Software de programação do Teste Módulo WIFI Nodemcu ESP8266 V3

```
/*************** Eletrodex Eletrônica ***************
            Teste Módulo WIFI Nodemcu ESP8266 V3
 ************************************************/


#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <ESP8266mDNS.h>

const char* ssid = "Eletrodex_Testes"; // Define o nome da rede Wi_fi.
const char* password = "1157eletro_teste"; // Define a senha da rede Wi_fi.

ESP8266WebServer server(80); // Objeto server, que atende pela porta padrão do navegador
80.

const int led = 2;

void handleRoot() {     // Função de requisição do navegador para o ESP8266.
 digitalWrite(led, 1);
 #define pin_lm35 A0
 String textoHTML;
 float valor_analog_lm35 = float(analogRead(pin_lm35)); // Obtém o valor analógico que varia
de 0 a 1023 (0 = 0V, e 5V =1023)
 float tensao = ((valor_analog_lm35 * 5) / 1023)/2; // Vamos converter esse valor para tensão
elétrica
 float temperatura = tensao / 0.010; // dividimos a tensão por 0.010 que representa os 10 mV
pois cada °C = 10mV, Então 10mV/°C

 textoHTML = "<h2><strong>Eletrodex Eletr&ocirc;nica - </strong>Teste de
atua&ccedil;&atilde;o M&oacute;dulo WiFi NodeMcu ESP8266 V3:</h2>";
 textoHTML += "<p>Temperatura ambiente atual:</p>";
 textoHTML += temperatura;
 textoHTML += "&deg;C";
 textoHTML += "<h3><strong>Venha para a eletrodex.  </strong>Acesse
eletrodex.net</h3>";

 server.send(200, "text/html", textoHTML);  //Função que envia a página WEB
 digitalWrite(led, 0);
}

void handleNotFound(){   // Página Defaut, caso a página requisitada não exista
 digitalWrite(led, 1);
 String message = "File Not Found\n\n";
 message += "URI: ";
```

```
  message += server.uri();
  message += "\nMethod: ";
  message += (server.method() == HTTP_GET)?"GET":"POST";
  message += "\nArguments: ";
  message += server.args();
  message += "\n";
  for (uint8_t i=0; i<server.args(); i++){
    message += " " + server.argName(i) + ": " + server.arg(i) + "\n";
  }
  server.send(404, "text/plain", message);
  digitalWrite(led, 0);
}

void setup(void){
  pinMode(led, OUTPUT);
  digitalWrite(led, 0);
  Serial.begin(115200);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  Serial.println("");

 // testando a conexão
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("Connected to ");
  Serial.println(ssid);
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());

  if (MDNS.begin("esp8266")) {
    Serial.println("MDNS responder started");
  }

  server.on("/", handleRoot);

  server.on("/inline", [](){
    server.send(200, "text/plain", "this works as well");
  });

  server.onNotFound(handleNotFound);

  server.begin();
  Serial.println("HTTP server started");
}
```

```
void loop(void){
  server.handleClient();
}
```