

Aula 9 – Web e IoT

PROFESSOR: HARLEY MACÊDO DE MELLO

Roteiro

- Conceitos de IoT
- Redes 5G
- Prototipagem de sistemas eletrônicos
- Ambiente de programação do dispositivo
- Enviando comandos pela rede (acionando led)
- Recebendo dados da rede (lendo sensor de temperatura e umidade)

Conceitos de IoT

- Internet of Things ou Internet das Coisas
- Coleção dos mais diversos dispositivos interagindo por redes
- Câmeras, lâmpadas, bicicletas, aspiradores, caixas de som
- Atualmente existem mais de 10 bilhões de dispositivos conectados
- Estima-se que o número de dispositivos chegará a 22 bilhões em 2025

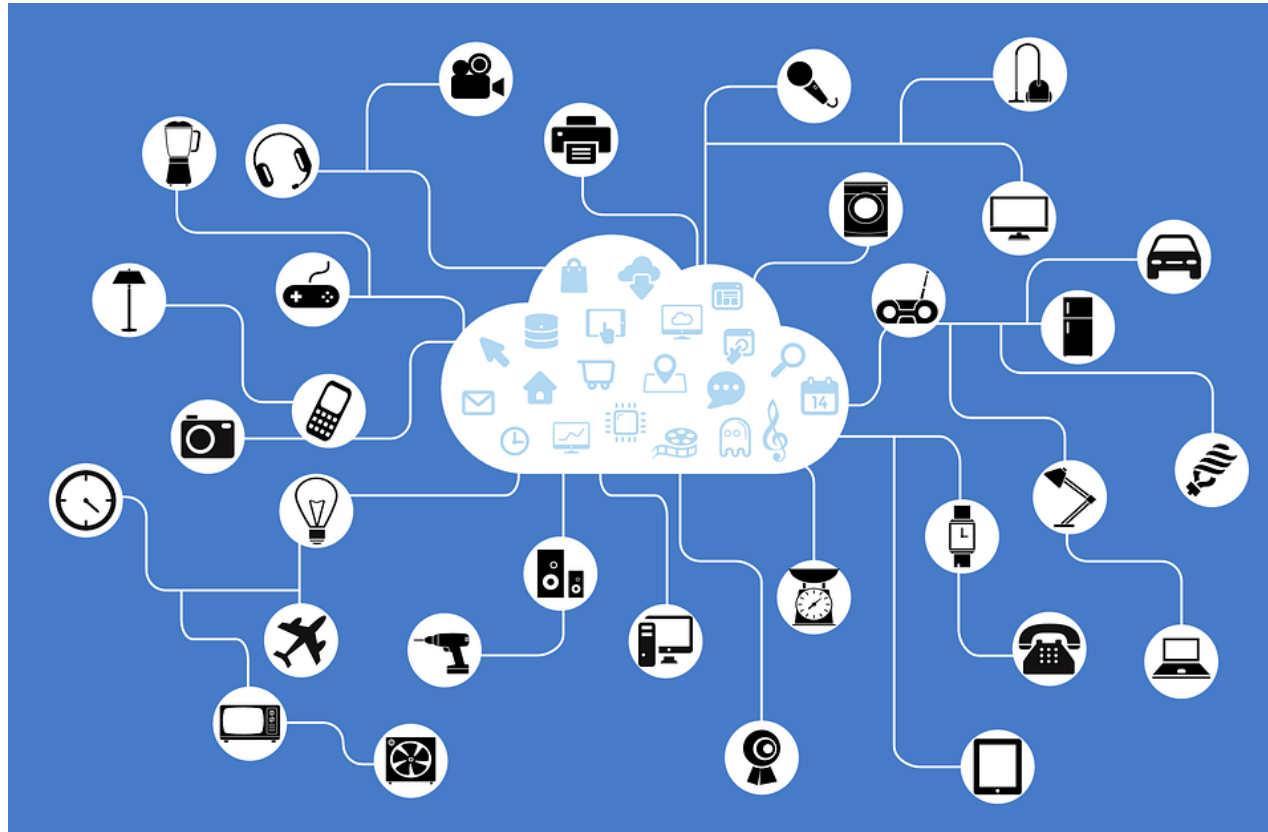
Conceitos de IoT



Segundo (Bagchi, 2020)

"A Internet das Coisas se refere a dispositivos em rede que interagem com seus arredores físicos e se comunicam por meio de redes sem fio em contextos sociais para agregar valor em aplicações centradas no ser humano" (Tradução nossa).

Conceitos de IoT

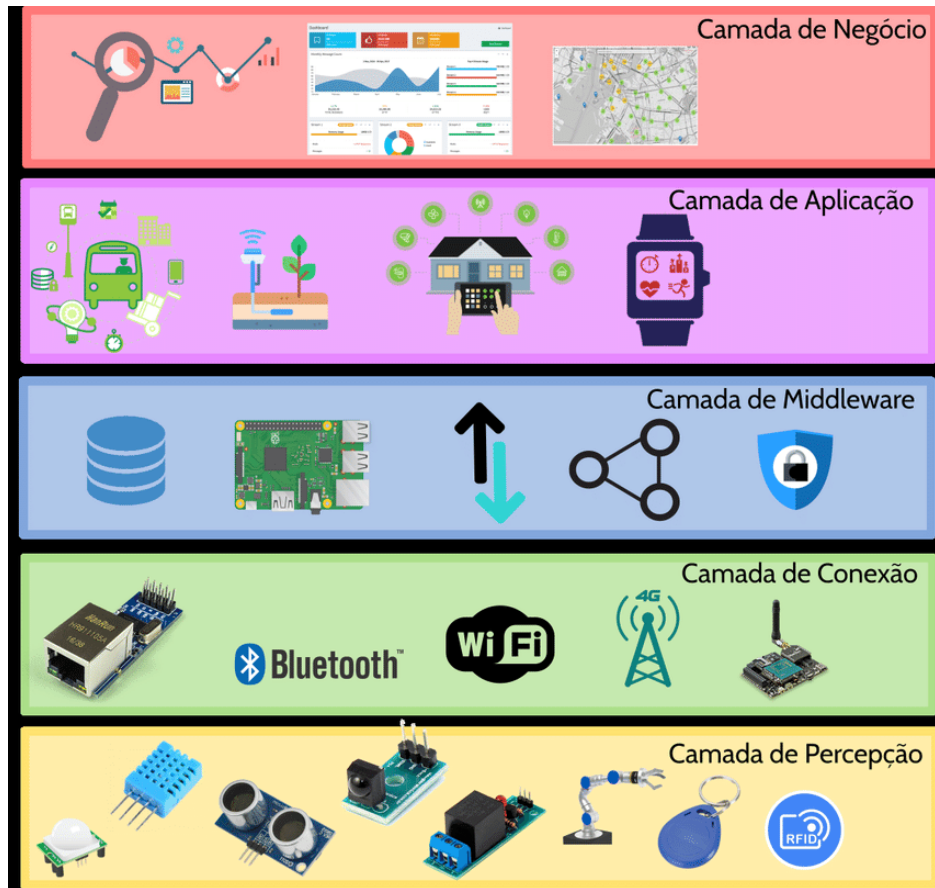


Coisas conectadas.

Conceitos de IoT

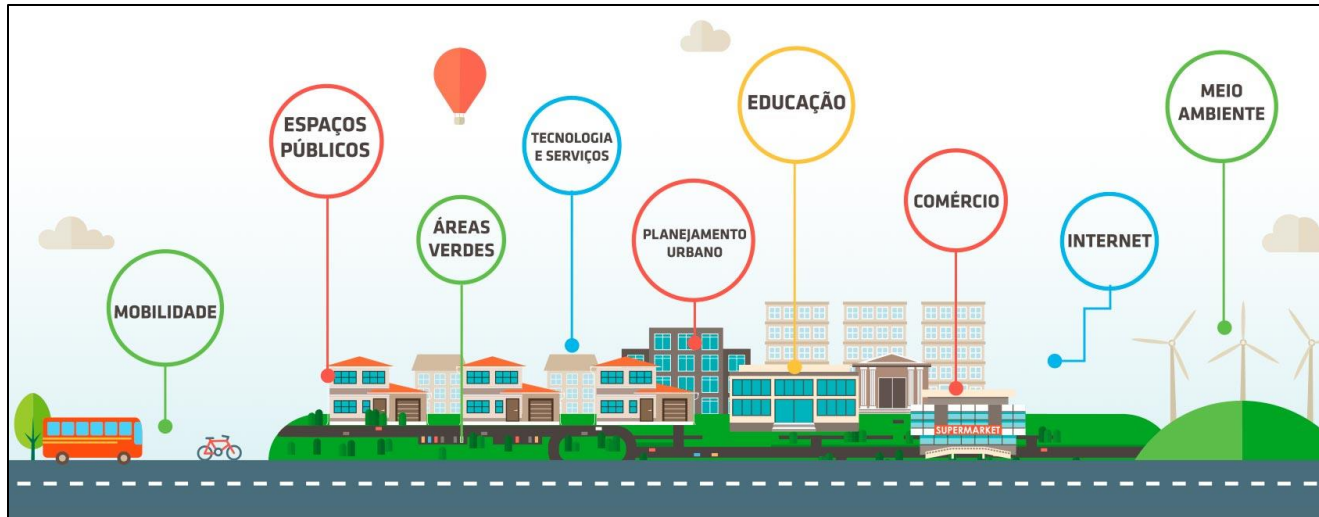
- Os componentes que geralmente formam uma rede IoT
 - Sensores: Capturam dados e informações do ambiente
 - Atuadores: Realiza tarefas para modificar o ambiente
 - Controladores: Gerencia o processamento, controla demais partes
 - Rede: Meio de comunicação dos componentes

Conceitos de IoT



Camadas que formam a IoT.

Conceitos de IoT



Os desafios passam por criar um IoT que integre diversos serviços e ajude a sociedade.

Redes 5G

- Quinta geração de internet móvel
- Maior alcance e velocidade
- Menor consumo de energia na conexão
- Auxilia o desenvolvimento da IoT

Redes 5G

- Latência de 2 milissegundos, ante 54 milissegundos da 4G
- Conexões simultâneas de 1 milhão de aparelhos por km, ante 10 mil da 4G
- Economia de 90% na energia para conexão

Redes 5G



Novos serviços, dispositivos e tendências devem surgir com as redes 5G e IoT.

Prototipagem de sistemas eletrônicos

- Prototipagem é uma etapa essencial no desenvolvimento de um sistema eletrônico
- Vários dispositivos eletrônicos de baixo custo foram desenvolvidos para prototipagem
- Os mais conhecidos são:
 - Arduino
 - Esp
 - Raspberry
- Estes dispositivos podem não ter os requisitos para o uso industrial

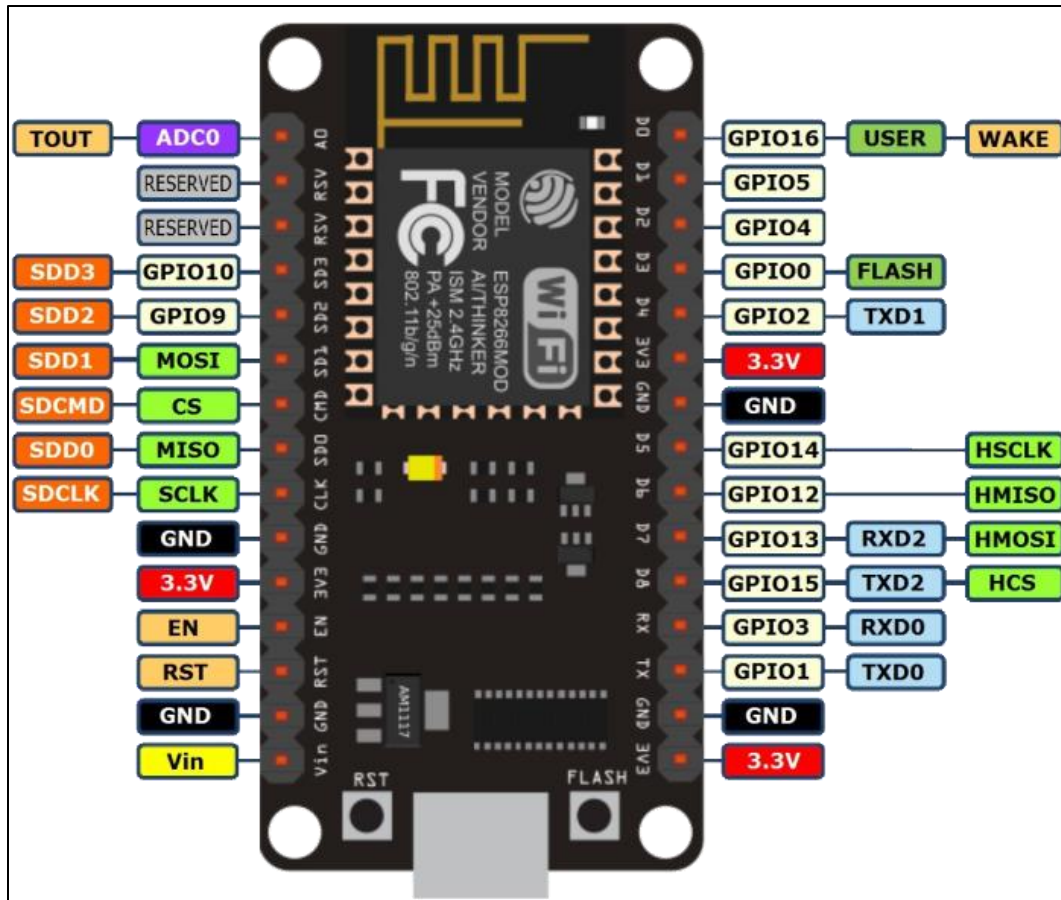
Prototipagem de sistemas eletrônicos

- ESP8266
 - Microcontrolador com conexão WiFi
 - Baixo custo
 - Aceita linguagem C++
 - Entende o protocolo TCP/IP
 - Pode realizar requisições HTTP

Prototipagem de sistemas eletrônicos

- ESP8266
 - Alimentação: 2,2V ~ 3,3V
 - Memória flash: 4MB
 - Pinos de I/O: 13
 - WiFi: IEEE 802.11 b/g/n

Prototipagem de sistemas eletrônicos

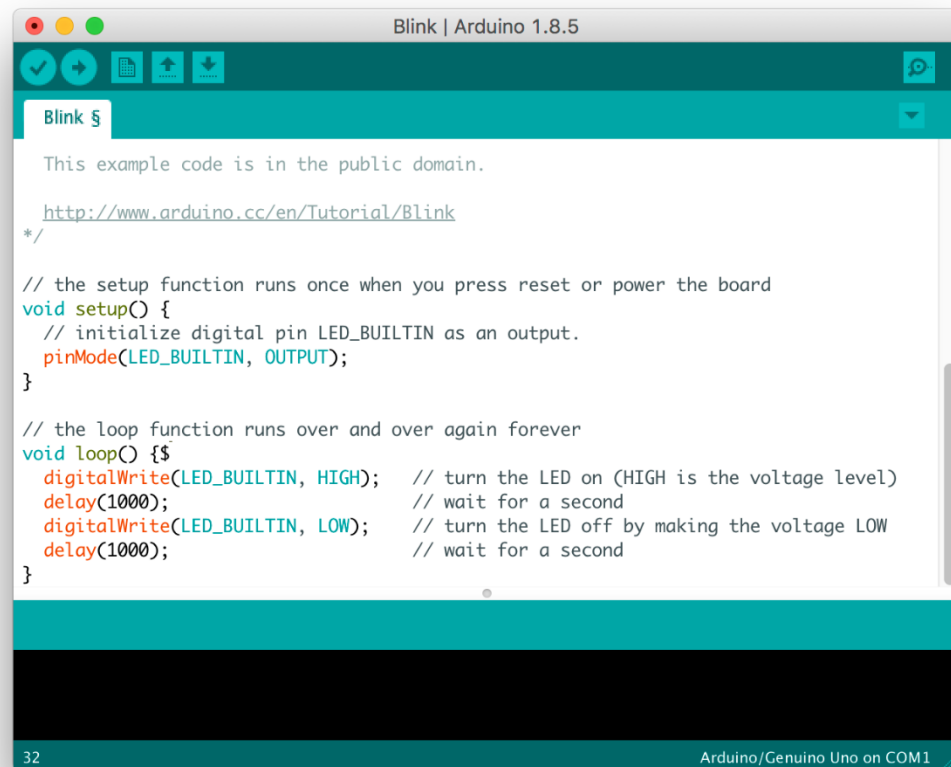


Legenda de pinos na placa ESP8266.

Ambiente de programação do dispositivo

- Arduino IDE
 - Ambiente próprio para Arduino e outros microcontroladores como o ESP8266
 - Reconhece a sintaxe da linguagem C++
 - Reconhece os dispositivos microcontroladores
 - Auxilia a adição de bibliotecas para os dispositivos
 - Monitora a saída do dispositivo

Ambiente de programação do dispositivo

A screenshot of the Arduino IDE window titled "Blink | Arduino 1.8.5". The window has a teal header bar with icons for file operations and a search icon. Below the header, there's a tab labeled "Blink \$". The main text area contains the following code:

```
This example code is in the public domain.  
  
http://www.arduino.cc/en/Tutorial/Blink  
*/  
  
// the setup function runs once when you press reset or power the board  
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000); // wait for a second  
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW  
  delay(1000); // wait for a second  
}
```

At the bottom of the window, there's a status bar showing "32" on the left and "Arduino/Genuino Uno on COM1" on the right.

Arduino IDE, cujo instalador está disponível em <https://www.arduino.cc/en/software>.

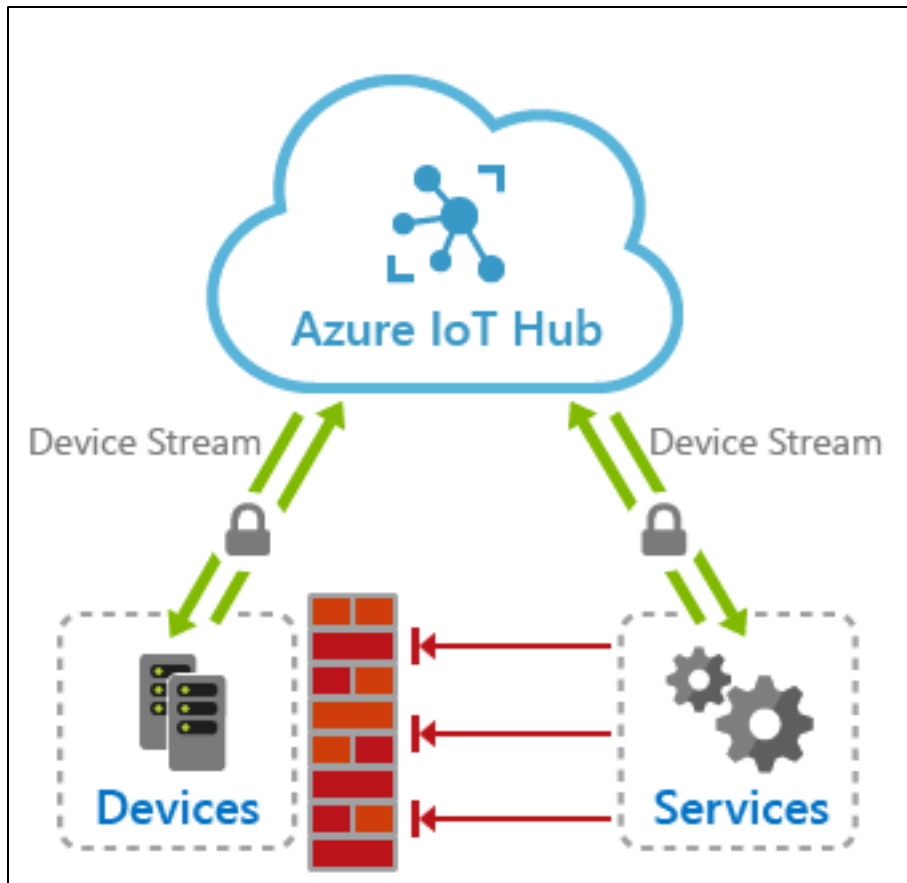
Enviando comandos pela rede

- Uma instância de um servidor web no ESP pode receber requisições HTTP
- O servidor fica ouvindo as requisições a todo momento
- O ESP pode realizar qualquer tarefa com os dados recebidos
- Pode adicionalmente fazer uma requisição em outro servidor

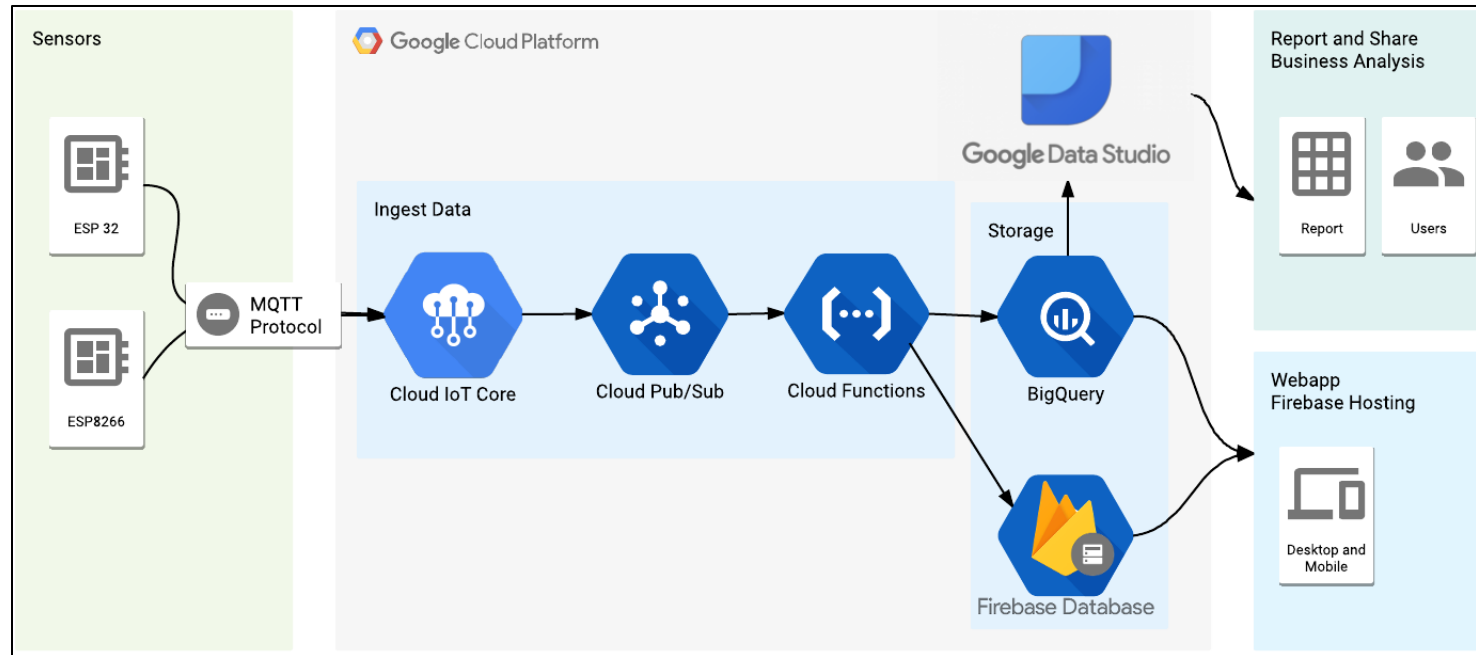
Enviando comandos pela rede

- Existem diversos serviços para auxiliar a comunicação de dispositivos na rede
- Podem usar socket para uma comunicação em tempo real
- Podem armazenar dados em banco de dado e manter comunicação assíncrona
- Alguns serviços também ajudam na criação da visualização dos dados

Enviando comandos pela rede



Enviando comandos pela rede



Arquitetura da Google Cloud Core.

Enviando comandos pela rede



Painel web para comunicação com dispositivo.

Enviando comandos pela rede

```
126 ptr += "<script>";  
127 ptr += "function ligarLed() { fetch('http://192.168.1.17/ligar_led')}";  
128 ptr += "function desligarLed() {fetch('http://192.168.1.17/desligar_led')}";  
129 ptr += "</script>";  
130 ptr += "</body>";
```

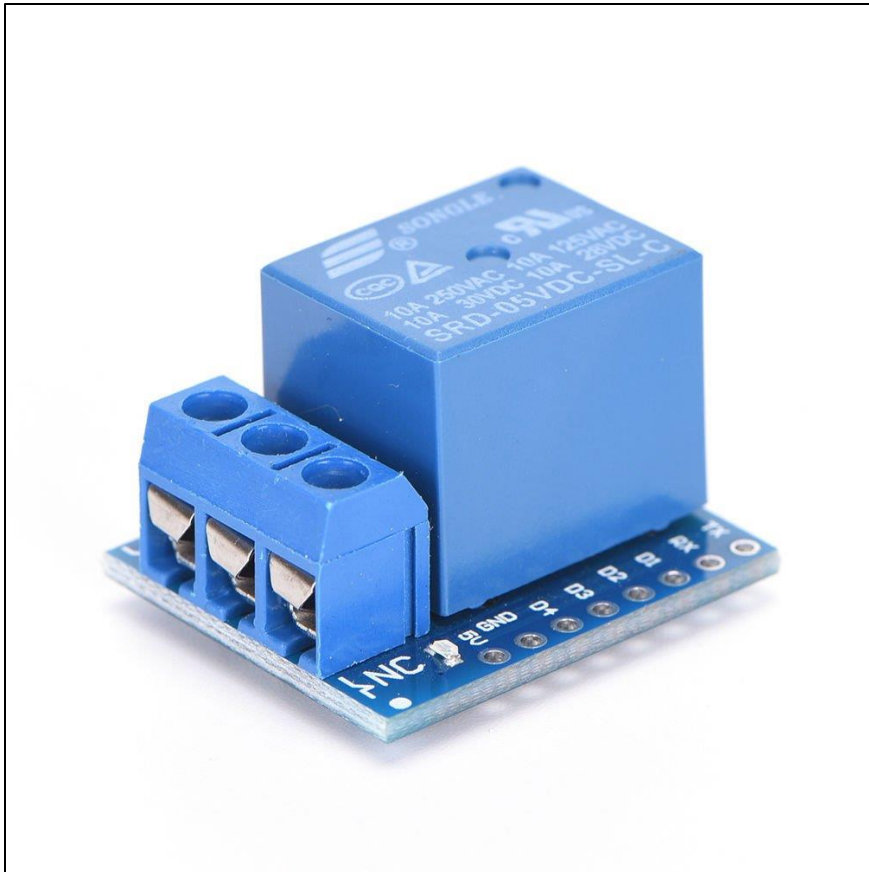
Enviando comandos pela rede

```
53 void loop() {  
54     // put your main code here, to run repeatedly:  
55     server.handleClient();  
56 }  
57  
58 void handle_OnConnect() {  
59     server.send(200, "text/html", SendHTML());  
60 }  
61  
62 void handle_ligar_led() {  
63     digitalWrite(D1, HIGH);  
64 }  
65  
66 void handle_desligar_led() {  
67     digitalWrite(D1, LOW);  
68 }
```


Módulos e sensores para ESP

- Módulo relé
 - Funciona como chave interruptor
 - Dispositivos alimentados por tomada externa
 - Liga e desliga uma lâmpada ou TV por exemplo
 - Necessário verificar a amperagem de suporte do relé

Módulos e sensores para ESP

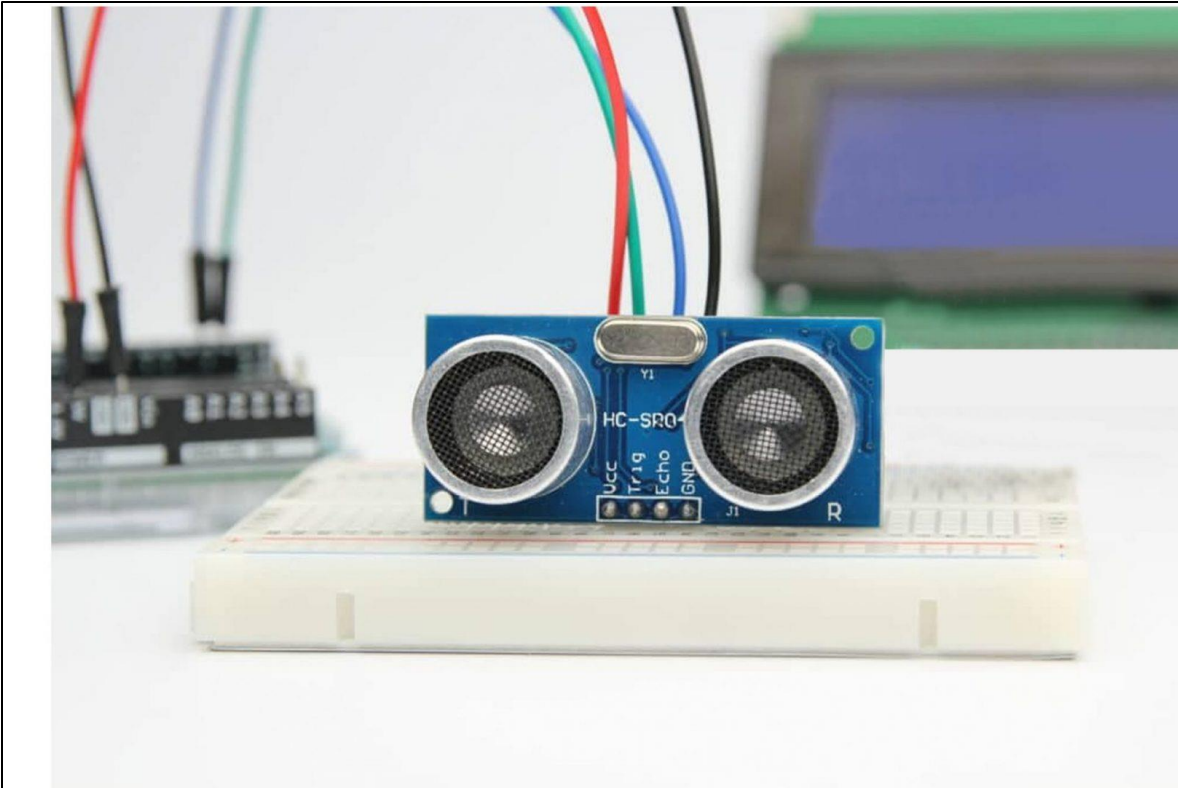


Exemplo de relé.

Módulos e sensores para ESP

- Sensor de distância
 - Usa tecnologia ultrassom para medir distância
 - Distâncias que variam de 2cm a 4m
 - Precisão de 3 mm

Módulos e sensores para ESP



Sensor de distância encaixado em uma protoboard.

Módulos e sensores para ESP

- Sensor de presença/movimento
 - Usa tecnologia de infravermelho
 - Baixo consumo de energia
 - Ótima precisão
 - Possui ajuste de sensibilidade e tempo

Módulos e sensores para ESP

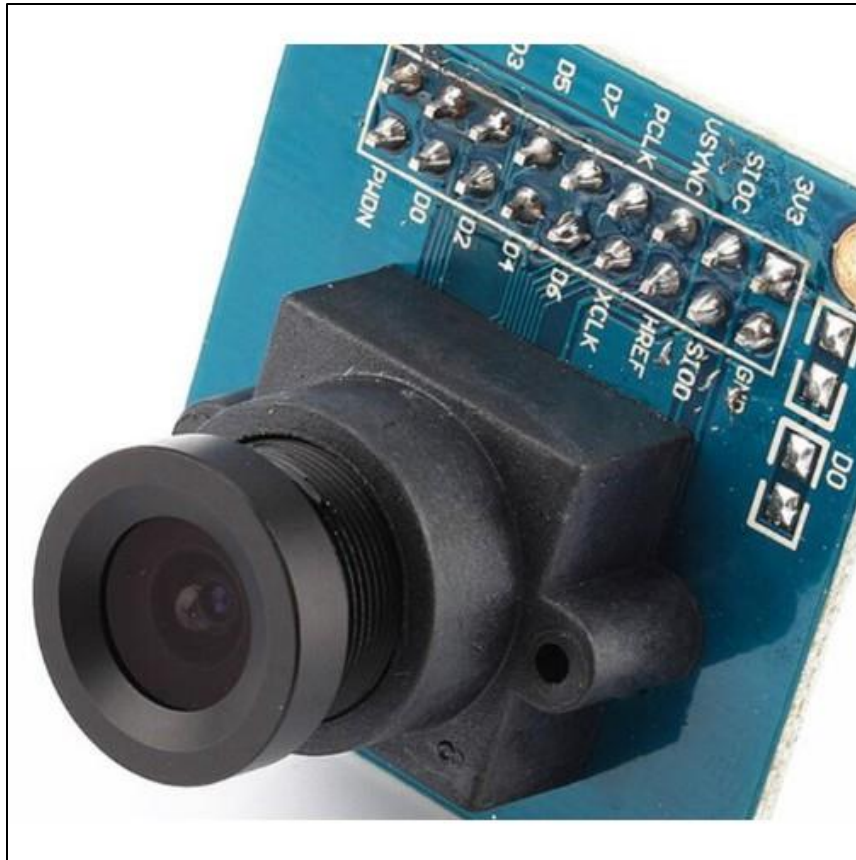


Sensor de presença HC-SR501.

Módulos e sensores para ESP

- Sensor de câmera
 - Captura imagens coloridas
 - Resolução máxima de 640 x 480 pixels
 - Controle automático de qualidade de imagem

Módulos e sensores para ESP

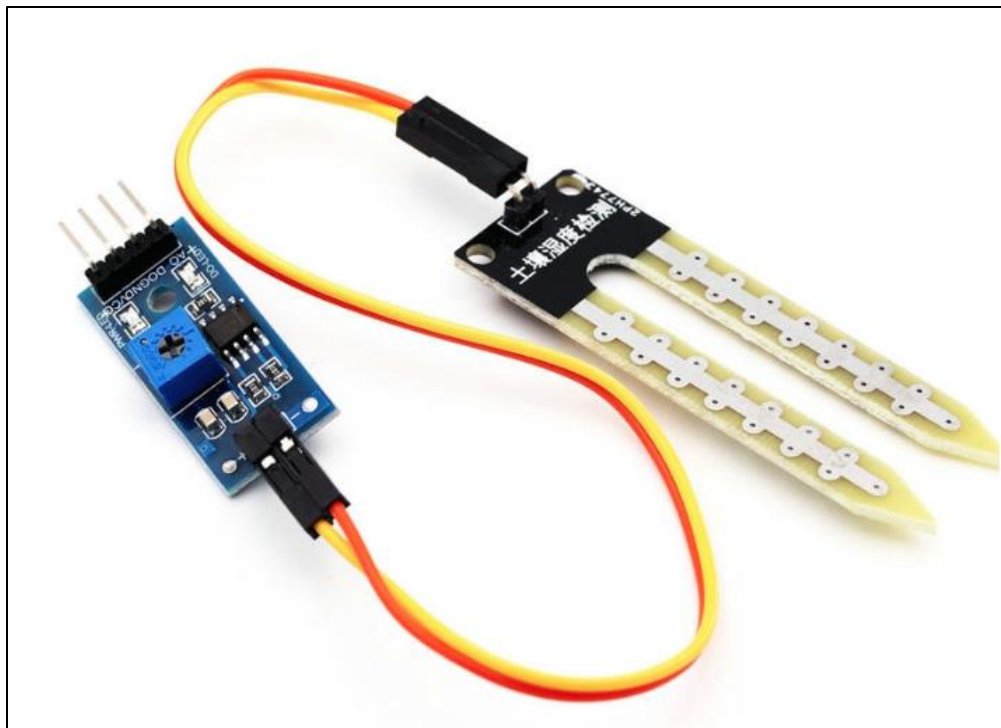


Sensor de câmera OV7670.

Módulos e sensores para ESP

- Sensor de umidade do solo
 - Mede a quantidade de água existente no solo
 - É necessário que uma parte do sensor esteja aterrada ao solo
 - Muito usado no setor agrário

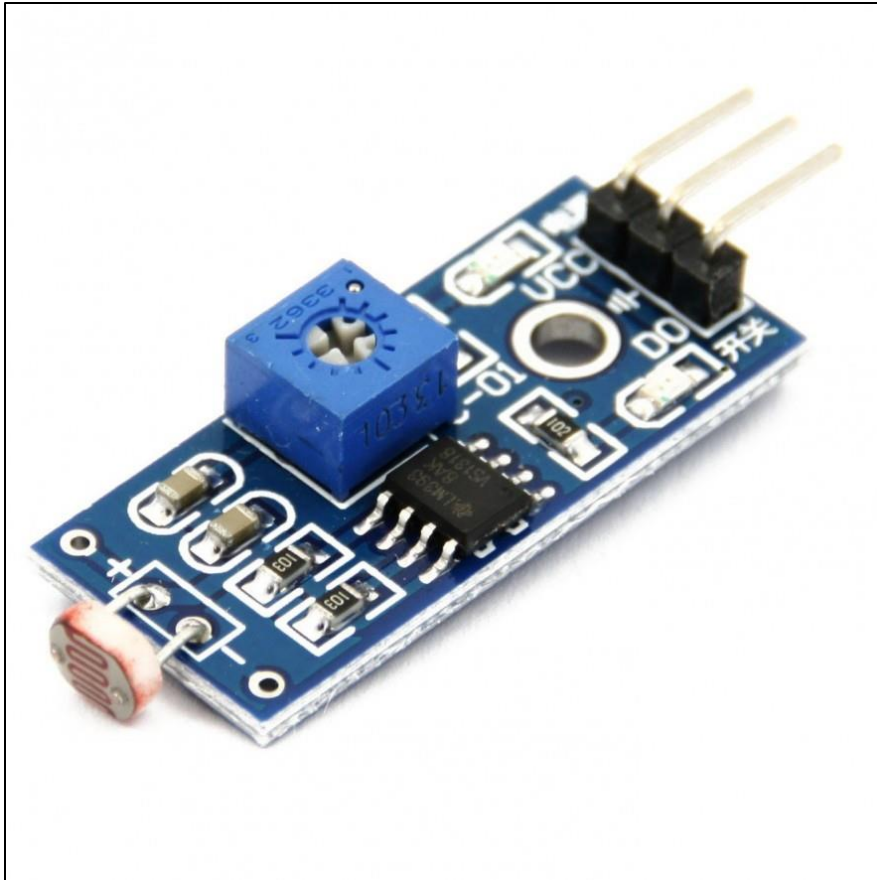
Módulos e sensores para ESP



Módulos e sensores para ESP

- Sensor de luminosidade
 - Mede a luminosidade a qual está exposto
 - Possui ajuste de sensibilidade
 - Muito usado para acionamento automático de luzes

Módulos e sensores para ESP



Sensor de luminosidade LDR
5mm.

Módulos e sensores para ESP

- Sensor de som
 - Mede a intensidade do som ao seu redor
 - Possui um microfone
 - Possui ajuste de sensibilidade

Módulos e sensores para ESP

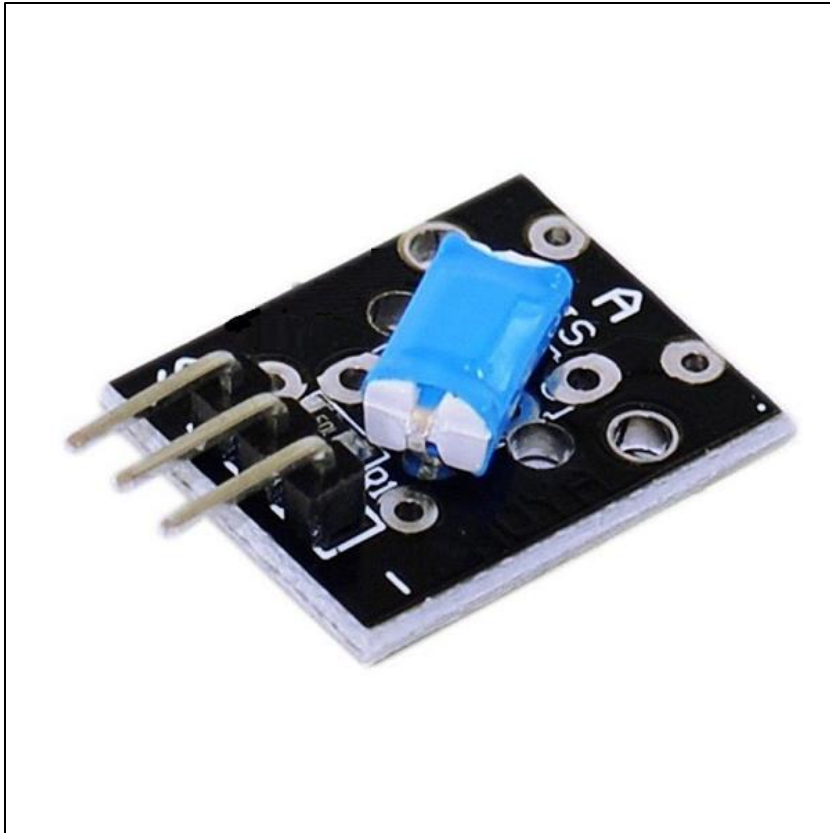


Sensor de som KY-038

Módulos e sensores para ESP

- Sensor de inclinação
 - Detecta quando algum objeto inclina
 - Deve estar acoplado ao objeto
 - Possui boa precisão
 - Normalmente é usado em projetos de segurança

Módulos e sensores para ESP

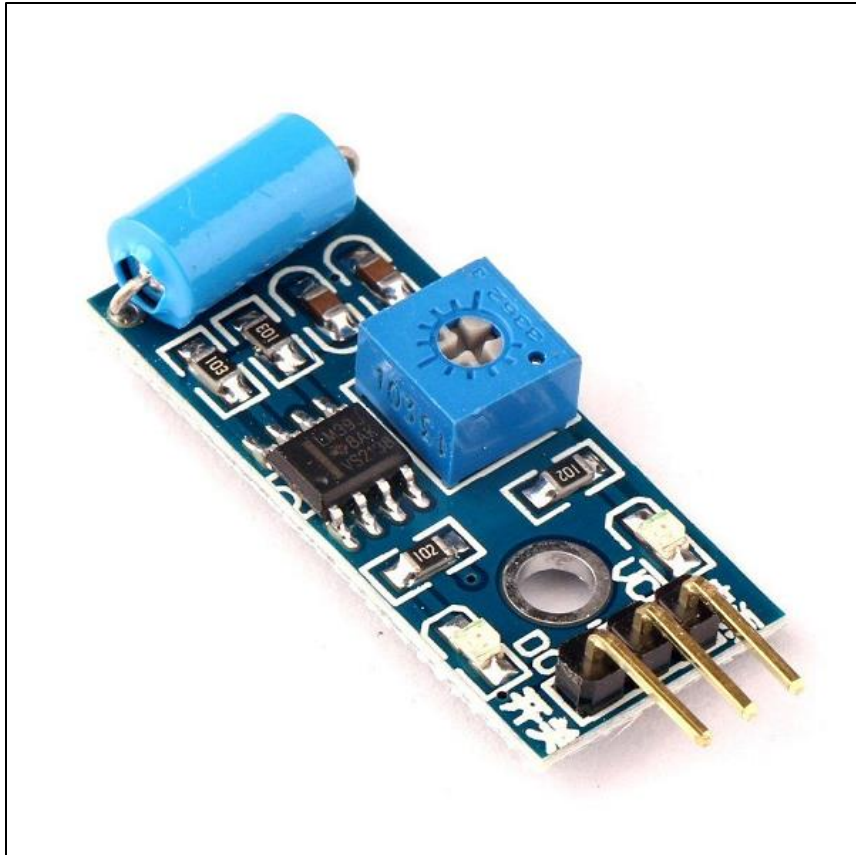


Sensor de inclinação KY-020.

Módulos e sensores para ESP

- Sensor de vibração
 - Detectar vibrações em ambiente ou superfície
 - Muito usado para medir vibração de máquinas
 - Possui ajuste de sensibilidade

Módulos e sensores para ESP

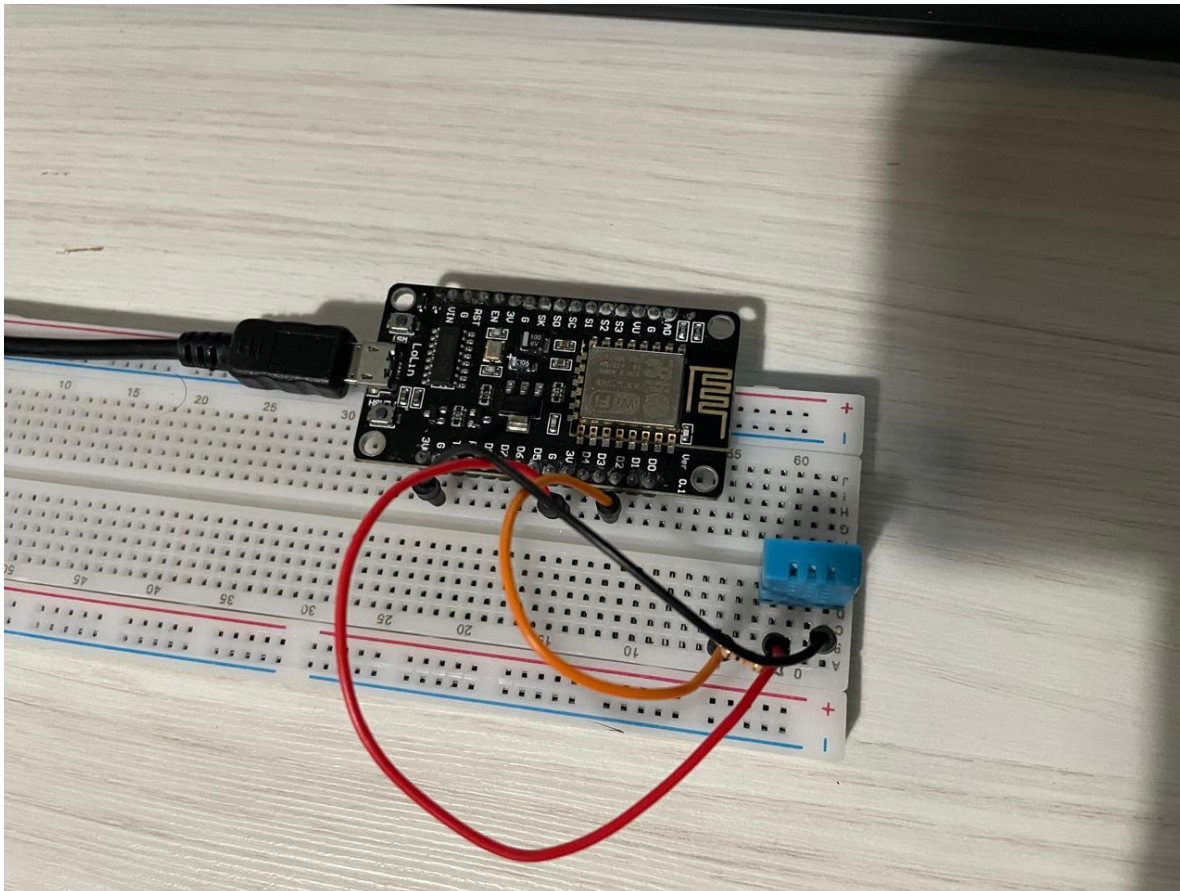


Sensor de vibração SW-420.

Exemplo de monitoramento do clima

- Usando ESP8266 e DHT11
- Registrar em nuvem a temperatura e umidade a cada 10 segundos
- Manter o registro das últimas 5 informações de clima
- Enviar dados via GET com query string
- Exibir os dados em frontend web com HTML ou React

Exemplo de monitoramento do clima



Exemplo de monitoramento do clima

```
JS index.js > ...
10  server.use(express.static('public'))
11
12  var climas = []
13
14  server.use(express.json())
15
16  server.get('/listar', (req, res) => {
17    res.json({clima: climas})
18  })
19
20  server.get('/clima', (req, res) => {
21    climas.push({"temperatura": req.query.temp, "umidade": req.query.umid})
22    if (climas.length === 6) {
23      climas.shift()
24    }
25    res.json({mensagem: 'Clima registrado'})
26  })
27
28  server.get('/painelClima', (req, res) => {
29    res.sendFile(__dirname + '/public/index.html')
30  })
31
32  server.listen(3000)
```

Exemplo de monitoramento do clima

```
9  <body>
10  <h1>Observatório do clima</h1>
11  <button onclick="listar()">Listar climas</button>
12  <div id="resp"></div>
13  <script>
14      async function listar() {
15          const url = "/listar"
16          const resposta = await fetch(url)
17          const respFormatada = await resposta.json()
18          respFormatada.forEach(function (clima) {
19              document.getElementById("resp").innerHTML += '<button>' +
20                  clima.temperatura + '</button> + <button>' + clima.umidade + '</button>'
21          })
22      }
23  </script>
24 </body>
```

Exemplo de monitoramento do clima

Ex1.ino

```
1
2  #include <SPI.h> //INCLUSÃO DE BIBLIOTECA
3  #include <dht.h> //Biblioteca do sensor de temperatura
4
5  #include <ESP8266WiFi.h>
6  #include <ESP8266HTTPClient.h>
7  #include <WiFiClient.h>
8  #include <Arduino_JSON.h>
9
10 const char* ssid = "brisa-948110";
11 const char* password = "123456";
12
13 const int pinoDHT11 = D2; //PINO ANALÓGICO UTILIZADO PELO DHT11
14 dht DHT; //VARIÁVEL DO TIPO DHT
```

Exemplo de monitoramento do clima

```
16 void setup() {
17     Serial.begin(9600); //INICIALIZA A SERIAL
18     Serial.print("Vai começar a impressao \n");
19
20     //Modulo wifi
21     WiFi.mode(WIFI_STA);
22     WiFi.begin(ssid, password);
23     Serial.println("");
24     while (WiFi.status() != WL_CONNECTED) {
25         delay(500);
26         Serial.print(".");
27     }
28     Serial.println("");
29     Serial.print("Rede WiFi: ");
30     Serial.println(ssid);
31     Serial.print("Endereço IP: ");
32     Serial.println(WiFi.localIP());
33     delay(100);
34 }
```


Exemplo de monitoramento do clima

```
36 void loop() {
37     DHT.read11(pinoDHT11); //LÊ AS INFORMAÇÕES DO SENSOR
38     Serial.print("Umidade: "); //IMPRIME O TEXTO NA SERIAL
39     Serial.print(DHT.humidity); //IMPRIME NA SERIAL O VALOR DE UMIDADE MEDIDO
40     Serial.print("%"); //ESCREVE O TEXTO EM SEGUIDA
41     Serial.print(" / Temperatura: "); //IMPRIME O TEXTO NA SERIAL
42     Serial.print(DHT.temperature, 0); //IMPRIME NA SERIAL O VALOR DE UMIDADE MEDIDO E REMOVE A PARTE DECIMAL
43     Serial.println("*C"); //IMPRIME O TEXTO NA SERIAL
44     int temperaturaFinal = DHT.temperature;
45     int umidadeFinal = DHT.humidity;
```

Exemplo de monitoramento do clima

```
47  if (WiFi.status() == WL_CONNECTED) {
48      WiFiClient client;
49      HTTPClient http;
50      String serverName = "http://192.168.1.7:3000/clima?temp=";
51      serverName.concat(temperaturaFinal);
52      serverName.concat("&umid=");
53      serverName.concat(umidadeFinal);
54      Serial.println(serverName);
55      http.begin(client, serverName);
56      int httpCode = http.GET();
57      Serial.println(httpCode);
58      Serial.printf(http.errorToString(httpCode).c_str());
59      String payload = "{}";
60      if (httpCode>0) {
61          payload = http.getString();
62      }
63      Serial.println(payload);
64      http.end();
65  }
66
67  delay(10000);
68 }
```

Exemplo de monitoramento do clima

Message (Enter to send message to 'NodeMCU 0.9 (ESP-12 Module)' on 'COM4')

```
http://192.168.1.7:3000/clima?temp=26&umid=36  
200
```

```
{"mensagem":"Clima registrado"}
```

```
Umidade: 36.00% / Temperatura: 26°C
```

```
http://192.168.1.7:3000/clima?temp=26&umid=36  
200
```

```
{"mensagem":"Clima registrado"}
```

```
Umidade: 35.00% / Temperatura: 26°C
```

```
http://192.168.1.7:3000/clima?temp=26&umid=35  
200
```

```
{"mensagem":"Clima registrado"}
```

```
Umidade: 36.00% / Temperatura: 26°C
```

```
http://192.168.1.7:3000/clima?temp=26&umid=36
```

Exemplo de monitoramento do clima



```
{
  "clima": [
    {
      "temperatura": "26",
      "umidade": "36"
    },
    {
      "temperatura": "26",
      "umidade": "36"
    },
    {
      "temperatura": "26",
      "umidade": "36"
    },
    {
      "temperatura": "26",
      "umidade": "36"
    },
    {
      "temperatura": "26",
      "umidade": "35"
    }
  ]
}
```