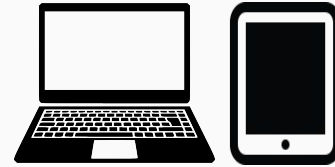
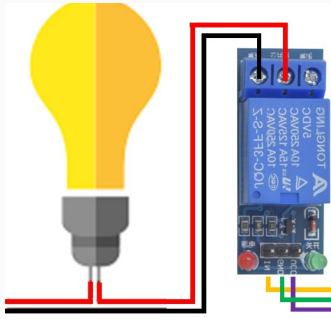


Professor: Luiz Felipe Oliveira

ESP32 AP Webserver

WIFI MODE AP



WIFI
MODE AP



1. beacon WiFi para divulgação da rede ESPNet

2. Requisições de autenticação e associação à rede sem fio ESPNet

3. Resposta da autenticação e associação



http://

1. Requisição HTTP GET /

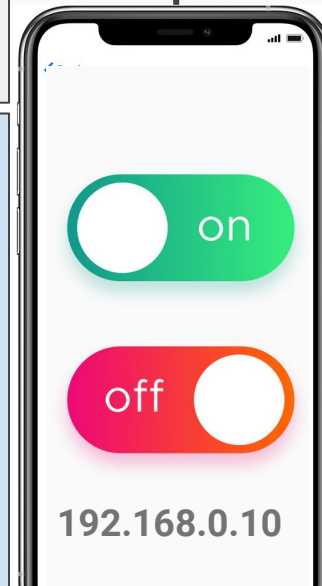
2. Resposta com código HTML/CSS/JS

3. Requisição HTTP GET /on

4. Resposta com código HTML/CSS/JS

5. Requisição HTTP GET /off

6. Resposta com código HTML/CSS/JS



```
1.  #include <WiFi.h>
2.  #include <WiFiClient.h>
3.  #include <WiFiAP.h>
4.  #include <WebServer.h>
5.  #include <ESPmDNS.h>
6.
7.  #define LED_BUILTIN 2    //Configura o pino do LED embutido
8.
9.  //Configura o nome da rede sem fio que será criada
10. const char *ssid = "ESPLuiz";
11.
12. //Configura a senha de proteção da rede sem fio
13. // ATENÇÃO a qtd mínima de 8 caracteres da senha
14. const char *password = "yourPassword";
15.
16. WebServer server(80);
```

```
18. void setup(void) {
19. pinMode(LED_BUILTIN, OUTPUT);
20.
21.
22. //Configura o baud rate da comunicação serial
23. Serial.begin(115200);
24.
25. //Você pode remover o parâmetro de senha, caso deseje criar uma rede sem senha
26. WiFi.softAP(ssid, password);
27. IPAddress myIP = WiFi.softAPIP();
28. Serial.print("SSID: ");
29. Serial.println(ssid);
30. Serial.print("AP IP address: ");
31. Serial.println(myIP);
32. server.begin();
33.
34. Serial.println("Servidor Iniciado");
```

```
36.  if (MDNS.begin("esp32")) {
37.      Serial.println("MDNS responder started");
38.  }
39.
40.  server.on("/" , handleRoot);
41.
42.  server.on("/on", handleOn);
43.
44.  server.on("/off", handleOff);
45.
46.  /*server.on("/inline", []() {
47.      server.send(200, "text/plain", "this works as well");
48.      });*/
49.
50.  server.onNotFound(handleNotFound);
51.
52.  server.begin();
53.  Serial.println("HTTP server started");
54.  }
```

```
56. void handleRoot() {
57.   String html = "";
58.   html+= "<style>a {font-size:40px; background-color:#CCCCCC;}</style>";
59.   html+= "<meta charset='UTF-8'>";
60.   html += "<h1>Exemplo SoftAP IFRJ campus Niterói</h1>";
61.   html += "Clique <a href=\"/on\">aqui</a> para ligar o LED.<br><br><br>";
62.   html += "Clique <a href=\"/off\">aqui</a></h2> para desligar o LED.";
63.   html += "<h3>Autor: Luiz Oliveira</h3>";
64.   server.send(200, "text/html", html);
65. }
66.
67. void handleOn() {
68.   digitalWrite(LED_BUILTIN, 1);
69.   handleRoot();
70. }
71.
72. void handleOff() {
73.   digitalWrite(LED_BUILTIN, 0);
74.   handleRoot();
75. }
76.
```

```
77. void handleNotFound () {
78.     digitalWrite (LED_BUILTIN, 1);
79.     String message = "File Not Found \n\n";
80.     message += "URI: ";
81.     message += server.uri ();
82.     message += "\nMethod: ";
83.     message += (server.method () == HTTP_GET) ? "GET" : "POST";
84.     message += "\nArguments: ";
85.     message += server.args ();
86.     message += "\n";
87.     for (uint8_t i = 0; i < server.args (); i++) {
88.         message += " " + server.argName (i) + ": " + server.arg (i) + "\n";
89.     }
90.     server.send (404, "text/plain", message);
91.     digitalWrite (LED_BUILTIN, 0);
92. }
93.
94. void loop (void) {
95.     server.handleClient ();
96.     delay (2); //allow the cpu to switch to other tasks
97. }
98.
```


WIFI
MODE AP

http://

1. Requisição HTTP GET /
2. Resposta com código HTML/CSS/JS
3. Requisição HTTP GET /on
4. Resposta com código HTML/CSS/JS
5. Requisição HTTP GET /off
6. Resposta com código HTML/CSS/JS



192.168.4.1

Exemplo SoftAP IFRJ campus Niterói

Clique [aqui](#) para ligar o LED.

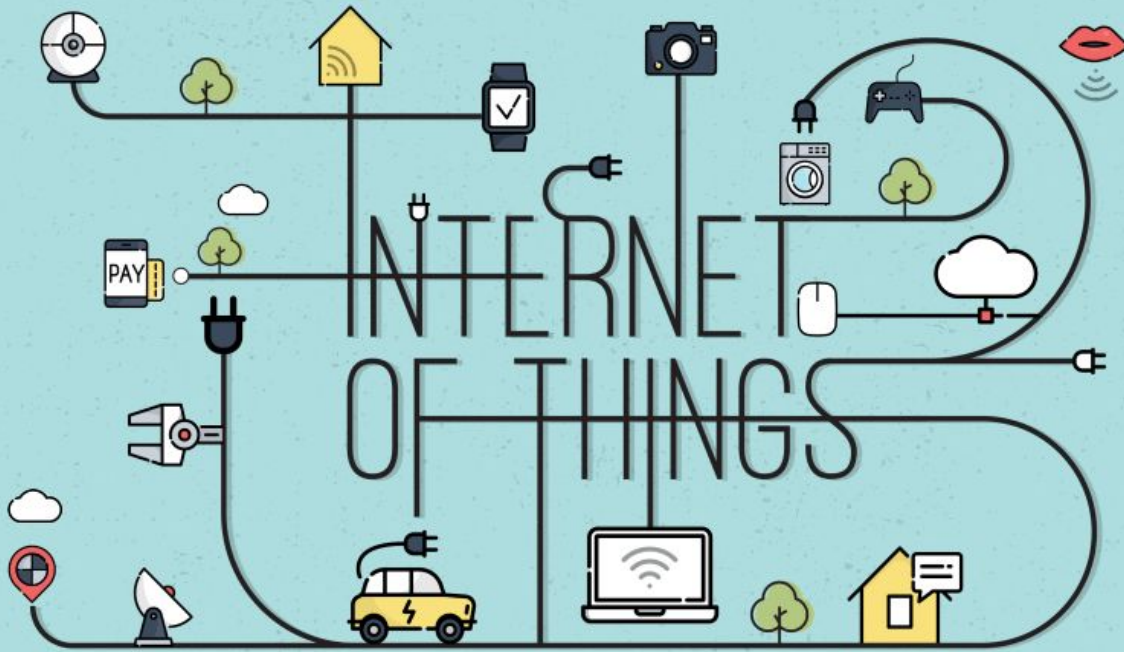
Clique [aqui](#) para desligar o LED.

Autor: Luiz Oliveira

```
void handleOn () {  
  digitalWrite (LED_BUILTIN,  
  1);  
  handleRoot ();  
}
```

```
void handleOff () {  
  digitalWrite (LED_BUILTIN,  
  0);  
  handleRoot ();  
}
```





Professor: Luiz Felipe Oliveira

Dúvidas