UniCEUB – Centro Universitário de Brasília

FATECS – Faculdade de Tecnologia e Ciências Sociais Aplicadas

Major: Engenharia da Computação

Final Project

Environment monitoring with IoT

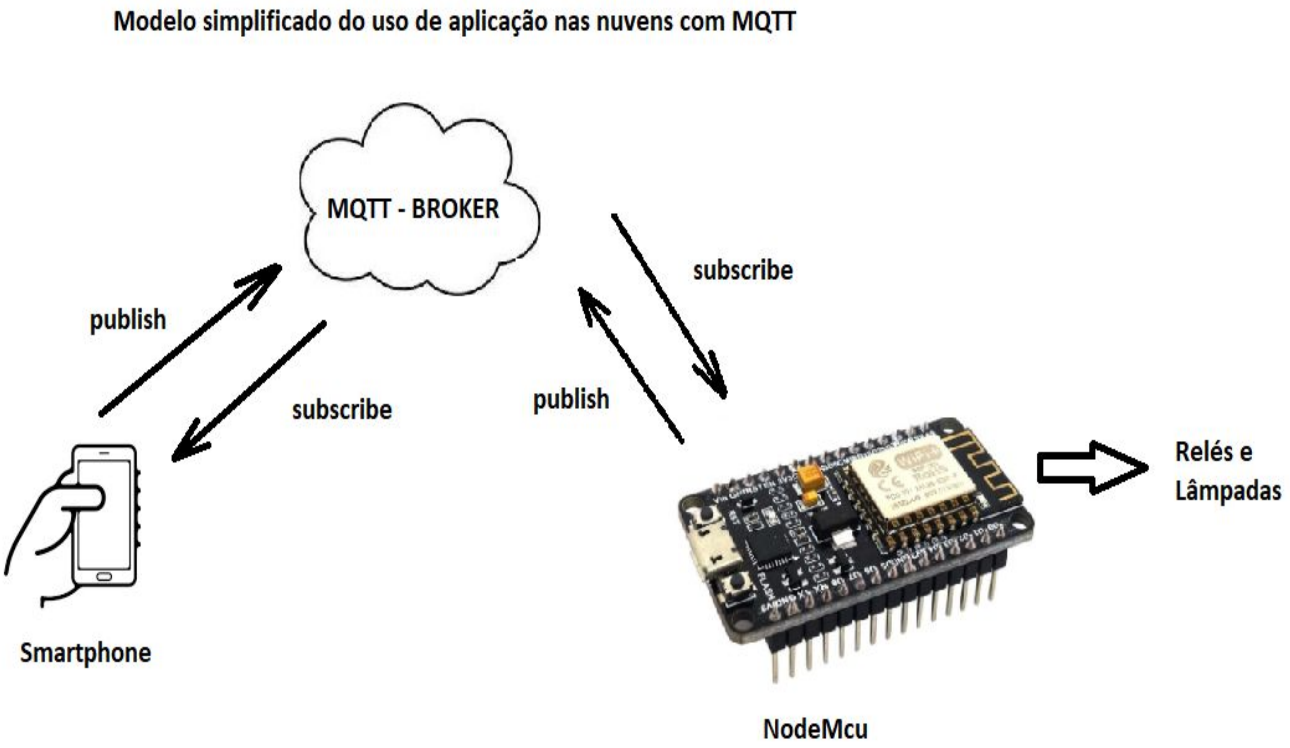
Author: Flávia Resende Peixoto

Professor: Ivandro da Silva Ribeiro

Index

Introduction

- The main objective of this project is to develop a prototype, using environment monitoring sensors, built from low-cost components.



Justification and Objectives

- Assurance of security is one of the priority concerns, as home invasions and thefts are the most frequent cases.
- This project aims to contribute to the constant search for more tranquility and well-being in society.

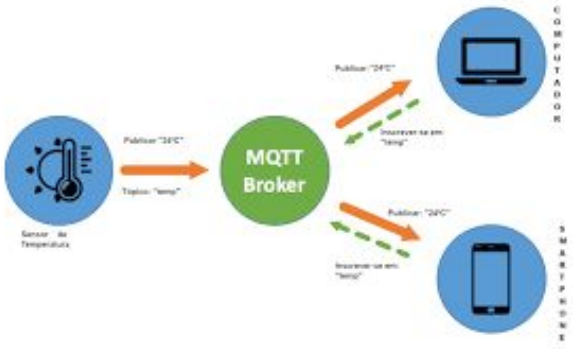
Resources Used

- Internet of Things
- sensors
- DHT11 temperature and humidity sensor
- PIR Presence Sensor
- MQ-7 Gas Sensor
- MQTT network protocol
- NodeMCU microcontroller
- Blynk app

Internet of Things and MQTT

- It is where devices become able to communicate with each other, services or people on a global scale.

- MQTT is a lightweight, inexpensive, and fast network protocol that is widely used in small-scale projects.
- MQTT defines two types of entities in the network: the broker and the clients.

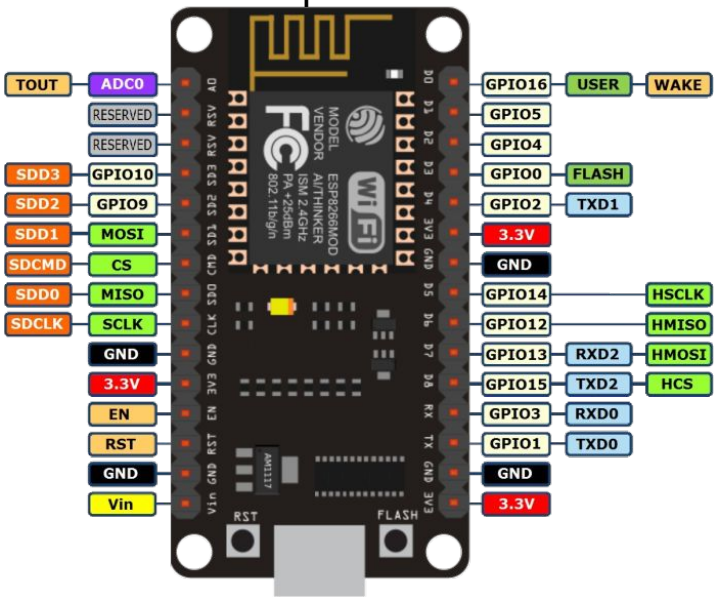


Sensors

- MQ-7 Sensor:
  - This is capable of detecting gases in the environment.
- DHT11 sensor:
  - It is capable of measuring the temperature and humidity of an environment.
- PIR presence sensor
  - This is capable of detecting movement in a specific environment.

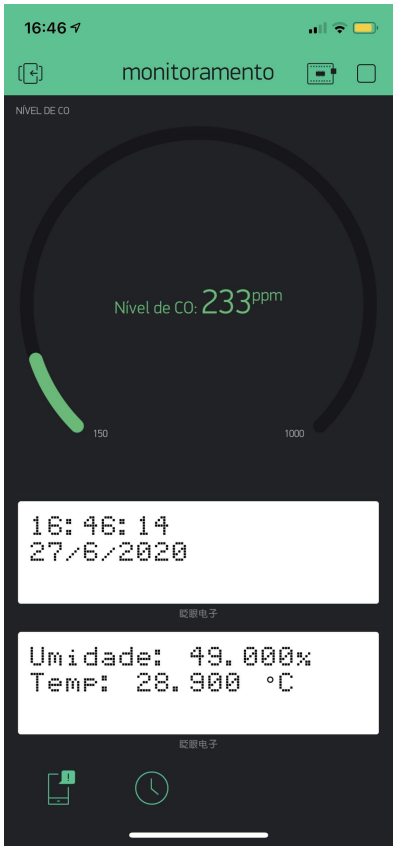
NodeMCU

- The WiFi module ESP8266 NodeMCU is a development board that combines the ESP8266 chip, a usb-serial interface and a 3.3V voltage regulator.
- This board is very interesting for the integrated serial USB.
- It can be programmed in Luna and Arduino.
- It behaves like an Arduino, but it is much cheaper and has native Wi-Fi, thanks to the integration with the ESP8266 module.



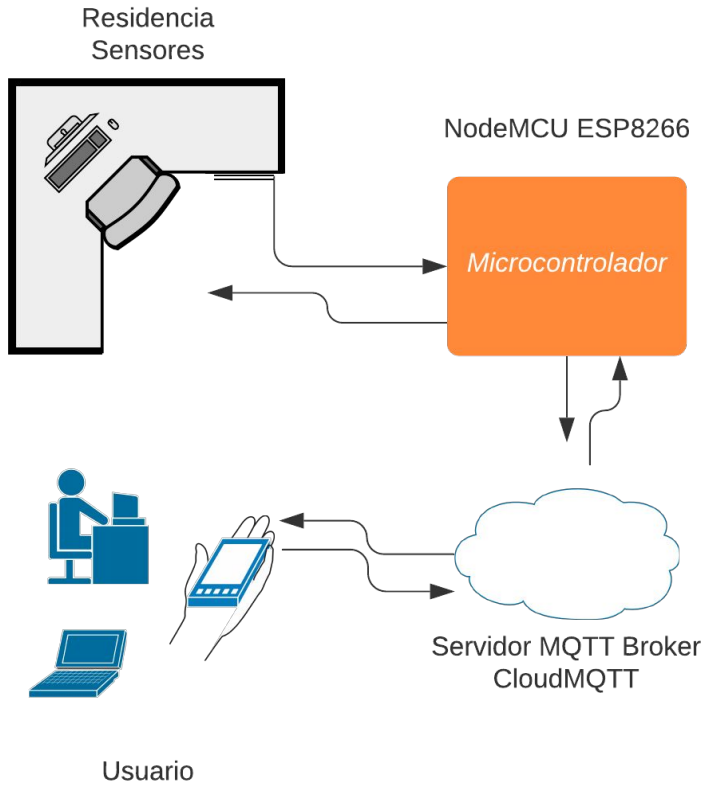
Blynk App

- Blynk is an application for manipulating microcontrollers over the internet.
- In this case it was used to monitor the environment, by the level of CO Gas.



Methodology

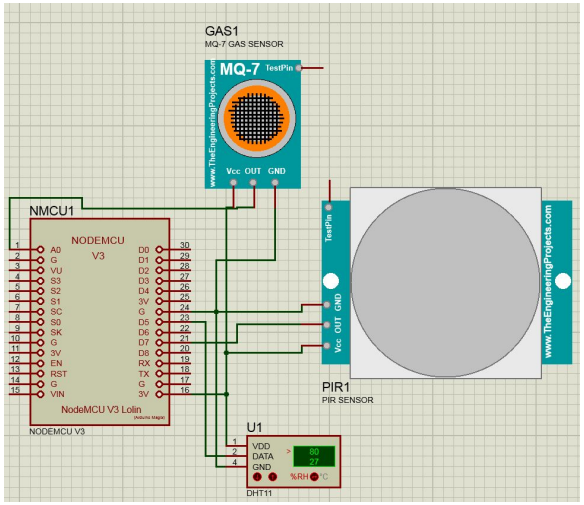
- The development starts from a detailed base of a set of hardware and software that allows monitoring a room in a residential house.

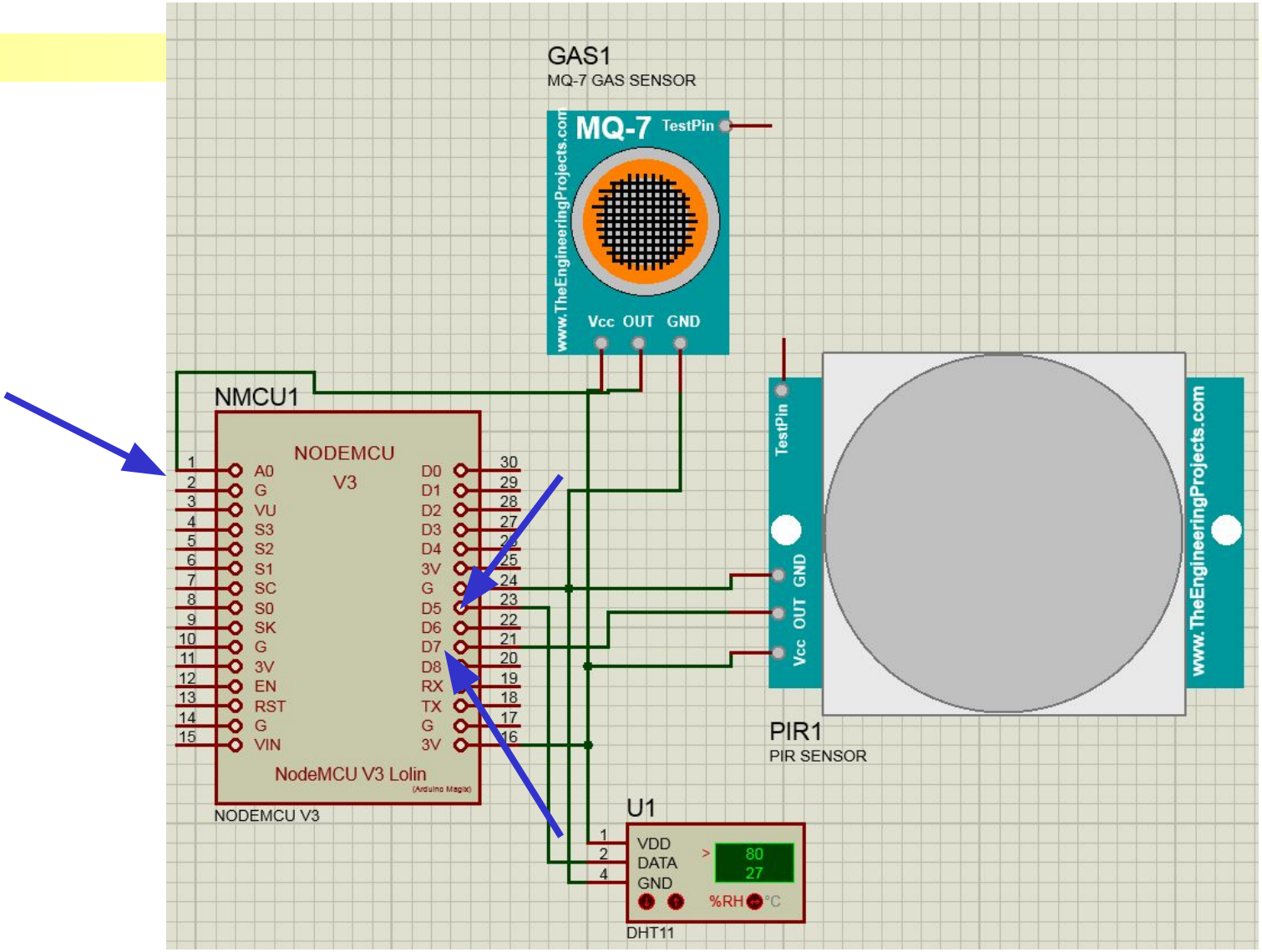


Development

- First step: The room chosen with the prototype will send the information collected.
- Second step: The microcontroller will receive the information emitted by the sensors, process this data and send it to the next step.
- Third step: Creating the CloudMQTT instance.
- Fourth step: The user will see the data in the CloudMQTT WebSocket UI or Blynk app.

Circuit schematic:




GAS1
MQ-7 GAS SENSOR
MQ-7
TestPin
www.TheEngineeringProjects.com
Vcc OUT GND
NMCU1
NODEMCU
V3
A0
G
VU
S3
S2
S1
SC
S0
SK
G
3V
EN
RST
G
VIN
D0
D1
D2
D3
D4
3V
G
D5
D6
D7
D8
RX
TX
G
3V
NodeMCU V3 Lolin
NODEMCU V3
TestPin
Vcc OUT GND
www.TheEngineeringProjects.com
PIR1
PIR SENSOR
U1
VDD
DATA
GND
80
27
%RH
°C
DHT11

Subjects used as a basis for the Project

- Electronics for the Internet of Things
- Digital systems
- Programming logic
- Electronic circuits

Arduino code

- Internet connection and Broker server configuration
  Broker Configuration:

```
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
```

Broker connection:

```
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);

    }
}
}
```

Network connection:

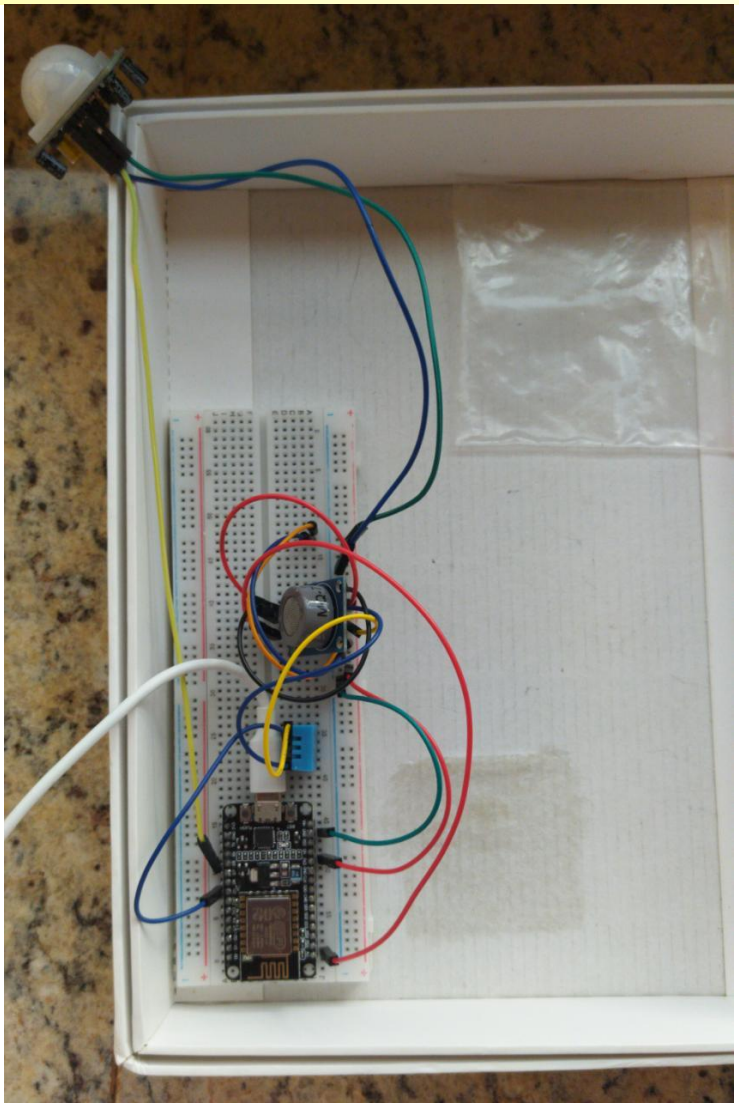
```
void setup_wifi() {

 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());
}
```

Code responsible for Blynk

- To show the CO level, the carbon monoxide variation widget was connected to pin A0.
- The virtual LCDs on the platform use digital pins created in the arduino code, V1 and V2 for date and time. And virtual pins V3 and V4 for humidity and temperature.
- The small smartphone and clock widgets are respectively for real-time notifications and time location.

```
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();
}
```

Prototype



Results Obtained

- Tests were carried out increasing the CO level of the place, making movement in the environment and measuring the temperature and humidity

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 and Arduino IDE Serial Monitor

Figure 16 - Result on the CloudMQTT platform

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!!!

Figure 17 - Arduino IDE Serial Monitor Result

```
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
```

Budget

Tabela 4. Tabela de Gastos com protótipo.

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

Fonte: Próprio Autor (2020)

The prototype cost approximately R$258.00 and even if it is a prototype, it can be considered an alternative.

Final Considerations

- As night falls, the temperature drops and the humidity increases, as seen in the tests.
- Average of errors an average value:
- Plus or minus 4.91% for humidity.
- Approximately 5.74% for temperature.
- The data difference may have been caused by the measurement reference.

Author’s contact

Flávia Resende Peixoto

flaviaresende98@gmail.com

Computer Engineering

UniCEUB – Asa Norte – Brasília,DF

THANK YOU!