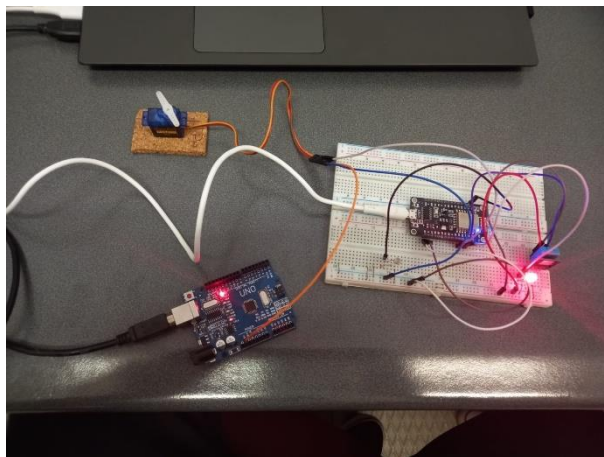


# Internet das Coisas

## Domótica



**Trabalho realizado por:**

José Vale – a40238

Kelson Metzger – a38380

## **Introdução**

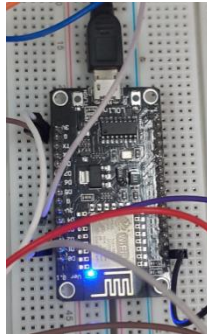
O presente projeto simula equipamentos elétricos e eletrónicos em um edifício, e permite que os mesmos se possam controlar e automatizar, através de um sistema integrado.

O mesmo, tem como objetivo desenvolver um controlo de iluminação local, da temperatura e da qualidade do ar do ambiente e visualização dos mesmos numa dashboard.

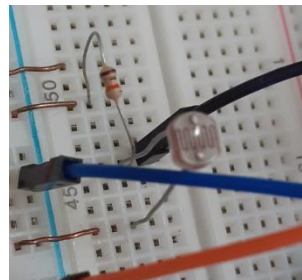
## Materiais

O ESP8266 era o dispositivo que era pedido no enunciado, no entanto, para o desenvolvimento mais fluído do projeto, um dos elementos tinha o ESP-12E e assim o mesmo foi utilizado.

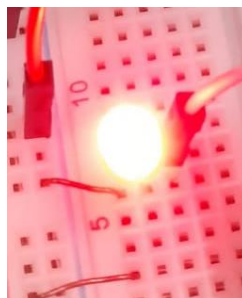
Este dispositivo é um módulo Wi-Fi em miniatura utilizado para estabelecer uma ligação de rede sem fios para um microcontrolador.



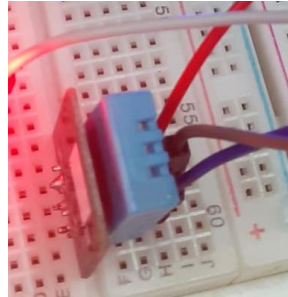
O LDR é um resistor cuja resistência varia conforme a intensidade da luz que incide sobre ele. Normalmente, à medida que a intensidade da luz aumenta, a sua resistência diminui.



Um LED é díodo emissor de luz, que é usado para a emissão de luz em locais e instrumentos onde se torna mais conveniente a sua utilização no lugar de uma lâmpada.



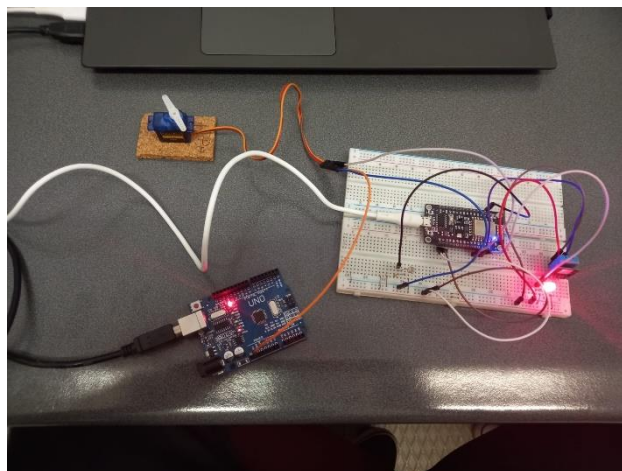
O DHT11 é um sensor digital básico de temperatura e humidade. Utiliza um sensor capacitivo de humidade e um termistor para medir o ar circundante, e emite um sinal digital no pino de dados.



Um Servo Motor é um atuador eletromecânico, que apresenta movimento proporcional a um comando, que recebem um sinal de controle eletrónico; que verifica a posição atual para controlar o seu movimento indo para a posição desejada com velocidade monitorizada.



O circuito final obteve o seguinte resultado, onde ainda adicionamos um Arduino para alimentar o servo motor a 5V pois o Esp-12E só obtinha 3V.



## Programação

Para a conexão dos componentes foi desenvolvido o seguinte código, onde adicionamos a explicação do mesmo em comentários:

```
#include <ESP8266WiFi.h> |
#include <PubSubClient.h>
#include <DFRobot_DHT11.h>
#include <Servo.h>

#define DHT11_PIN 4
#define MOTOR_PIN 2
#define SENSOR_LUZ_PIN A0
#define LED_PIN 5
#define VALUE_LUZ 200

DFRobot_DHT11 DHT;
Servo myservo; // create servo object to control a servo
// twelve servo objects can be created on most boards

int sensorValue = 0; // variable to store the value coming from the sensor
int pos=0;

#define pinBotaol 12 //D6

//WiFi
const char* SSID ="DESKTOP-ARCU300 8640"; //"mente";//DESKTOP-ARCU300 8640
// SSID / nome da rede WiFi que deseja se conectar
const char* PASSWORD = "H7337;8v"; //H7337;8v // Senha da rede WiFi que deseja se conectar
WiFiClient wificlient;

,
//MQTT Server
const char* BROKER_MQTT = "broker.hivemq.com"; //URL do broker MQTT que se deseja utilizar
int BROKER_PORT = 1883; // Porta do Broker MQTT

//Informe um ID unico e seu. Caso sejam usados IDs repetidos a ultima conexão irá sobrepor a anterior.
#define ID_MQTT "BCI01"
#define TOPIC_PUBLISH "IPB_IOT_CeDRI"
#define TOPIC_SUBSCRIBE "IPB_IOT_CeDRI_J"

//Informe um Tópico único. Caso sejam usados tópicos em duplicidade, o último irá eliminar o anterior.
#define TOPIC_PUBLISH2 "IPB/IoT/Lab/ExternalNode"
PubSubClient MQTT(wificlient); // Instancia o Cliente MQTT passando o objeto espClient

//Declaração das Funções
void mantemConexoes(); //Garante que as conexoes com WiFi e MQTT Broker se mantenham ativas
void conectaWiFi(); //Faz conexão com WiFi
void conectaMQTT(); //Faz conexão com Broker MQTT
void enviaPacote(); //
void recebePacote(char* topic, byte* payload, unsigned int length);

void setup() {

    pinMode(LED_PIN, OUTPUT);
    myservo.attach(MOTOR_PIN); // attaches the servo on GIO2 to the servo object
    myservo.write(pos);
```

```
        Serial.begin(115200);
        conectaWiFi();
        MQTT.setServer(BROKER_MQTT, BROKER_PORT);
        MQTT.setCallback(recebePacote);
    }

    void loop() {
        mantemConexoes();
        enviaValores();
        MQTT.loop();
    }

    void mantemConexoes() {
        if (!MQTT.connected()) {
            conectaMQTT();
        }

        conectaWiFi(); //se não há conexão com o WiFi, a conexão é refeita
    }

    void conectaWiFi() {

        if (WiFi.status() == WL_CONNECTED) {
            return;
        }

        Serial.print("Conectando-se na rede: ");
        Serial.print(SSID);
        Serial.println("Aguarde!");

        WiFi.begin(SSID, PASSWORD); // Conecta na rede WI-FI
        while (WiFi.status() != WL_CONNECTED) {
            delay(100);
            Serial.print(".");
        }

        Serial.println();
        Serial.print("Conectado com sucesso, na rede: ");
        Serial.print(SSID);
        Serial.print(" IP obtido: ");
        Serial.println(WiFi.localIP());
    }

    void conectaMQTT() {
        while (!MQTT.connected()) {
            Serial.print("Conectando ao Broker MQTT: ");
            Serial.println(BROKER_MQTT);
            if (MQTT.connect(ID_MQTT)) {
                Serial.println("Conectado ao Broker com sucesso!");
                Serial.println("Conectado ao Broker com sucesso!");
                MQTT.subscribe(TOPIC_SUBSCRIBE);
            }
        }
    }
}
```

```
-
    else {
        Serial.println("Noo foi possivel se conectar ao broker.");
        Serial.println("Nova tentatica de conexao em 10s");
        delay(10000);
    }
}

void enviaValores() {
    char txt[30];

    String dados="";
    //Botao Apertado
    //MQTT.publish(TOPIC_PUBLISH, "1");
    DHT.read(DHT11_PIN);
    Serial.print("temp:");
    Serial.print(DHT.temperature);
    Serial.print("  humi:");
    Serial.println(DHT.humidity);

    sensorValue = analogRead(SENSOR_LUZ_PIN);
    Serial.print("  lux:");

    if(sensorValue<VALUE_LUZ){
        digitalWrite(LED_PIN,HIGH);

        }else{
            digitalWrite(LED_PIN,LOW);
        }
        Serial.println(sensorValue);

        dados+=String(DHT.temperature);
        dados+=",";
        dados+=String(DHT.humidity);
        dados+=",";
        dados+=String(sensorValue);
        dados+=",";
        dados+=String(digitalRead(LED_PIN));
        dados+=",";
        dados+=String(pos);
        dados+=",";

        char dat[20];
```

```
for (int i=0;i<dados.length();i++){
    if(dados.charAt(i)==';'){
        i=1000;
    }else{
        dat[i]=dados.charAt(i);
    }
}

Serial.print("  dados:");

Serial.println(dat);

MQTT.publish(TOPIC_PUBLISH, dat);
}
|
void recebePacote(char* topic, byte* payload, unsigned int length)
{
    String msg;

    //obtem a string do payload recebido
    for(int i = 0; i < length; i++)
    {
        char c = (char)payload[i];
        msg += c;
    }

    if(msg=="0"){

        for (; pos > 0; pos -= 1) {
            myservo.write(pos);
            delay(15);
        }

    }

    if(msg=="1"){

        if(pos>90){
            for (; pos >= 90; pos -= 1) {
                myservo.write(pos);
                delay(15);
            }
        }else{

            for (; pos <= 90; pos += 1) {
                myservo.write(pos);
                delay(15);
            }
        }
    }
}
```



```
}
```

```
    if (msg=="2") {
```

```
        for (; pos <= 180; pos += 1) {  
            myservo.write(pos);  
            delay(15);  
        }
```

```
    }
```

```
Serial.print('>');  
Serial.println(msg);  
}
```

---

## Dashboard

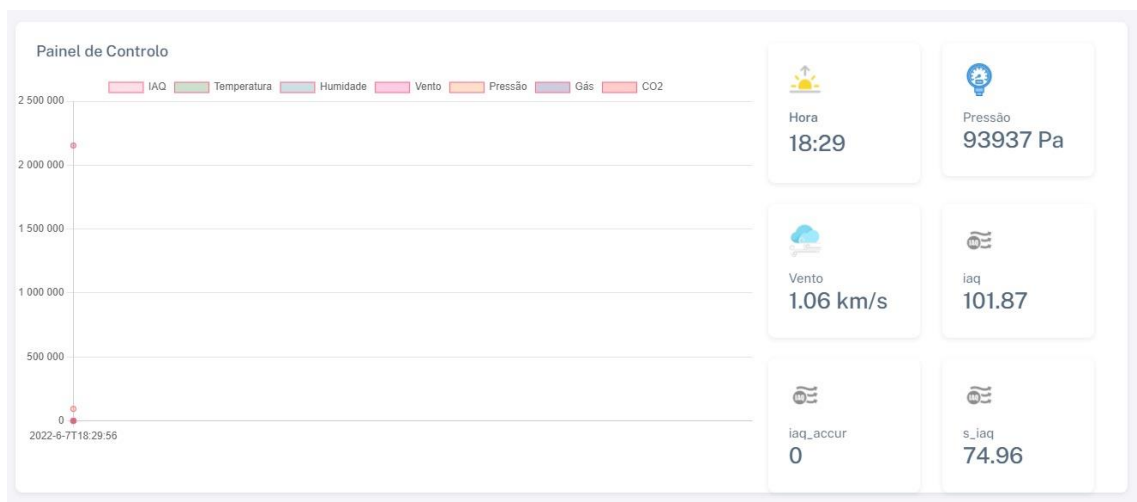
Na realização da dashboard, tentamos realizar a mesma em Node-Red, mas estávamos a ter dificuldades a realizar a mesma no controlo do servo motor(janela).

Assim, a dashboard foi toda ela realizada em javascript e o seu interface gráfico em html e css, onde obtemos o seguinte resultado.

Onde conseguimos obter todos os dados que eram desejados, como a visualização da temperatura, humidade, emissões de CO2, controlar a janela, etc.



Ainda conseguimos guardar todos os dados obtidos ao longo dos 7 dias no “Painel de Controlo” apesar de no mesmo só ser possível registar esses dados se a mesma dashboard estiver sempre em funcionamento.



## **Conclusão**

O presente relatório permitiu demonstrar todo o trabalho desenvolvido na realização do projeto “Domótica”. Todo este projeto constitui uma oportunidade para consolidar e aprofundar todos os conhecimentos adquiridos na unidade curricular de Internet das Coisas.

Assim, obtendo a nosso ver um desempenho positivo, visto que conseguimos realizar todos os requisitos pedidos no projeto e ainda conseguindo ultrapassar dificuldades que foram contornadas com várias pesquisas e muito trabalho em equipa. <https://www.youtube.com/watch?v=YcHniezF9ys>