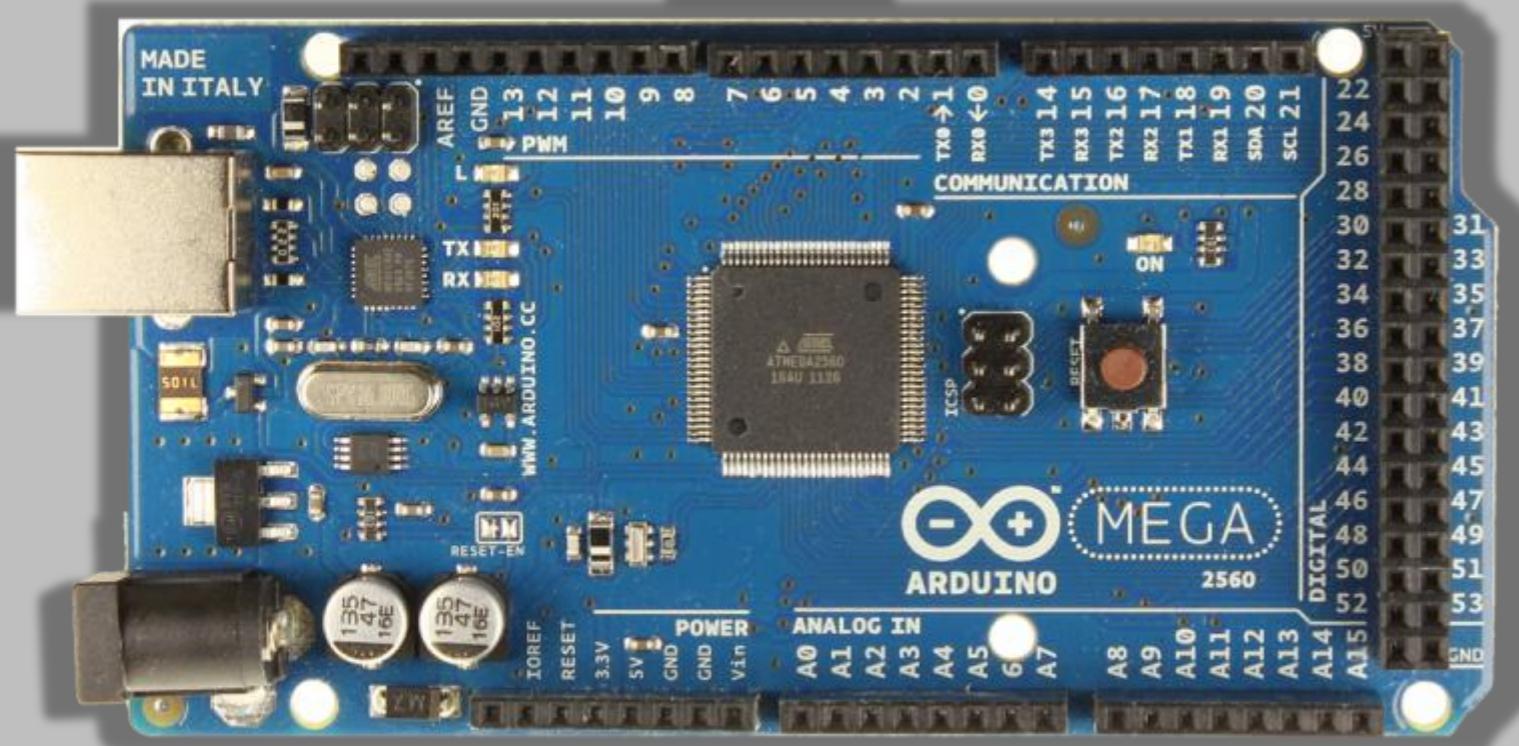
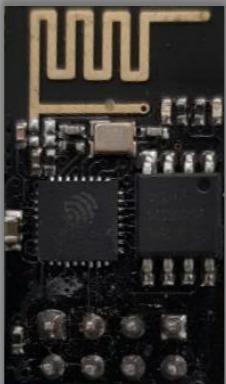


# Arduino Mega + WiFi com ESP-01 - Web Server

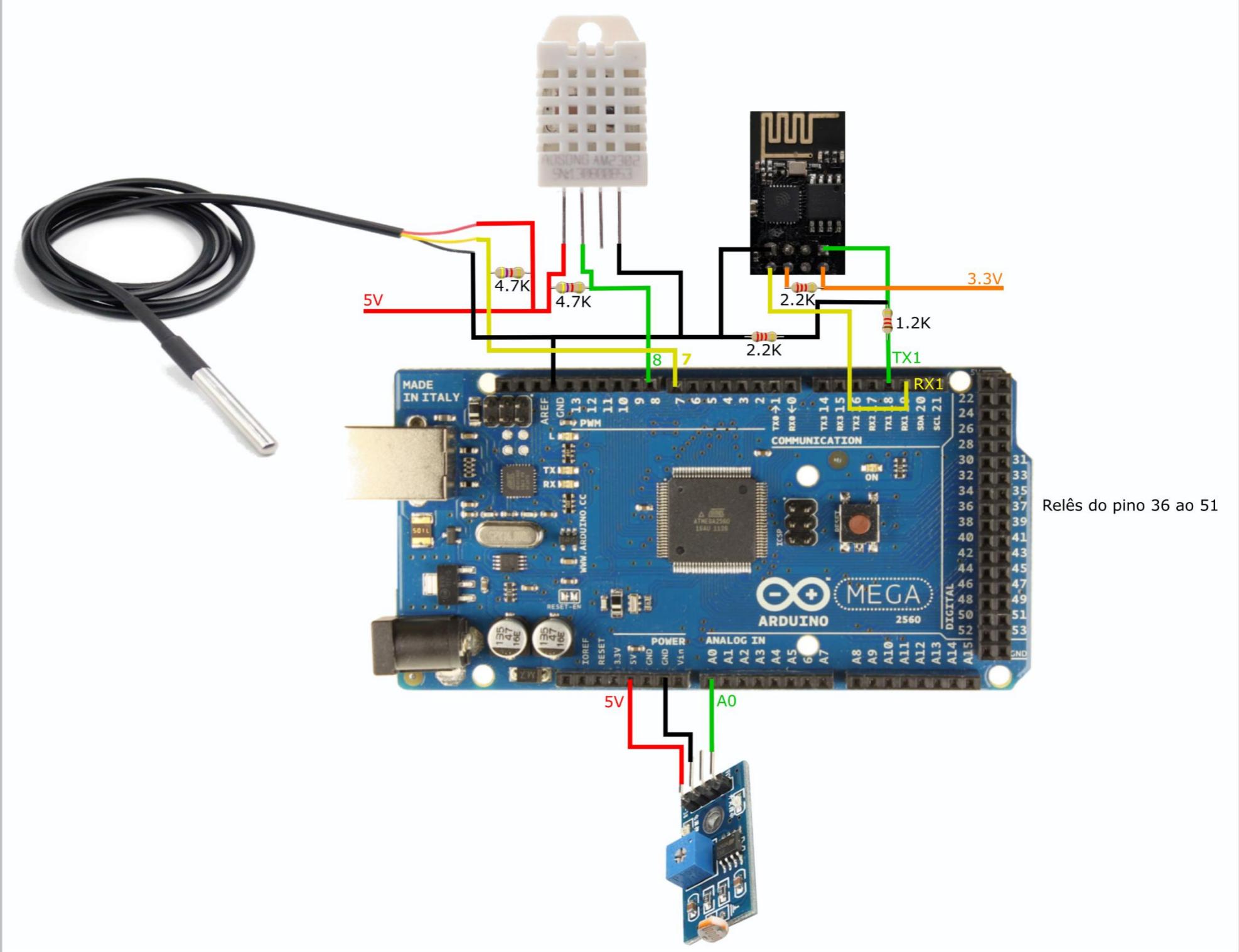


Por Fernando Koyanagi

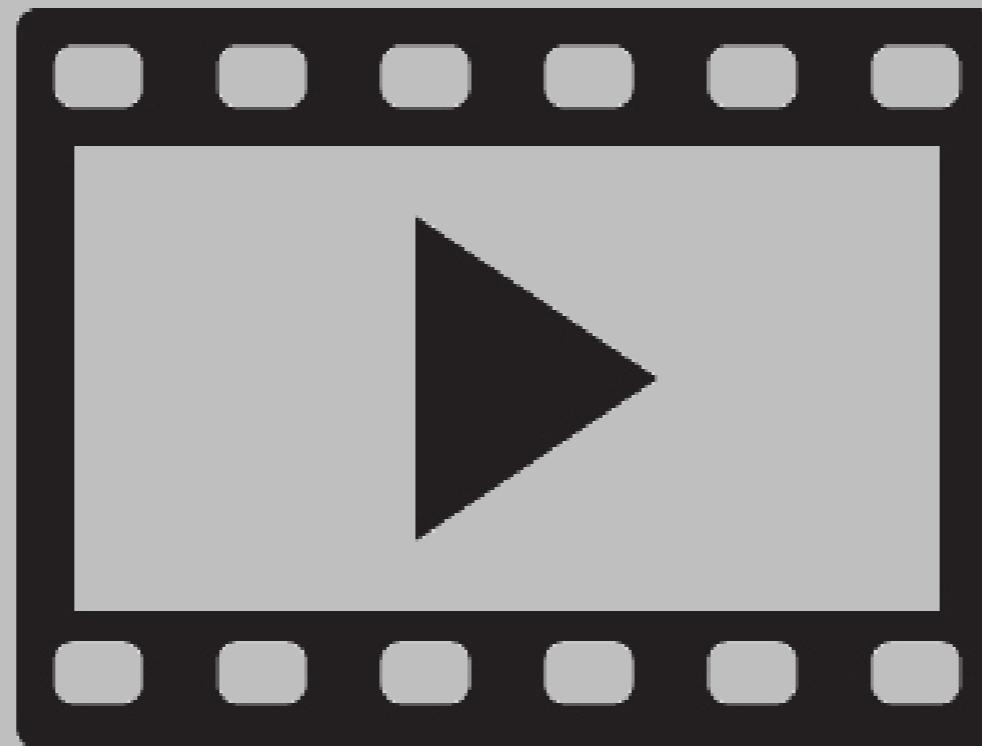
# **Intenção da Aula**

- 1. Fazer um Web Server para o Arduino Mega utilizando comunicação serial com o ESP-01**
  
- 2. Mostrar uma página com os valores de sensores e botões para modificarmos o estado de relês**

# Montagem



# Demonstração



# forum.fernandok.com

Fórum Fernando K Tecnologia  
Fórum sobre dúvidas com relação ao conteúdo disponibilizado pelo Fernando Koyanagi

Pesquisar...

[www.fernandok.com](#) [/fernandokoyanagi](#) [/fernandokoyanagi](#) [/fernandok\\_oficial](#) [/fernandok\\_oficial](#)

Links rápidos: [FAQ](#) [ACP](#) [MCP](#)

Notificações Mensagens privadas [fernandokoyanagi](#)

Bem-vindo: 23/Ago/2018, 14:59 A sua última visita foi em 19/Ago/2018, 08:45

Assinalar todos os fóruns como lidos

**BUROPE FÓRUM FERNANDOK**

	TÓPICOS	MENSAGENS	ÚLTIMA MENSAGEM
Feedback Dúvidas, críticas ou sugestões sobre o Fórum FernandoK. Para demais questões utilize o fórum correto.	3	5	<b>IBM WATSON + NODE-RED</b> por AlcinoVilela 21/Ago/2018, 08:55

**FERNANDO K**

	TÓPICOS	MENSAGENS	ÚLTIMA MENSAGEM
Arduino Projetos de arduino	7	25	<b>Cartucho de Aquecimento de Im...</b> por ralphmhs 17/Ago/2018, 21:52
ESP32 Projetos de ESP32	15	44	<b>Re: ESP32 WIFI to ESPNow to W...</b> por thiagorsl 21/Ago/2018, 22:50
ESP8266 O ESP8266 é um microcontrolador do fabricante chinês Espressif que inclui capacidade de comunicação por Wi-Fi.	15	37	<b>Onde Comprar</b> por PHSecchi 23/Ago/2018, 13:52
LoRa Projetos com LoRa	6	17	<b>ESP32 WiFi Heltec LoRa + NEO ...</b> por alexandrefcg 23/Ago/2018, 11:38
STM32 Projetos com STM32	2	5	<b>Re: STM32L432KC e STM32L476RG</b> por irenil 27/Jul/2018, 18:07
Motor Projetos com motor	3	7	<b>Impressora 3d sucata drive dv...</b> por Magnetran 13/Ago/2018, 17:20
Display Projetos com Display	3	5	<b>Display I2C 2004A no Marlin?</b> por Magnetran 15/Ago/2018, 10:17

**QUEM ESTÁ ONLINE**  
No total, há **4** usuários online :: 1 usuário registrado, 0 invisível e 3 visitantes (baseado em usuários ativos nos últimos 5 minutos)  
O recorde de usuários online foi de **17** em 26/Jul/2018, 18:20

Usuários registrados: [fernandokoyanagi](#)  
Legenda: Administradores, Moderadores globais

**ANIVERSÁRIO**  
Não há aniversários hoje

**ESTATÍSTICA**  
Total de mensagens **483** • Total de tópicos **55** • Total de membros **255** • Novo usuário: [PHSecchi](#)

[Contate-nos](#) [Equipe do fórum](#) [Membros](#) [Excluir todos os cookies deste painel](#) Todos os horários são UTC-03:00

Powered by phpBB® Forum Software © phpBB Limited  
Traduzido por: Suporte phpBB  
Painel de Controle da Administração



**Instagram**  
fernandok\_oficial



**Telegram**  
fernandok\_oficial



Domingo, Janeiro 14 2018



Em [www.fernandok.com](http://www.fernandok.com)

PRINCIPAL SOBRE FERNANDO K ARDUINO ESP8266 ESP32 LORAWAN MOTOR DISPLAY MATERIAIS DOWNLOAD

Receba o meu conteúdo  
GRATUITAMENTE

Insira aqui seu melhor email...

QUERO RECEBER GRÁTIS



Seu e-mail



## Motor de Passo Nema 23 com Driver TB6600 e Arduino Due

by Fernando K Tecnologia - 2:44 PM

Hoje vamos voltar a falar de Motor de Passo. Vamos utilizar um Nema 23 que será controlado por um Driver TB6600 e um Arduino Due. É p...

[Leia mais](#)

## QUAL ASSUNTO VOCÊ TEM?

- Arduino
- ESP8266
- ESP32
- Motor
- Display
- Sensor

You may select multiple answers.

[Votar](#) [Exibir resultados](#)

Votos até o momento: 32  
Dias restantes para votar: 49

## FACEBOOK



## ESP32 Longa Distância - LoRaWan

by Fernando K Tecnologia - 9:46 AM

Neste artigo vamos tratar da LoRaWAN, uma rede que vai longe gastando pouca energia. Mas, o quanto "longe"? Com o chip que uso no vídeo...

[Leia mais](#)



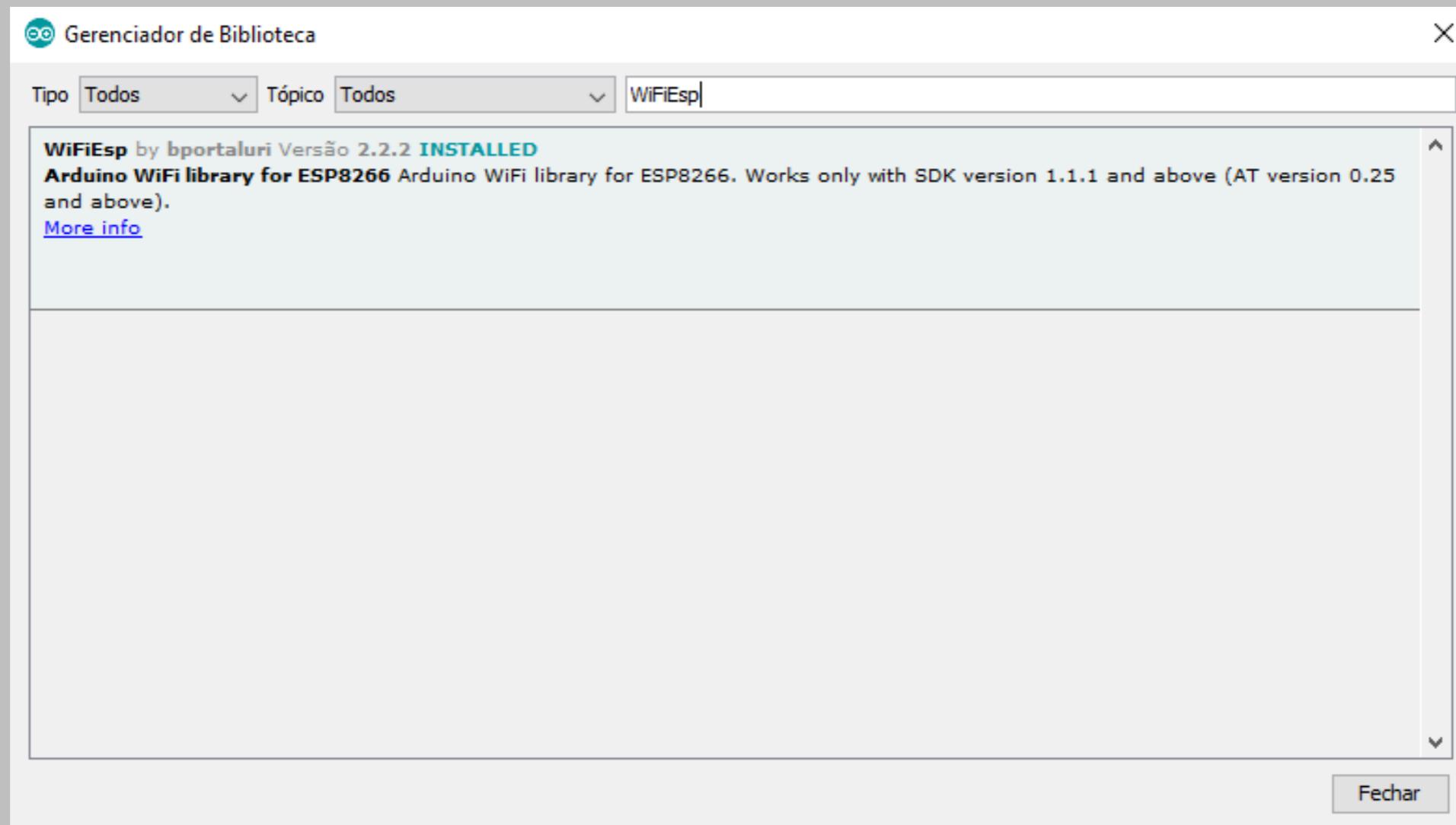
## Motor de HD com Arduino

by Fernando K Tecnologia - 2:00 PM

# Biblioteca WiFiEsp

Na IDE do Arduino vá em Sketch->Incluir Biblioteca->Gerenciar Bibliotecas...

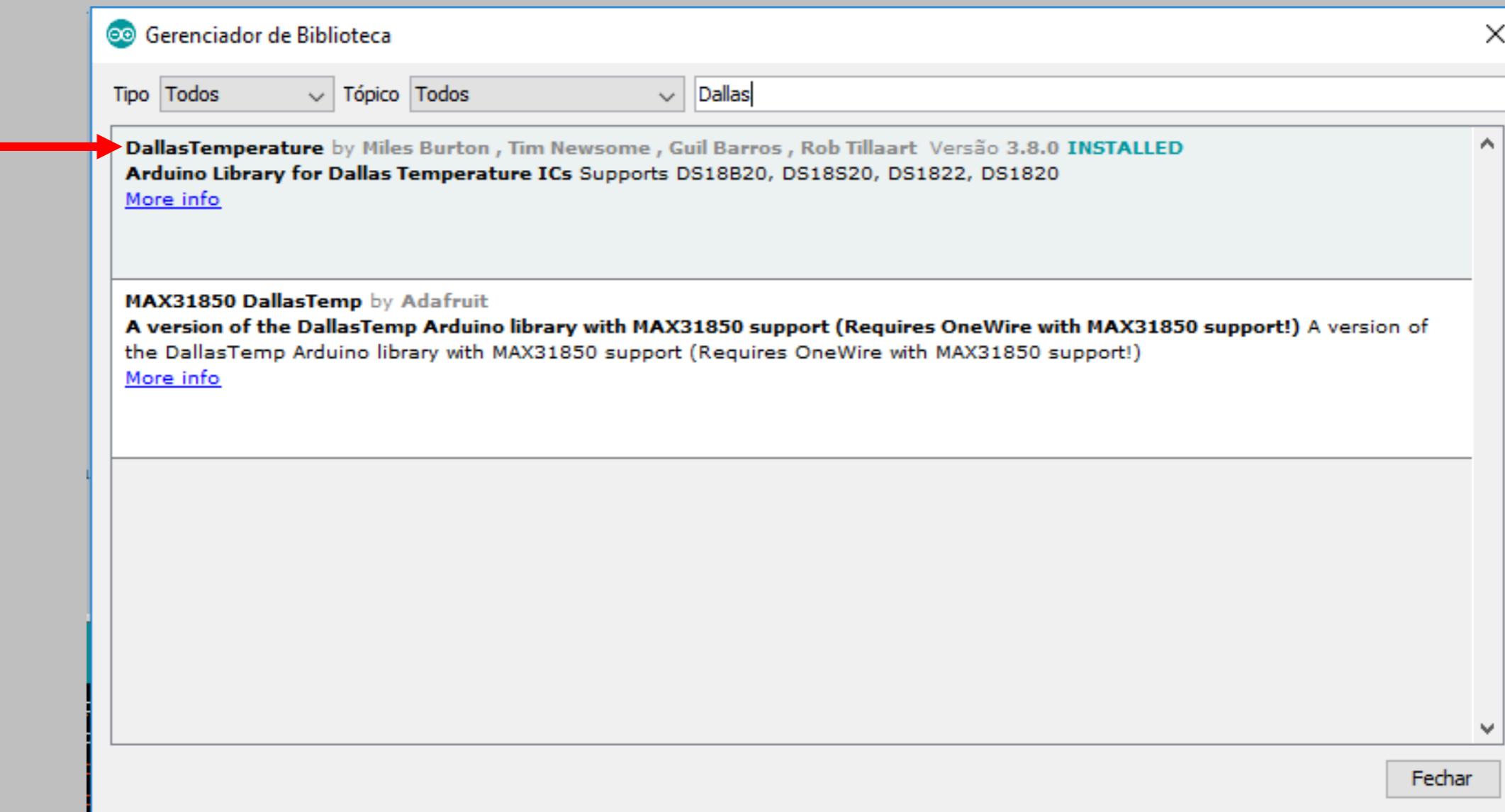
Instale **WiFiEsp**



# Biblioteca DallasTemperature

Na IDE do Arduino vá em Sketch->Incluir Biblioteca->Gerenciar Bibliotecas...

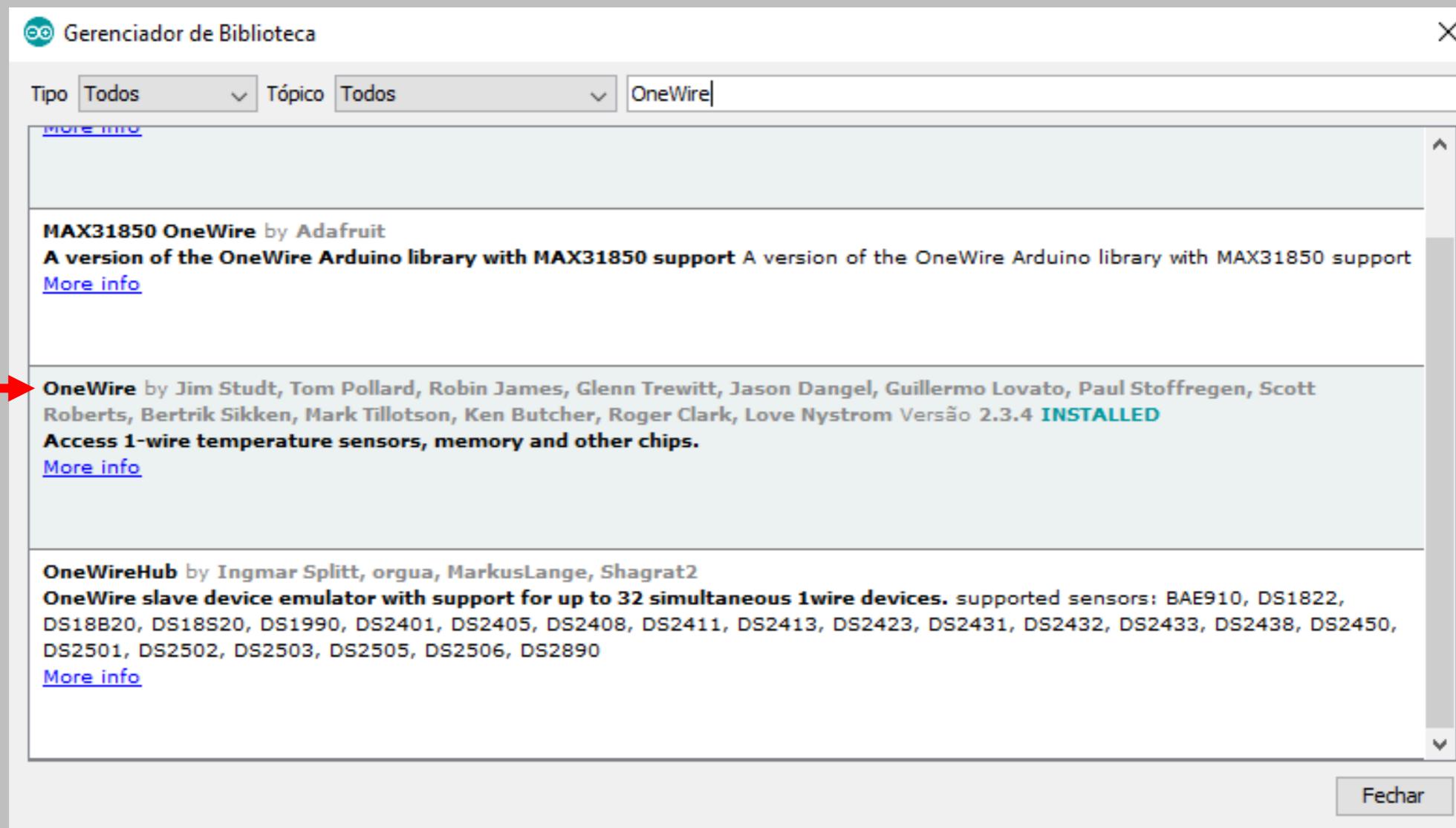
Instale **DallasTemperature**



# Biblioteca OneWire

Na IDE do Arduino vá em Sketch->Incluir Biblioteca->Gerenciar Bibliotecas...

Instale **OneWire**



# MEGAESP\_01.ino

```
#include <OneWire.h>
#include <DallasTemperature.h>
#include <DHT.h>
#include <WiFiEsp.h>

//Pinos onde estão os sensores
#define PIN_DS18B20 7
#define PIN_DHT22 8
#define PIN_LUMINOSITY A0

#define FIRST_PIN 36 //Pino onde está o primeiro relê
#define PINS_COUNT 16 //Quantos pinos a partir do primeiro serão utilizados
```



## MEGAESP\_01.ino

```
//Sensor de Temperatura DS18B20
OneWire oneWire(PIN_DS18B20);
DallasTemperature sensors(&oneWire);
DeviceAddress sensor;

//Sensor de temperatura e umidade DHT22
DHT dht(PIN_DHT22, DHT22);

//SSID e senha da rede wifi para o ESP se conectar
char ssid[] = "SSID";
char pass[] = "12345678";
char ip[] = "192.168.0.109";

//Servidor que receberá as requisições na porta 80 (porta padrão http)
WiFiEspServer server(80);

//Variáveis para armazenar os valores dos sensores
float temperatureDS18B20 = 0;
float temperatureDHT22 = 0;
float humidityDHT22 = 0;
int luminosity = 0;

//Mantém o estado atual dos pinos (HIGH ou LOW)
int pins[DHTC_COUNT];
```



## MEGAESP\_01.ino - setup

```
void setup()
{
    //Serial para o monitor serial
    Serial.begin(115200);

    //Serial onde está o ESP-01 com firmware AT
    Serial1.begin(115200);

    //Inicializa os pinos
    setupPins();

    //Inicializa o sensor DS18B20
    setupDS18B20();

    //Inicializa o sensor DHT22
    dht.begin();

    //Para o sensor de luminosidade apenas precisamos ler o pino analógico
    pinMode(A0, INPUT);

    //Inicializa WiFi e conecta à rede
    setupWiFi();

    //Inicializa o server
    server.begin();
}
```



## MEGAESP\_01.ino - setupPins

```
void setupPins()
{
    //Coloca os pinos que estão ligados os relês como saída
    for(int i=0; i<PINS_COUNT; i++)
    {
        pinsStatus[i] = LOW;
        int pinNumber = FIRST_PIN + i;
        pinMode(pinNumber, OUTPUT);
        digitalWrite(pinNumber, pinsStatus[i]);
    }
}
```



## MEGAESP\_01.ino - setupWiFi

```
void setupWiFi()
{
    //Serial onde está o ESP-01 com o firmware AT já instalado
    WiFi.init(&Serial1);

    Serial.print("Conectando a ");
    Serial.println(ssid);

    int status = WL_IDLE_STATUS;

    //Aguarda conectar à rede WiFi
    while (status != WL_CONNECTED)
    {
        status = WiFi.begin(ssid, pass);
    }

    Serial.println();
    Serial.println("Conectado");

    //Configura o IP
    IPAddress ipAddress;
    ipAddress.fromString(ip);
    WiFi.config(ipAddress);

    //Verifica o IP
    IPAddress localIP = WiFi.localIP();
    Serial.print("IP: ");
    Serial.println(localIP);
}
```



## MEGAESP\_01.ino - setupDS18B20

```
//Inicializa o sensor DS18B20
void setupDS18B20()
{
    sensors.begin();

    if (!sensors.getAddress(sensor, 0))
    {
        Serial.println("Sensor não encontrado!");
    }
}
```



## MEGAESP 01.ino - loop

```
void loop()
{
    WiFiEspClient client = server.available();

    //Verifica se há um novo cliente
    if (client)
    {
        Serial.println("Novo cliente conectou");
        //Faz a leitura da requisição
        char* request = readRequest(client);

        //Se a requisição não for para o favicon
        if(strstr(request, "favicon") == NULL)
        {
            //Executamos a ação com o valor passado na requisição
            execute(getAction(request), getValue(request));

            //Faz a leitura dos sensores
            readSensorDS18B20();
            readSensorDHT22();
            readSensorLuminosity();

            //Envia a resposta ao cliente
            sendResponse(client);

            //Tempo para o navegador receber os dados
            delay(100);
        }
        //Fecha a conexão com o cliente
        client.stop();
    }
}
```



GET /?on=1 HTTP/1.1\r\n

Host: 192.168.3.154\r\n

Connection: keep-alive\r\n

Cache-Control: max-age=0\r\n

Upgrade-Insecure-Requests: 1\r\n

User-Agent: Mozilla/5.0 (Linux; Android 8.0.0; SM-G955F Build/R16N ) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/68.0.3440.91 Mobile Safari/537.36\r\n

Accept:

text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8\r\n

Referer: http://192.168.3.154/\r\n

Accept-Encoding: gzip, deflate\r\n

Accept-Language: en-US,en;q=0.9\r\n

\r\n

**Tem que ler até o final senão a lib  
do wifiesp dá um timeout**

```
//Faz a leitura da primeira linha da requisição
char* readRequest(WiFiEspClient client)
{
    bool currentLineIsBlank = true;
    char request[50];
    int i = 0;
    bool firstLine = true;

    while (client.connected())
    {
        if(client.available())
        {
            char c = client.read();
            //Apenas a primeira linha da requisição nos interessa
            if(firstLine)
            {
                request[i] = c;
                i++;
            }
        }
    }
}
```



## MEGAESP\_01.ino – readRequest 1/2

```
//Faz a leitura da primeira linha da requisição
char* readRequest(WiFiEspClient client)
{
    bool currentLineIsBlank = true;
    char request[50];
    int i = 0;
    bool firstLine = true;

    while (client.connected())
    {
        if(client.available())
        {
            char c = client.read();
            //Apenas a primeira linha da requisição nos interessa
            if(firstLine)
            {
                request[i] = c;
                i++;
            }
        }
    }
}
```



## MEGAESP 01.ino – readRequest 2/2

```
if (c == '\n')
{
    //A última linha da requisição é um \r\n sozinho, após o \r\n da linha anterior
    if(currentLineIsBlank){
        //Se chegou aqui é porque a requisição foi lida por completo
        break;
    }

    currentLineIsBlank = true;
    firstLine = false;
}
else if (c != '\r')
{
    //Se leu qualquer caracter que não for \n e \r significa que a linha não está em
    branco
    currentLineIsBlank = false;
}
}

return request;
}
```



## MEGAESP\_01.ino - sendResponse

```
//Envia o HTML para o cliente
void sendResponse(WiFiEspClient client)
{
    //Envia o cabeçalho HTTP
    client.print(
        "HTTP/1.1 200 OK\r\n"
        "Content-Type: text/html; charset=UTF-8\r\n"
        "Connection: close\r\n"
        "Refresh: 10; URL=/\r\n" //Para fazer requisição a raiz do server a cada 10 segundos
        "\r\n");

    client.println("<!DOCTYPE HTML>");
    client.println("<html>");
    head(client); //Envia o cabeçalho do HTML
    body(client); //Envia o corpo do HTML
    client.println("</html>");
}
```



# MEGAESP 01.ino - head

```
//Envia o CSS para modificar a aparência da página
void head(WiFiEspClient client)
{
    client.println(F("<head>
<style>
body{
    text-align: center;
    font-family: sans-serif;
    font-size: 14px;
}
p{
    color:#555;
    font-size: 12px;
}
.button{
    outline: none;
    display: block;
    border: 1px solid #555;
    border-radius:18px;
    width: 150px;
    height: 30px;
    margin: 10px;
    margin-left: auto;
    margin-right: auto;
    cursor: pointer;
}
.button_off{
    background-color:#FFF;
    color: #555;
}
.button_on{
    background-color:#2C5;
    color: #fff;
}
</style>
</head>"));
}
```



## MEGAESP 01.ino - body

```
//Exibe os dados dos sensores e cria os botões
void body(WiFiEspClient client)
{
    client.println(
        "<body>"
        "DS18B20 Temperature: " + String(temperatureDS18B20) + " °C"
        "<br>"
        "DHT22 Temperature: " + String(temperatureDHT22) + " °C"
        "<br>"
        "DHT22 Humidity: " + String(humidityDHT22) + "%"
        "<br>");

    String buttons = "";

    //Cria um botão para cada pino que possui um relê
    for(int i=0; i<PINS_COUNT; i++)
    {
        buttons.concat(button(i));
    }

    client.println(buttons);
    client.println("</body>");
}
```



## MEGAESP\_01.ino - button

```
//Cria um botão com a aparência e ação correspondente ao estado atual do relê
String button(int number)
{
    String label = String(number + 1);
    String className = "button ";
    className += pinsStatus[number] == HIGH ? "button_on" : "button_off";
    String action = pinsStatus[number] == HIGH ? "off" : "on";
    return "<button class=\"" + className + "\" onclick=\"location.href='?" + action + "=" +
label + "'\">" + label + "</button>";
}
```



## MEGAESP 01.ino - sensors

```
//Faz a leitura do sensor DS18B20
void readSensorDS18B20()
{
    sensors.requestTemperatures();
    temperatureDS18B20 = sensors.getTempC(sensor);
}

//Faz a leitura do sensor DHT
void readSensorDHT22()
{
    humidityDHT22 = dht.readHumidity();
    temperatureDHT22 = dht.readTemperature();
}

//Faz a leitura do sensor de luminosidade
void readSensorLuminosity()
{
    luminosity = analogRead(PIN_LUMINOSITY);
}
```



## MEGAESP\_01.ino – **getAction** **getValue**

```
//Retorna a ação que o cliente deseja executar (on off)
String getAction(char *request)
{
    return getStringBetween(request, '?', '=');
}

//Retorna o valor (numero do relê) que a ação será executada
String getValue(char *request)
{
    return getStringBetween(request, '=', ' ');
}
```



## MEGAESP 01.ino - getStringBetween

```
//Retorna a string que fica entre o primeiro caractere 'start' e o primeiro caractere 'end'
String getStringBetween(char *input, char start, char end)
{
    String str = "";
    //retorna o endereço de memória do caractere 'start'
    char *c = strchr(input, start);

    //Se não achou o caractere
    if(c == NULL)
    {
        return "";
    }

    //Vai para o próximo caractere
    c++;

    //Enquanto não chegar ao caractere 'end' ou ao final da string
    while(*c != end && *c != '\0')
    {
        str += *c;
        c++;
    }

    return str;
}
```



## MEGAESP\_01.ino - execute

```
//Executada a ação junto ao valor (número do relê)
void execute(String action, String value)
{
    //Se é uma das duas ações que esperamos
    if(action == "on" || action == "off")
    {
        //Os relês são numerados a partir do 1, max o array começa do 0
        //então tiramos 1
        int index = value.toInt() - 1;
        //O número do pino será o índice mais o número do pino onde os relês
        //começam. Os relês devem estar em sequência a partir do pino inicial (FIRST_PIN)
        int pinNumber = FIRST_PIN + index;
        int status = action == "on" ? HIGH : LOW;
        digitalWrite(pinNumber, status);
        pinsStatus[index] = status;
    }
}
```



**Em [www.fernandok.com](http://www.fernandok.com)**

**Download arquivos PDF e INO do código fonte**

