```
#include <WiFi.h>
#include <WiFiClient.h>
#include <WebServer.h>
#include <ESPmDNS.h>
#include <DHT.h>

//variáveis de conexão
const char *ssid = "RENAN MARCELO FIBRA";
const char *password = "11/09/2003";

#define LED 4

const int gas_digital = 35;
const int buzzer = 2;

WebServer server(80);
DHT dht(26, DHT11);



void handleRoot() {
 char msg[1500];

 snprintf(msg, 1500,
        "<html>\
 <head>\
  <meta http-equiv='refresh' content='4'/>\
  <meta name='viewport' content='width=device-width, initial-scale=1'>\
  <link rel='stylesheet' href='https://use.fontawesome.com/releases/v5.7.2/css/all.css' integrity='sha384-
fnmOCqbTlWIlj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr' crossorigin='anonymous'>\
  <title>Monitoramento de DataCenter</title>\
  <style>\
  html { font-family: Arial; display: inline-block; margin: 0px auto; text-align: center;}\
  h2 { font-size: 3.0rem; }\
  p { font-size: 3.0rem; }\
  .units { font-size: 1.2rem; }\
  .dht-labels{ font-size: 1.5rem; vertical-align:middle; padding-bottom: 15px;}\
  </style>\
 </head>\
 <body>\
    <h2>Monitoramento de DataCenter</h2>\
    <p>\
      <i class='fas fa-thermometer-half' style='color:#ca3517;'></i>\
      <span class='dht-labels'>Temperatura</span>\
      <span>%.2f</span>\
      <sup class='units'>&deg;C</sup>\
    </p>\
    <p>\
      <i class='fas fa-tint' style='color:#00add6;'></i>\
      <span class='dht-labels'>Humidade</span>\
      <span>%.2f</span>\
      <sup class='units'>&percnt;</sup>\
    </p>\
    <p>\
      <i class='fas fa-smoke' style='color:#00add6;'></i>\
      <span class='dht-labels'>Gas</span>\
      <span>%.2f</span>\
    </p>\
 </body>\
</html>",
         readDHTTemperature(), readDHTHumidity(), readGasSensor()
        );
```

```
  server.send(200, "text/html", msg);
}

void setup(void) {


  Serial.begin(115200);
  dht.begin();
  pinMode(LED, OUTPUT);
  pinMode(buzzer, OUTPUT);

  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  Serial.println("");

  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("esp32")) {
    Serial.println("MDNS responder started");
  }
  server.on("/", handleRoot);

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



void loop(void) {

  int sensorValue_digital = analogRead(gas_digital);

  if(sensorValue_digital < 400){
    digitalWrite(LED, HIGH);
  }
  else{
    digitalWrite(LED, LOW);
  }

  server.handleClient();
  delay(2);//allow the cpu to switch to other tasks
}


float readGasSensor() {

  int sensorValue_digital = analogRead(gas_digital);

  Serial.println(sensorValue_digital);
  return sensorValue_digital;
}

float readDHTTemperature() {
  float t = dht.readTemperature();
```

```
  if (isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return -1;
  }
  else {
    Serial.println(t);
    return t;
  }
}

float readDHTHumidity() {
  float h = dht.readHumidity();

  if (isnan(h)) {
    Serial.println("Failed to read from DHT sensor!");
    return -1;
  }
  else {
    Serial.println(h);
    return h;
  }
}
```