

UniCEUB – Centro Universitário de Brasília

FATECS – Faculdade de Tecnologia e Ciências Sociais Aplicadas

Curso: Engenharia da Computação

Disciplina: Projeto Final

Monitoramento de ambiente IoT

Autora: Flávia Resende Peixoto

Prof. Orientador: Ivandro da Silva Ribeiro

Sumário

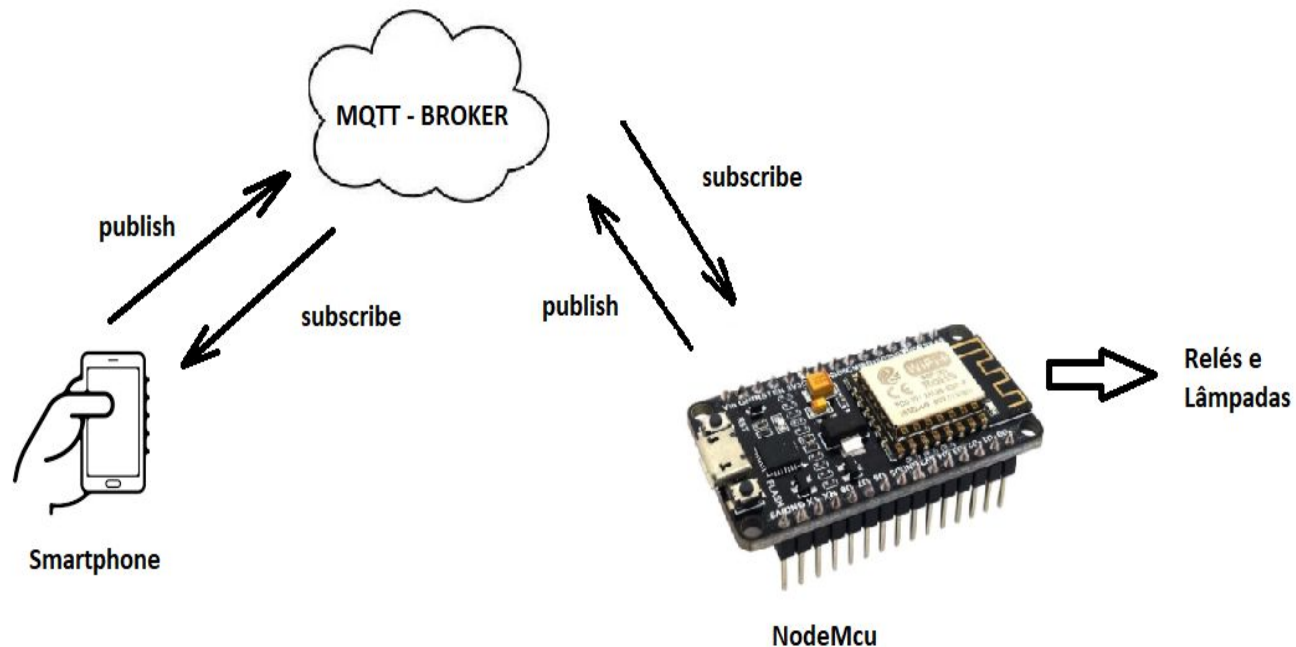


1. Introdução
2. Revisão Bibliográfica
3. Metodologia
4. Apresentação e Análise de resultados
5. Considerações finais

Introdução

- Este projeto tem como principal objetivo desenvolver um protótipo, utilizando-se de sensores de monitoramento de ambiente, construído a partir de componentes de baixo custo.

Modelo simplificado do uso de aplicação nas nuvens com MQTT



Justificativa e Objetivos

- Garantia de segurança é uma das preocupações prioritárias, pois invasões e furtos residenciais são os casos mais frequentes
- Este projeto tem por objetivo contribuir para a constante busca de mais tranquilidade e bem-estar da sociedade.



Recursos Utilizados



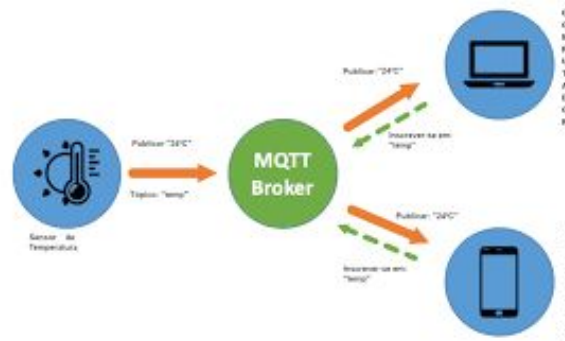
- Internet das Coisas
- Sensores
 - Sensor de temperatura e umidade DHT11
 - Sensor de Presença PIR
 - Sensor de Gás MQ-7
- Protocolo de rede MQTT
- Microcontrolador NodeMCU
- Aplicativo Blynk

Internet das Coisas e MQTT

- É onde dispositivos se tornam capazes de comunicar uns com os outros, serviços ou pessoas em escala global.



- MQTT é um protocolo de rede leve, barato e rápido, muito utilizado em projetos de pequena escala.
- o MQTT define dois tipos de entidades na rede: o broker e os clientes.



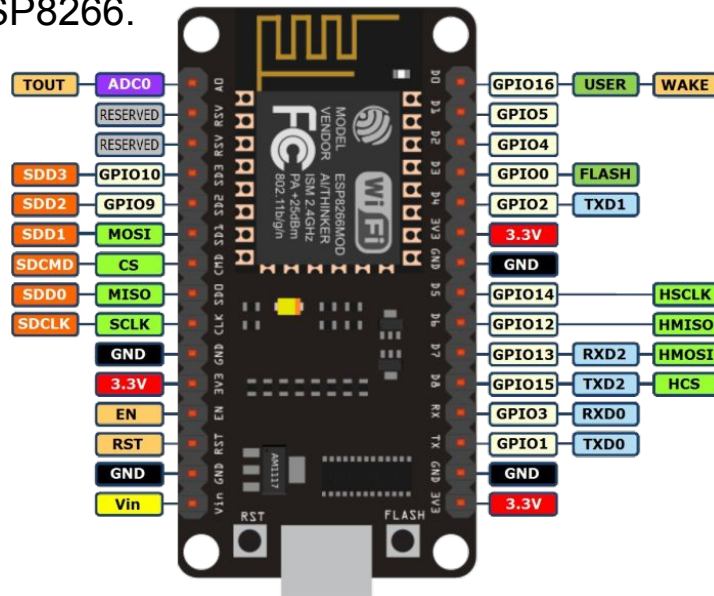
Sensores



- Sensor MQ-7:
 - Este é capaz de detectar gases no ambiente.
- Sensor DHT11:
 - É capaz de medir a temperatura e umidade de um ambiente.
- Sensor de presença PIR
 - Este é capaz de detectar movimento em um ambiente específico.

NodeMCU

- O módulo Wifi ESP8266 NodeMCU é uma placa de desenvolvimento que combina o chip ESP8266, uma interface usb-serial e um regulador de tensão 3.3V.
- Essa placa é muito interessante pelo USB serial integrado.
- Pode ser programada em Luna e Arduino.
- Se comporta como um Arduino, porém é bem mais em conta e possui Wi-Fi nativo, graças a integração com o módulo ESP8266.



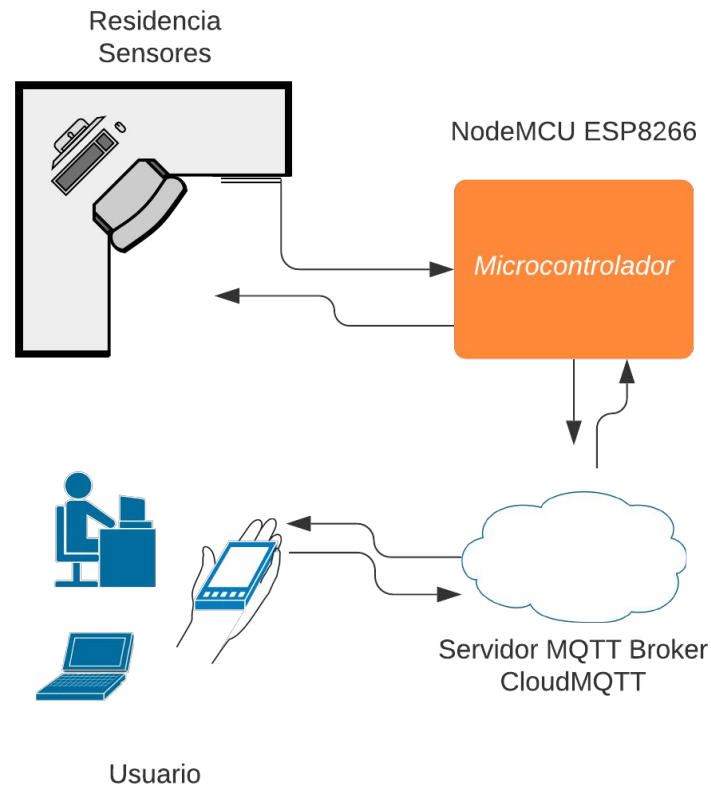
Aplicativo Blynk

- Blynk é um aplicativo para manipulação de microcontroladores pela internet.
- Nesse caso foi utilizado para monitorar o ambiente, pelo nível de Gás CO.



Metodologia

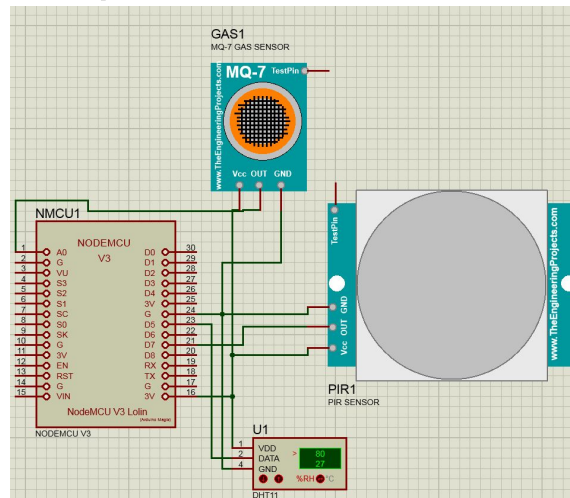
- O desenvolvimento parte de uma base detalhada de um conjunto de hardware e software que permite monitorar um cômodo em uma casa residencial.

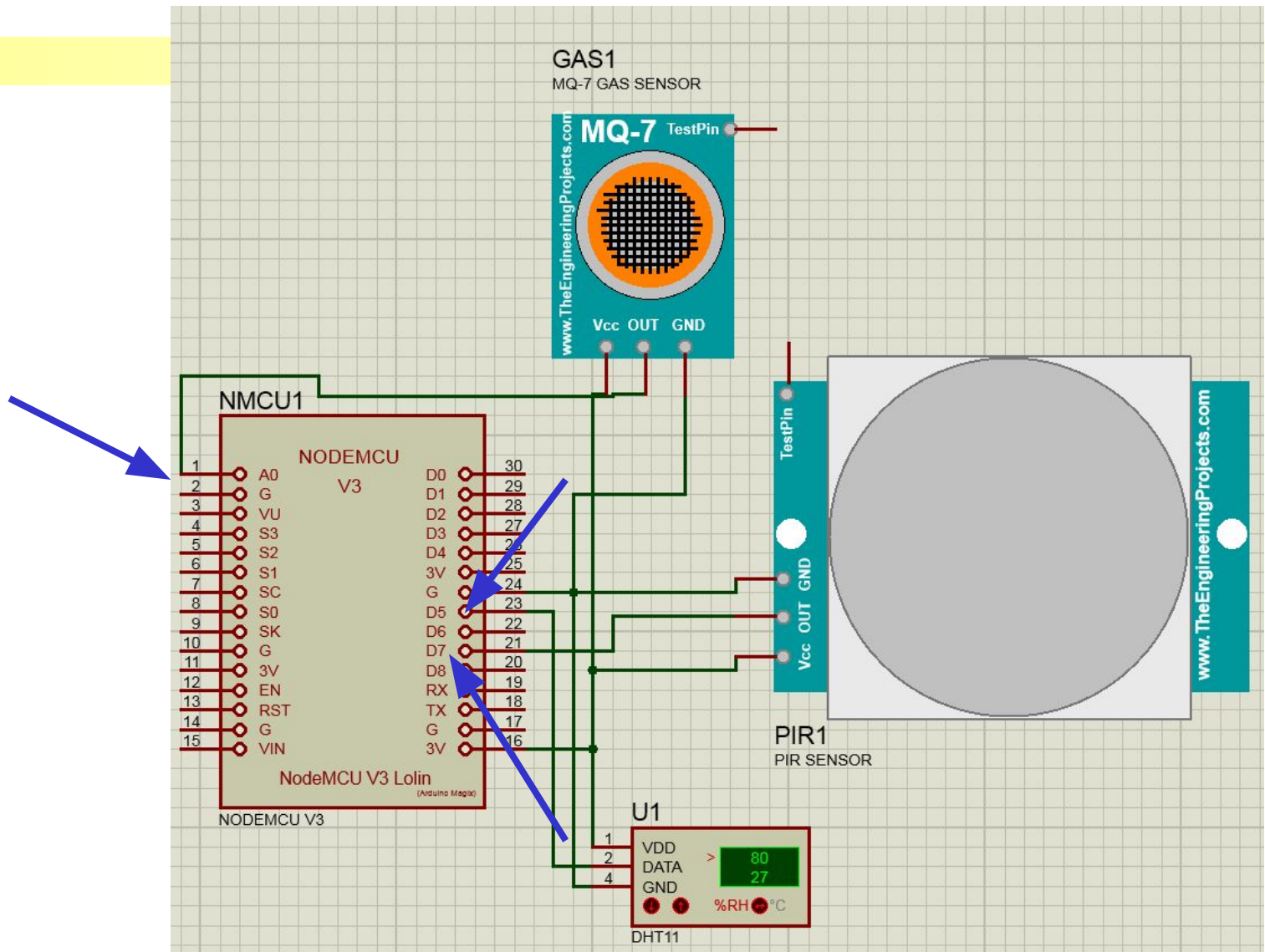


Desenvolvimento

- Primeira etapa: O cômodo escolhido com o protótipo enviará as informações coletadas.
- Segunda etapa: O microcontrolador receberá as informações emitidas pelos sensores, tratará esses dados e enviará para a próxima etapa.
- Terceira etapa: Criação da instância CloudMQTT.
- Quarta etapa: o usuário verá os dados no WebSocket UI do CloudMQTT ou no aplicativo Blynk.

Esquemático do circuito:





Disciplinas utilizadas como base para o Projeto

- Eletrônica para Internet das Coisas
- Sistemas Digitais
- Lógica de Programação
- Circuitos Eletrônicos

Código Arduino

- Conexão com a internet e configuração do servidor Broker

Configuração do Broker:

```
const char* ssid = "PINHONET";// wifi da minha casa
const char* password = "*****"; //Senha, colocamos a que usaremos no WIFI
const char* mqttServer = "tailor.cloudmqtt.com"; // servidor do broker
const int mqttPort = 17678; // porta
const char* mqttUser = "bvxcyhby"; // usuário criado para conexão com o broker
const char* mqttPassword = "Y3q7p47nJrMS"; // senha criada para conexão com o broker
```

Conexão com o broker:

```
void reconnect() { // tentará reconectar ao broker, caso não tenha conseguido
  while (!client.connected()) {
    Serial.println("Connecting to MQTT...");
    WiFi.mode(WIFI_STA);
    if (client.connect("ESP8266Client", mqttUser, mqttPassword )) {

      Serial.println("connected");

    } else {

      Serial.print("failed with state ");
      Serial.print(client.state());
      delay(5000);

    }
  }
}
```

Conexão a rede:

```
void setup_wifi() {

  delay(10);
  // Conectamos a WiFi network
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}
```

Código responsável pelo Blynk

- Para mostrar o nível de CO, o widget de variação do monóxido de carbono foi conectado ao pino A0.
- Os LCDs virtuais na plataforma utilizam pinos digitais criados no código do arduino, V1 e V2 para data e hora. E os pinos virtuais V3 e V4 para umidade e temperatura.
- Os pequenos widgets de smartphone e relógio, são respectivamente para notificações em tempo real, e localização de hora.

```
BlynkTimer timer;

WidgetRTC rtc;

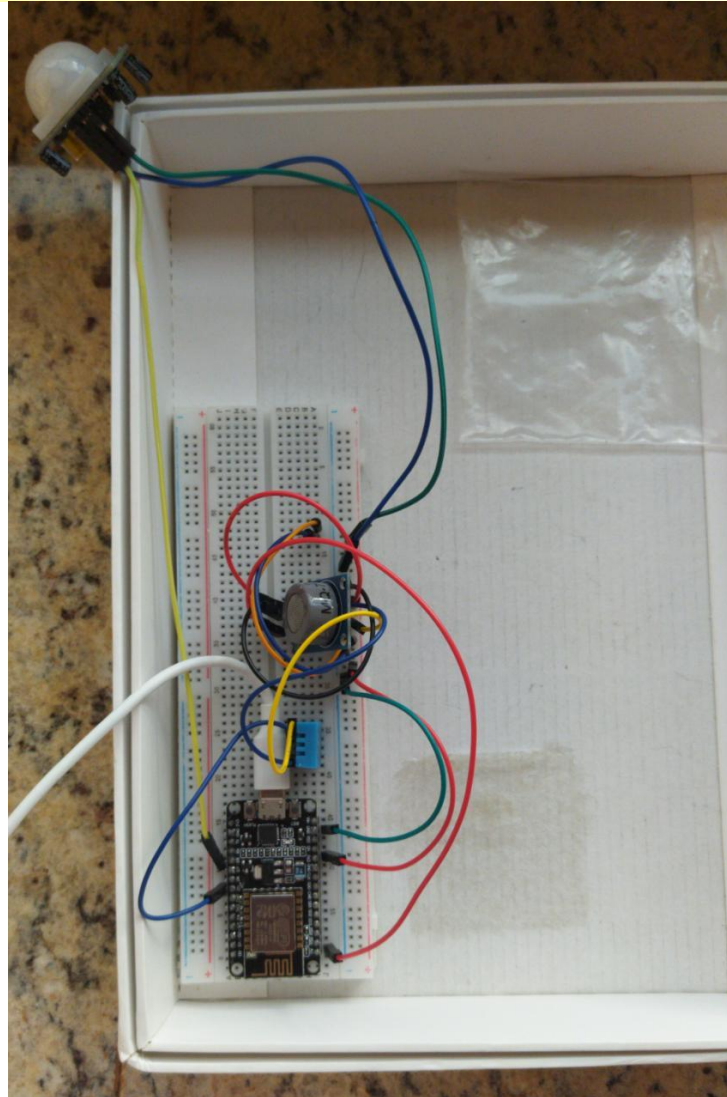
// Digital clock display of the time
void clockDisplay()
{
  // You can call hour(), minute(), ... at any time
  // Please see Time library examples for details

  String currentTime = String(hour()) + ":" + minute() + ":" + second();
  String currentDate = String(day()) + "/" + month() + "/" + year();
  Serial.print("Current time: ");
  Serial.print(currentTime);
  Serial.print(" ");
  Serial.print(currentDate);
  Serial.println();

  // Send time to the App
  Blynk.virtualWrite(V1, currentTime);
  // Send date to the App
  Blynk.virtualWrite(V2, currentDate);
}

BLYNK_CONNECTED() {
  // Synchronize time on connection
  rtc.begin();
}
```


Protótipo



Resultados Obtidos

- Foram realizados testes aumentando o nível de CO do local, fazendo movimento no ambiente e medindo a temperatura e umidade

Tabela 2. Testes temperatura.

Data	Hora	Projeto (°C)	Clima (°C)	Erro (%)
16/06	21:30	18.2	18	1,09
16/06	22:30	16.8	15	10,71
16/06	23:30	14.5	14	3,44
17/06	13:30	24.4	22	9,83
17/06	16:30	22.6	22	2,65
17/06	19:30	18.9	18	4,76
17/06	21:10	17.9	17	5,02
17/06	23:50	16.6	16	3,61
18/06	12:15	22.1	20	9,50
18/06	14:10	27.5	26	5,45
18/06	17:30	25.4	24	5,51
18/06	22:10	17.8	17	4,49
19/06	10:20	18.5	17	8,10
19/06	13:10	21.6	23	6,48
19/06	23:20	19.2	18	6,25
20/06	01:00	17.9	17	5,02

Fonte: Próprio Autor (2020)

Tabela 3. Testes umidade.

Data	Hora	umidade (%)	Clima (%)	Erro (%)
16/06	21:30	81	79	2,40
16/06	22:30	92	80	13,04
16/06	23:30	93	82	11,83
17/06	13:30	64	67	4,68
17/06	16:30	71	73	2,81
17/06	19:30	93	89	4,30
17/06	21:10	92	90	2,17
17/06	23:50	92	90	2,17
18/06	12:15	80	77	3,75
18/06	14:10	52	56	7,69
18/06	17:30	62	59	4,83
18/06	22:10	92	88	4,34
19/06	10:20	93	89	4,30
19/06	13:10	85	79	7,05
19/06	23:20	92	90	2,17
20/06	01:00	93	92	1,07

Fonte: Próprio Autor (2020)

CloudMQTT e Monitor Serial Arduino IDE

Figura 16 – Resultado na plataforma CloudMQTT

Topic	Message
outTopic	18.20 *C!!!
outTopic	93.00 %
outTopic	18.30 *C!!!
outTopic	93.00 %
outTopic	18.30 *C!!!
outTopic	93.00 %
outTopic	18.30 *C!!!
outTopic	93.00 %
outTopic	18.30 *C!!!

Figura 17 - Resultado Monitor Serial Arduino IDE

```
19:11:13.701 -> Temperatura DHT11: 24.30 *C
19:11:13.701 -> Humidade DHT11: 54.00 %
19:11:13.737 -> 436
19:11:13.737 -> GAS DETECTADO !!!
19:11:19.276 ->
19:11:19.276 ->
19:11:19.777 ->
19:11:19.777 -> Temperatura DHT11: 24.30 *C
19:11:19.813 -> Humidade DHT11: 54.00 %
19:11:19.851 -> 414
19:11:19.851 -> GAS DETECTADO !!!
19:11:25.357 ->
19:11:25.391 ->
19:11:25.890 ->
19:11:25.890 -> Temperatura DHT11: 24.30 *C
19:11:25.926 -> Humidade DHT11: 54.00 %
19:11:25.926 -> 402
19:11:25.926 -> GAS DETECTADO !!!
19:11:31.477 ->
19:11:31.477 ->
19:11:31.964 ->
19:11:31.964 -> Temperatura DHT11: 24.30 *C
19:11:32.001 -> Humidade DHT11: 54.00 %
19:11:32.040 -> 393
19:11:32.040 -> GAS AUSENTE !!! e Nenhum movimento detectado
19:11:37.497 ->
19:11:37.497 ->
19:11:37.991 ->
19:11:37.991 -> Temperatura DHT11: 24.30 *C
19:11:38.027 -> Humidade DHT11: 54.00 %
19:11:38.078 -> 387
19:11:38.078 -> GAS AUSENTE !!! e Nenhum movimento detectado
```

Orçamento

Tabela 4. Tabela de Gastos com protótipo.

Material	Valor(R\$)
<u>NodeMCU</u>	39,90
DHT11	13,00
Sensor MQ-7	34,90
Protoboard	19,90
Jumpers	19,80
<u>PowerBank</u>	130,00
Total	257,50

Fonte: Próprio Autor (2020)

O protótipo custou aproximadamente R\$258,00 e mesmo se tratando de um protótipo, pode ser considerado uma alternativa.

Conclusão



- Ao cair da noite a temperatura diminui e a umidade aumenta, como visto nos testes.
- Média dos erros um valor médio:
- Mais ou menos 4,91% para a umidade.
- Aproximadamente 5,74% para temperatura.
- A diferença dos dados pode ter sido ocasionada pela referência de medição.

Contato do Autora



Flávia Resende Peixoto

flaviaresende98@gmail.com

Engenharia da Computação

UniCEUB – Asa Norte – Brasília, DF

OBRIGADA!