MQTT
Instalar a biblioteca PubSubClient.h
Instalar no seu celular os aplicativos Mqttdashboard e Mqttdash

Configurar o arquivo abaixo para seu IP e MAC

```
/*
 Basic MQTT example

 This sketch demonstrates the basic capabilities of the library.
 It connects to an MQTT server then:
  - publishes "hello world" to the topic "outTopic"
  - subscribes to the topic "inTopic", printing out any messages
    it receives. NB - it assumes the received payloads are strings not binary

 It will reconnect to the server if the connection is lost using a blocking
 reconnect function. See the 'mqtt_reconnect_nonblocking' example for how to
 achieve the same result without blocking the main loop.

*/

#include <SPI.h>
#include <Ethernet.h>
#include <PubSubClient.h>

// Update these with values suitable for your network.
byte mac[]    = {  0xDE, 0xED, 0xBA, 0xFE, 0xFE, 0xED };
IPAddress ip(192,168,10,105);

void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i=0;i<length;i++) {
    Serial.print((char)payload[i]);
  }
  Serial.println();
}

EthernetClient ethClient;
PubSubClient client(ethClient);
int i = 0;

void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Attempt to connect
    if (client.connect("arduinoClient")) {
      Serial.println("connected");
      // Once connected, publish an announcement...
      String msg = String(i); i++;
      client.publish("publico/outTopic",msg.c_str());
```

```
      // ... and resubscribe
      client.subscribe("publico/inTopic");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      // Wait 5 seconds before retrying
      delay(5000);
    }
  }
}


long int t,last;

void setup()
{
  Serial.begin(9600);

  client.setServer("mqtt.dti.ufv.br", 1888);
  client.setCallback(callback);

  Ethernet.begin(mac, ip);
  // Allow the hardware to sort itself out
  delay(1500);
  t = last = millis();
}


void loop()
{
  if (!client.connected()) {
    reconnect();
  }
  client.loop();
  t = millis();
  if ( t - last > 1000 ) {
     last = t;
     String msg = String(t);
     client.publish("publico/outTopic",msg.c_str());
  }

}
```

---

cuidado com os conflitos na publicacao de topicos com os colegas…

1. Criar um botão texto para receber o outTopic e um botão para enviar no inTopic.
2. Modificar para ler temperatura ou distancia (DHT, 18b20 ou Ultrasonic)
3. Modificar para ajustar um alarme com um valor que voce envia para temperatura ou distancia.
4. Crie botões para o exemplo do led RGB abaixo

```
#include <SPI.h>
#include <Ethernet.h>
#include <PubSubClient.h>

// Update these with values suitable for your network.
byte mac[]    = {  0xDE, 0xED, 0xBA, 0xFE, 0xFE, 0xED };
IPAddress ip(192,168,10,105);
//IPAddress server(172, 16, 0, 2);
#define RED 3
#define BLUE 6
#define GREEN 5
void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i=0;i<length;i++) {
    Serial.print((char)payload[i]);
    if ( payload[i] == 'r' ) analogWrite(RED,255);
    if ( payload[i] == 'R' ) analogWrite(RED,0);
    if ( payload[i] == 'b' ) analogWrite(BLUE,255);
    if ( payload[i] == 'B' ) analogWrite(BLUE,0);
    if ( payload[i] == 'g' ) analogWrite(GREEN,255);
    if ( payload[i] == 'G' ) analogWrite(GREEN,0);
  }
  Serial.println();
}

EthernetClient ethClient;
PubSubClient client(ethClient);
int i = 0;

void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Attempt to connect
    if (client.connect("arduinoClient")) {
      Serial.println("connected");
      // Once connected, publish an announcement...
      String msg = String(i); i++;
      client.publish("publico/outTopic",msg.c_str());
      // ... and resubscribe
      client.subscribe("publico/inTopic");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      // Wait 5 seconds before retrying
      delay(5000);
    }
  }
}
```

```
long int t,last;

void setup()
{
  Serial.begin(9600);

  client.setServer("mqtt.dti.ufv.br", 1888);
  client.setCallback(callback);

  Ethernet.begin(mac, ip);
  // Allow the hardware to sort itself out
  delay(1500);
  t = last = millis();
  pinMode(RED,OUTPUT);
  pinMode(BLUE,OUTPUT);
  pinMode(GREEN,OUTPUT);

}


void loop()
{
  if (!client.connected()) {
    reconnect();
  }
  client.loop();
  t = millis();
  if ( t - last > 1000 ) {
     last = t;
     String msg = String(t);
     client.publish("publico/outTopic",msg.c_str());
  }

}
```