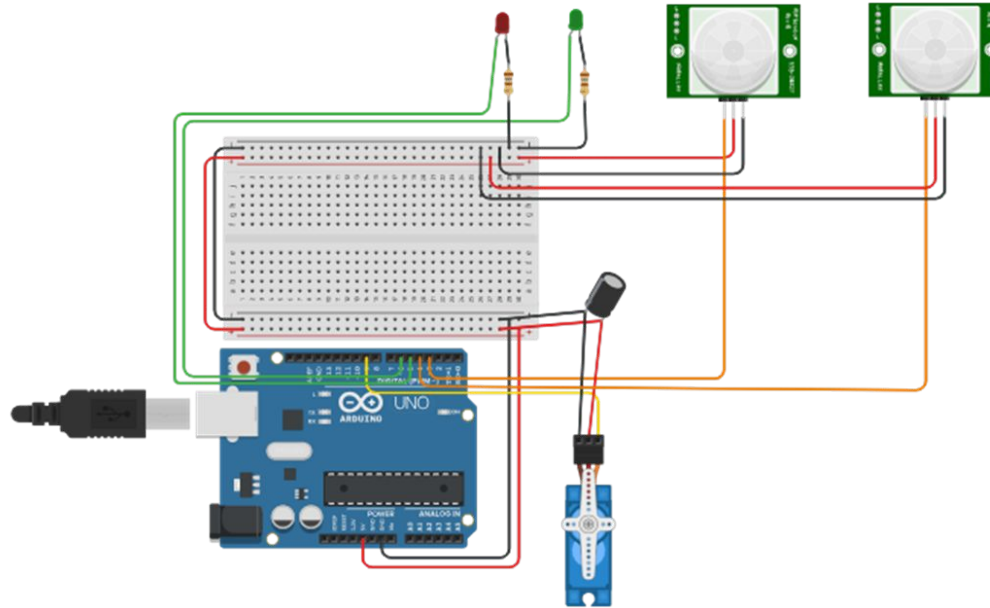
Esta versão do arduino é uma placa bem versátil, pequena. Compartilha algumas funcionalidades com o Arduino Uno.

# Arquitetura IoT

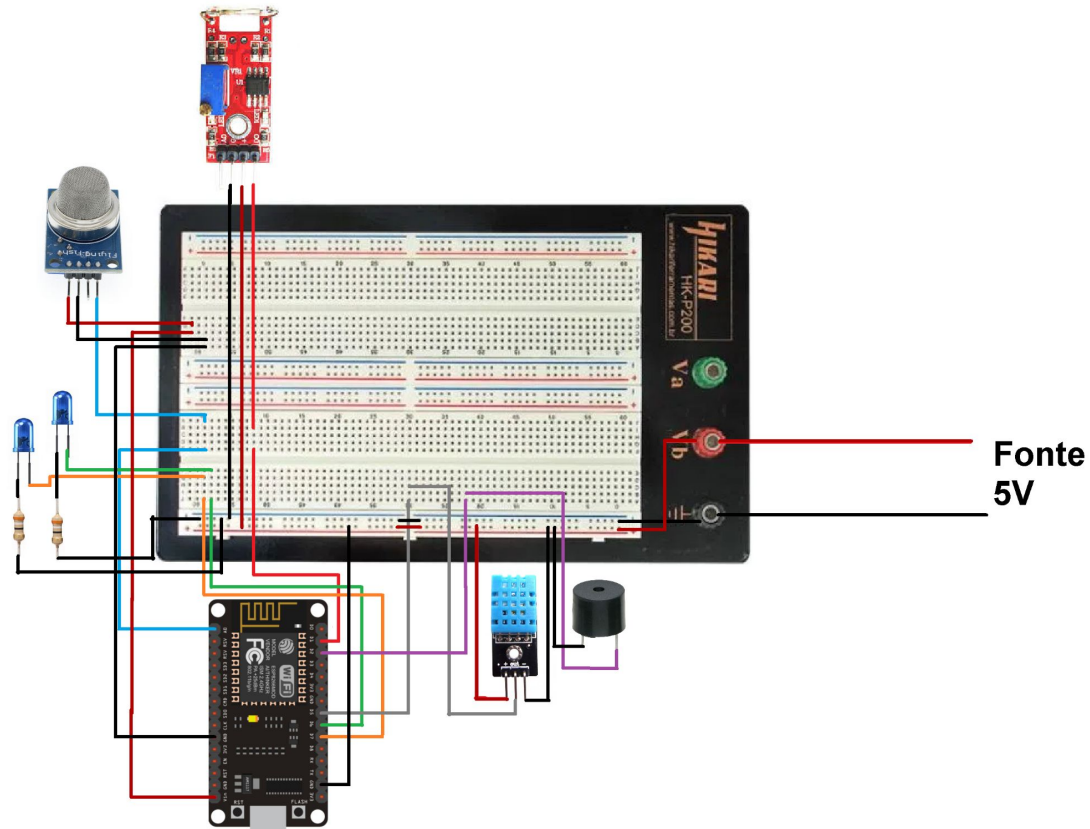


# Circuito Eletrônico - Arduino Nano

## Abertura e fechamento automático de cancela



# Circuito Eletrônico - ESP8266



# Software Embarcado - ESP8266

```
1  #include <ESP8266WiFi.h>
2  #include "DHTesp.h"
3  // #include "Servo.h"
4
5
6
7  #define LED 12
8  #define LED2 13
9  // #define PIR 15
10 #define BUZ 4
11 #define REED 5
12
13 DHTesp dht;
14 // Servo servoD1;
15
16 const char* ssid = "";
17 const char* password = "1";
18
19 WiFiServer server(80);
20 String currentLine = "";
21 void setup() {
22     Serial.begin(115200);
23
24     dht.setup(14, DHTesp::DHT11); // DHT11
25     pinMode(LED, OUTPUT); // LED
26     pinMode(LED2, OUTPUT); // LED2
27     // pinMode(PIR, INPUT); // Sensor de presença
```

```
28     pinMode(BUZ, OUTPUT);
29     pinMode(REED, INPUT);
30     // servoD1.attach(D1);
31
32     delay(10);
33
34
35     Serial.println();
36     Serial.println();
37     Serial.println("Conectando à");
38     Serial.println(ssid);
39
40     WiFi.begin(ssid, password);
41
42     while (WiFi.status() != WL_CONNECTED) {
43         delay(500);
44         Serial.println(".");
45     }
46
47     IPAddress ip(192,168,100,98);
48     IPAddress gateway(192,168,100,67);
49     IPAddress subnet(255,255,255,0);
50
51     WiFi.config(ip, gateway, subnet);
52
53     Serial.println("");
54     Serial.println("WiFi conectado");
```

# Software Embarcado - ESP8266

```
55 Serial.println("Endereço IP");
56 Serial.println(ip);
57
58 server.begin();
59
60 }
61
62 int value = 0;
63
64 void loop() {
65
66     WiFiClient client = server.available();
67
68     String u = String(dht.getHumidity()); //Declaração da variável de umidade
69     String t = String(dht.getTemperature()); //Declaração da variável de temperatura
70     float g = analogRead(A0);
71     //Verifica o status do sensor de presença:
72
73     if(client) {
74         Serial.println("New Client.");
75         currentLine = "";
76         while(client.connected()){
77             if(client.available()){
78                 char c = client.read();
79                 Serial.write(c);
80
81                 if(c == '\n') {
```

```
84         if(currentLine.length() == 0){
85
86             client.println("HTTP/1.1 200 OK");
87             client.println("Content-type:text/html");
88             client.println();
89
90             //Para LEDs diferentes
91             //client.println("<a href=\""/bot1\""></a><br>");
92             //client.println("<a href=\""/botd\""></a><br>");
93
94             //client.println("<a href=\""/bot2l\""></a><br>");
95             //client.println("<a href=\""/bot2d\""></a><br>");
96
97             client.print("Umidade: " + u); //String de umidade
98             client.print("Temperatura: " + t); //String de temperatura
99             client.print("Gas: " + String(g));
100             //client.print("Presenca: ");
101             //int pres = digitalRead(PIR);
102             int por = digitalRead(REED);
103             if(por == 0){
104                 client.print("SPor");
105             }else{
106                 client.print("NPor");
107             }
108
109             client.println();
110         }
```



# Software Embarcado - ESP8266

```
111         break;
112     }else{
113         currentLine = "";
114     }
115
116     } else if(c != '\r'){
117         currentLine += c;
118     }
119
120     //Para LEDS diferentes - verifica a URL enviada como requerimento do cliente para o servidor
121     if(currentLine.endsWith("GET /botl")){
122         digitalWrite(LED, HIGH);
123     }else if(currentLine.endsWith("GET /botd")){
124         digitalWrite(LED, LOW);
125     }
126
127     if(currentLine.endsWith("GET /bot2l")){
128         digitalWrite(LED2, HIGH);
129     }else if(currentLine.endsWith("GET /bot2d")){
130         digitalWrite(LED2, LOW);
131     }
132
133
134
135
136 }
137
```

```
138     }
139
140     client.stop();
141     Serial.println("Client Disconnected.");
142 }
143
144 if(g > 600)
145 {
146     digitalWrite(BUZ, HIGH);
147 }
148 else
149 {
150     digitalWrite(BUZ, LOW);
151 }
152
153 delay(10);
154
155
156 }
157
158
```

# Software - Arduino Nano

```
1  #include <Servo.h>
2
3  Servo myservo;      // create servo object to control a servo
4
5  int pos = 0;        // variable to store the servo position
6
7  int pinLedRed = 5;   //Pino ligado ao led vermelho
8  int pinLedGreen = 6; //Pino ligado ao led azul
9  int pinPir = 3;      //Pino ligado ao sensor PIR
10 int acionamento;     //Variavel para guardar valor do sensor
11
12 void setup()
13 {
14     // servo
15     myservo.attach(9); // attaches the servo on pin 9 (PWM) to the servo object
16     myservo.write(0);  // initial position of servo
17     delay(3000);
18     // pir
19     pinMode(pinLedRed, OUTPUT); //Define pino como saida
20     pinMode(pinLedGreen, OUTPUT); //Define pino como saida
21     pinMode(pinPir, INPUT); //Define pino sensor como entrada
22 }
23 void loop()
24 {
25     // pir
26     acionamento = digitalRead(pinPir); //Le o valor do sensor PIR
27     if (acionamento == LOW)             //Sem movimento, mantem led azul ligado
28     {
29         digitalWrite(pinLedRed, LOW);
30         digitalWrite(pinLedGreen, HIGH);
31     }
32     else //Caso seja detectado um movimento, aciona o led vermelho
33     {
34         // pir
35         digitalWrite(pinLedRed, HIGH);
36         digitalWrite(pinLedGreen, LOW);
```

```
37     // servo - sobe cancela
38     for (pos = 0; pos <= 90; pos += 1) { // goes from 0 degrees to 180 degrees
39         myservo.write(pos);              // tell servo to go to position in variable 'pos'
40         delay(15);                        // waits 15 ms for the servo to reach the position
41     }
42     delay(5000);
43     // servo - desce cancela
44     // servo - desce cancela
45     for (pos = 90; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
46         myservo.write(pos);              // tell servo to go to position in variable 'pos'
47         delay(15);                        // waits 15 ms for the servo to reach the position
48     }
49 }
50 }
```

## Software Aplicativo - Screen 1



# Software Aplicativo - Menu

quando sair .Clique  
fazer abrir outra tela nomeDaTela ▶ Screen1

quando lampada .Clique  
fazer abrir outra tela nomeDaTela ▶ Iluminacao

quando compartilhar .Clique  
fazer chamar Sharing1 .CompartilharMensagem  
mensagem " https://play.google.com/store/ "

inicializar global mostrar para 0

quando opcoes .Clique  
fazer se  
então  
ajustar lampada . Visível para verdadeiro  
ajustar compartilhar . Visível para verdadeiro  
ajustar sair . Visível para verdadeiro  
ajustar global mostrar para 1  
senão, se  
então  
ajustar lampada . Visível para falso  
ajustar compartilhar . Visível para falso  
ajustar sair . Visível para falso  
ajustar global mostrar para 0

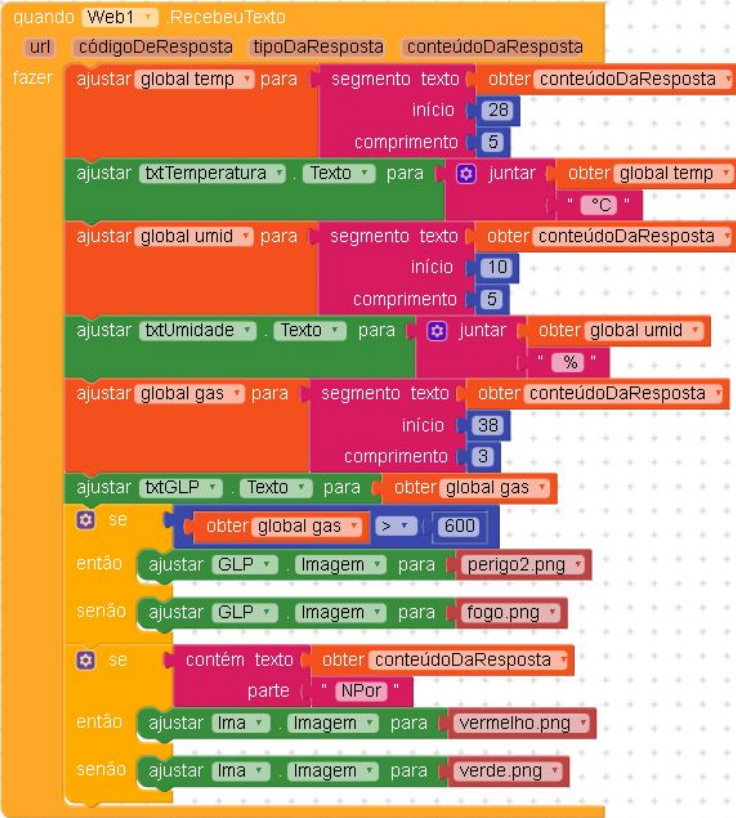
inicializar global temp para " "

inicializar global umid para " "

inicializar global gas para " "

quando Temporizador1 .Disparo  
fazer ajustar Web1 . Url para " http://192.168.165.98 "  
chamar Web1 .Obter

# Software Aplicativo - Menu



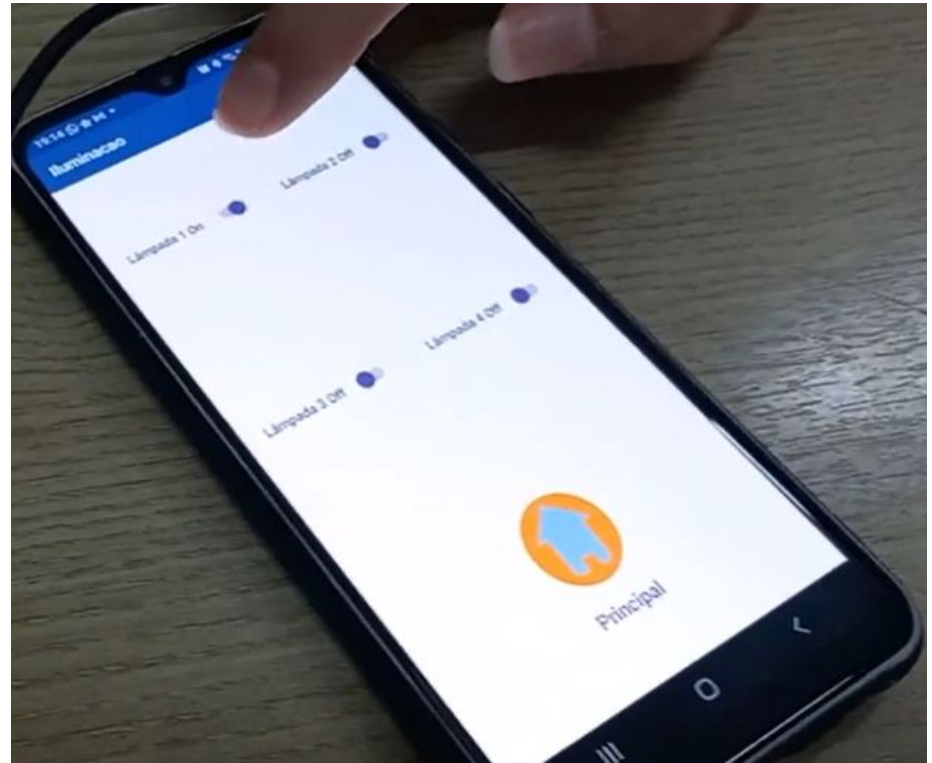
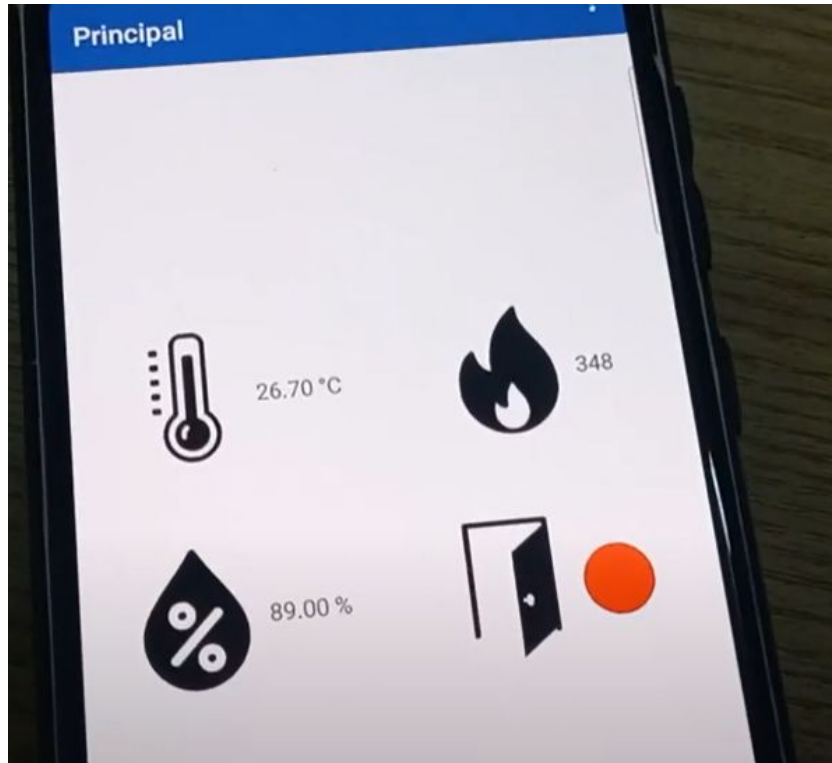
# Software Aplicativo - Iluminação

quando Button1 .Clique  
fazer abrir outra tela nomeDaTela Menu

quando Lamp1 .Clicado  
is Checked  
fazer se obter is Checked  
então ajustar Web1 . Url para " http://192.168.165.98/botl "  
chamar Web1 .Obter  
senão ajustar Web1 . Url para " http://192.168.165.98/botd "  
chamar Web1 .Obter

quando Lamp2 .Clicado  
is Checked  
fazer se obter is Checked  
então ajustar Web1 . Url para " http://192.168.165.98/bot2l "  
chamar Web1 .Obter  
senão ajustar Web1 . Url para " http://192.168.165.98/bot2d "  
chamar Web1 .Obter

## Tela do aplicativo





### MQ - 2

- VCC(5v) → VIN do ESP8266;
- GND → GND do ESP8266;
- A0 (Analógico) → A0 do ESP8266.

### Módulo Reed Switch

- D0 (Digital) → Pino D1 (05);
- Positivo → Positivo da protoboard;
- GND → Negativo da protoboard.



### DHT 11

- Out → Pino D5 (14);
- Negativo → Negativo da protoboard;
- Positivo → Positivo da protoboard.

### Módulo Reed Switch

- D0 (Digital) → Pino D1 (05);
- Positivo → Positivo da protoboard;
- GND → Negativo da protoboard.

### LED 1

- Negativo → Resistor → Negativo da protoboard;
- Positivo → Pino D6 (12).

### LED 2

- Negativo → Resistor → Negativo da protoboard;
- Positivo → Pino D7 (13).

### Buzzer

- Negativo → Negativo da protoboard;
- Positivo → Pino D2 (04).

### ESP8266

- GND → Negativo da protoboard.

Com uma fonte externa (5v), conectar nos lados positivos e negativos da protoboard.

## Resultados

- Resultados satisfatórios;
- Dados de temperatura, umidade e gás na tela do aplicativo;
- Aplicativo recebe informação do módulo Reed Switch;
- Pode-se ativar os leds através do app.
- Motor Servo funcionando com os dois sensores de presença;
- Levanta a cancela ao detectar a presença;
- Abaixa a cancela quando o sensor de presença detectar novamente;
- Leds respondem a essa comunicação.

## Conclusão

- Desejo de implementar em casa;
- Apresentamos certa dificuldade inicialmente;
- Não conseguir identificar como captar os dados recebidos pelos ESP8266;
- Procuramos vídeos e documentos para saber como fazer a comunicação entre app e microcontrolador;
- Optamos por usar a biblioteca ESP8266WiFi;
- Ter como ponto positivo ver o trabalho funcionando e o aplicativo comunicando-se com o ESP8266;
- Melhorias: receber notificação no celular sem precisar que o aplicativo esteja aberto. Tornar o IP volátil. Colocar caixa de texto para

## Links

- SMARTHOME - <https://www.youtube.com/watch?v=YAO0O0cxzMU>
- CANCELA - <https://www.youtube.com/watch?v=G2ikk1JuvDo>

# Bibliografia

- [1] ROZSA, Vitor, *et al.* O PARADIGMA TECNOLÓGICO DA INTERNET DAS COISAS E SUA RELAÇÃO COM A CIÊNCIA DA INFORMAÇÃO. BRAPCI - Base de Dados em Ciência da Informação, 2017. Disponível em: <https://brapci.inf.br/index.php/res/download/96349>. Acesso em: 13/06/2023.
- [2] GUIMARÃES, Fábio. Introdução ao ESP8266 - Aula 1 - EA. Mundo Projetado. Disponível em: <https://mundoprojetado.com.br/introducao-esp8266>. Acesso em: 13/06/2023.
- [3] DOS REIS, Fábio. Como funciona um Sensor de Movimento PIR - Passive Infrared. Bóson Treinamentos. Disponível em: <http://www.bosontreinamentos.com.br/eletronica/como-funciona-um-sensor-de-movimento-pir-passive-infrared>. Acesso em: 13/06/2023.
- [4] SILVA, Oderlando. Conhecendo a Fundo o Buzzer. Auto Core Robótica Blog. Disponível em: <https://autocorerobotica.blog.br/conhecendo-a-fundo-o-buzzer>. Acesso em: 13/06/2023.
- [5] CRAVO, Edilson. O que é um Servo Motor, como funciona e quais as vantagens?. Kalatec Automação. Disponível em: <https://blog.kalatec.com.br/o-que-e-servo-motor/>. Acesso em: 13/06/2023.
- [6] MARCELO DOS SANTOS, Diego. LED - Diodo Emissor de Luz. InfoEscola. Disponível em: <https://www.infoescola.com/eletronica/led-diodo-emissor-de-luz>. Acesso em: 13/06/2023.
- [7] ENGEASIER. COMO CRIAR SEU APLICATIVO PELO ESP32. Youtube, 2 de fevereiro de 2023. Disponível em: <https://www.youtube.com/watch?v=u2-edCqHRBk&t=151s>. Acesso em: 04 de julho de 2023.

# Fim!

