

```
// Inclusão Biblioteca Wifi
#include <WiFi.h>

// Configuração Rede Local
const char* ssid      = "I.O.T_Prof.Michel";
const char* password = "IOT20222";

// Servidor Wifi na Porta 80
WiFiServer server(80);

void setup()
{
    // Configura Comunicação Serial
    Serial.begin(115200); // velocidade da porta
    pinMode(2, OUTPUT);   // Seta pino 4 (GPIO2) como saída
    delay(10);

    // Conectando a uma rede WiFi
    Serial.println();
    Serial.println();
    Serial.print("Conectando para a rede ");
    Serial.println(ssid);
}
```

```
// Inicializando e validando a rede
WiFi.begin(ssid, password);

// aguarda a conexão WiFi
while (WiFi.status() != WL_CONNECTED)
{
    delay(500);
    Serial.print(".");
}
// mensagem de Conectado à Rede
Serial.println("");
Serial.println("WiFi connected.");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
Serial.print("ESP Mac Address: ");
Serial.println(WiFi.macAddress());
Serial.print("Subnet Mask: ");
Serial.println(WiFi.subnetMask());
Serial.print("Gateway IP: ");
Serial.println(WiFi.gatewayIP());
Serial.print("DNS: ");
```

```
Serial.println(WiFi.dnsIP());
server.begin();
}

void loop()
{
    // Aguarda a conexão de clientes
    WiFiClient client = server.available();
    if (client) // Se algum cliente se conectar
    {
        Serial.println(client);
        Serial.println("Novo Cliente."); // imprime a mensagem
        String currentLine = ""; // armazena os dados recebidos pelo cliente
        while (client.connected()) // enquanto o cliente estiver conectado
        {
            if (client.available()) // se tiver dados para ler do cliente
            {
                char c = client.read(); // lê o byte e imprime no terminal serial
                Serial.println(c);
                Serial.write(c);
                if (c == '\n') // se o byte for um caractere de nova linha
                {
```

```
// se a linha atual estiver em branco,  
// você tem dois caracteres de nova linha seguidos.  
// esse é o fim da solicitação HTTP do cliente,  
// então envie uma resposta:  
  
if (currentLine.length() == 0)  
{  
    // Os cabeçalhos HTTP sempre começam com um  
    // código de resposta (por exemplo, HTTP/1.1 200 OK)  
    // e um tipo de conteúdo para que o cliente saiba  
    // o que está por vir, então uma linha em branco:  
    client.println("HTTP/1.1 200 OK");  
    client.println("Content-type:text/html");  
    client.println();  
  
    //o conteúdo da resposta HTTP segue o cabeçalho:  
    client.print("Click <a href=\"/H\">here</a> to turn the LED on pin 5 on.<br>");  
    client.print("Click <a href=\"/L\">here</a> to turn the LED on pin 5 off.<br>");  
  
    // A resposta HTTP termina com outra linha em branco:  
    client.println();
```

```
        //sair do loop while
        break;
    }
    else
    {
        currentLine = ""; //se você tiver uma nova linha, limpe currentLine:
    }
} else if (c != '\r')
{ // se você tiver qualquer outra coisa além de
  // um caractere de retorno de carro
  currentLine += c; // adicione-o ao final da currentLine
}
```

```
if (currentLine.endsWith("GET /H"))
{
    digitalWrite(2, HIGH);
}
if (currentLine.endsWith("GET /L"))
{
    digitalWrite(2, LOW);
}
```

```
    }  
}  
// encerra a conexão:  
client.stop();  
Serial.println("Client Disconnected.");  
}  
}
```

*Prof. Michel Chaparro*