

PIMA LITE

ERES LO QUE RESPIRAS

PIMA LITE



Juan Andrés
Holguín



Marcos Sebastián
Andrade



Miguel Ernesto
Figueroa

AGENDA

1. Presentación
2. Introducción
3. Trabajo previo
4. Correcciones fases previas
5. Funcionamiento general de la solución
 - I. Activación remota de un actuador
 - II. Analítica de datos
6. Video
7. Demostración
8. Preguntas

INTRODUCCION

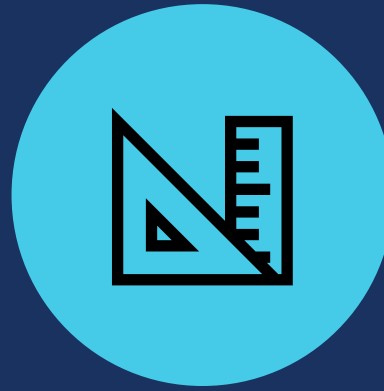
¿Qué es?

¿Para que sirve?

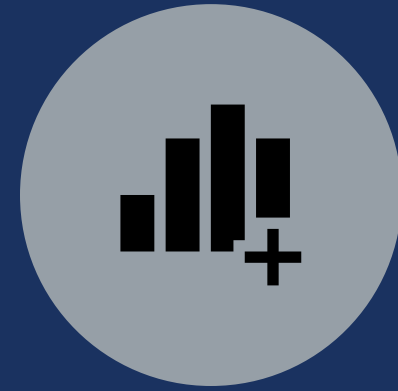
¿Es útil?



FASE I



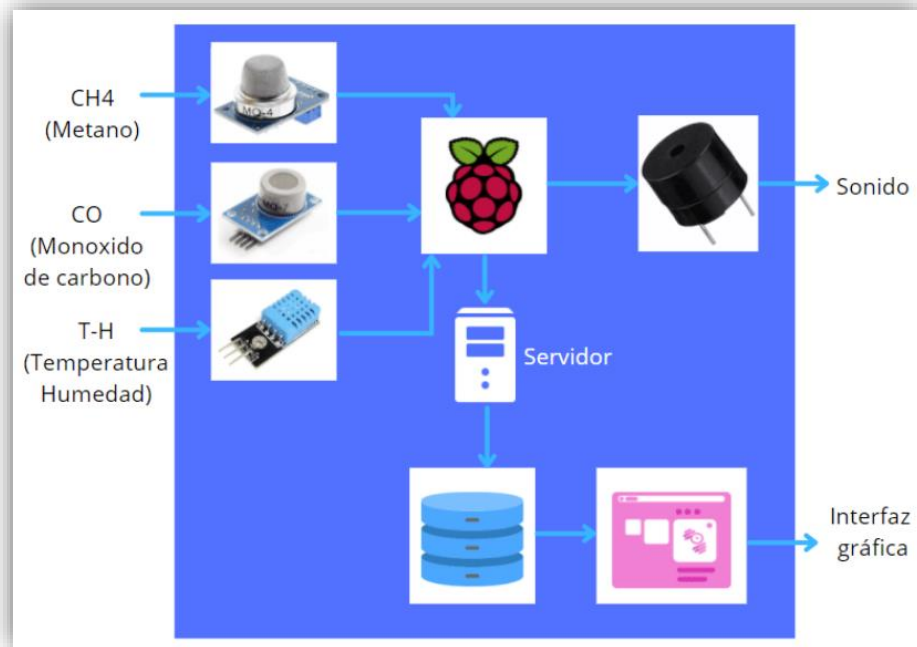
FASE 2



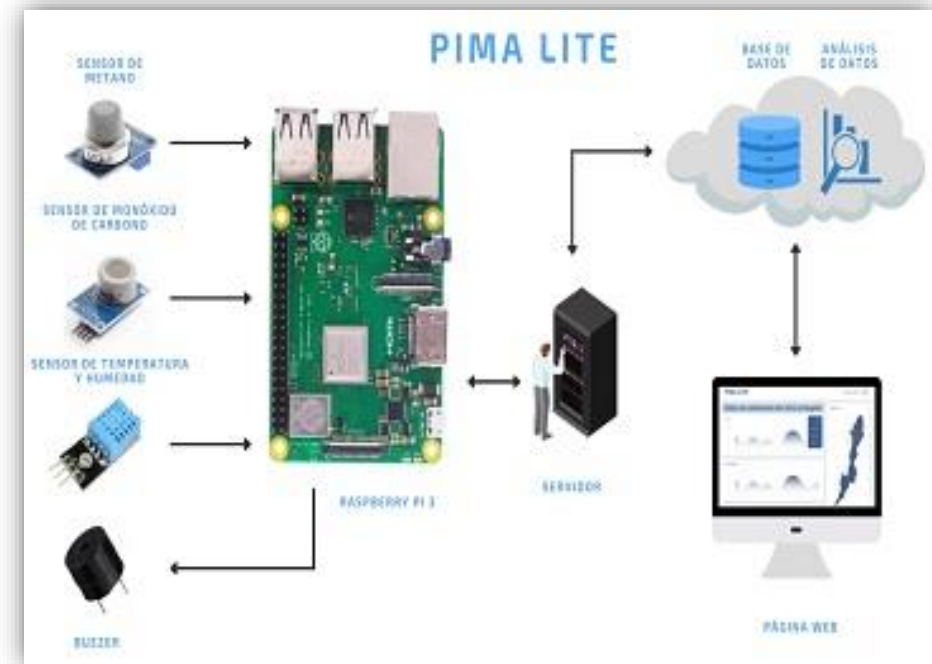
CORRECCIONES

TRABAJO PREVIO

FASE I



FASE 2



FASE I

FASE 2



CORRECCIONES



LOGO



AUDIO



ÚTIL

FUNCIONAMIENTO GENERAL DE LA SOLUCIÓN



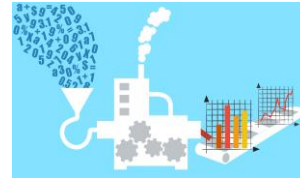
Identificación



Detección



Comunicación



Computación



Servicios



Semántica

1

Sensores+sistema embebido

2

Nodos inalámbricos(MQTT)

3

Internet(análítica básica)ThingSpeak

Nuestro proyecto

ACTIVACIÓN REMOTA DE UN ACTUADOR-CÓDIGO

```
# Importa la librería de GPIO.
import RPi.GPIO as GPIO

# Importa la librería SPI, MCP3008 y DHT.
import Adafruit_GPIO.SPI as SPI
import Adafruit_MCP3008
import Adafruit_DHT

# Importa la librería paho para publicar.
import paho.mqtt.publish as publish
from time import sleep

# Importa la librería para times tamp.
from datetime import datetime
now = datetime.now()

# Configuración de pines tipo BCM.
GPIO.setmode(GPIO.BCM)

# Deshabilitar alarmas GPIO
GPIO.setwarnings(False)

# Configuración de puertos.
GPIO.setmode(GPIO.BCM)
GPIO.setup(4, GPIO.IN)
GPIO.setup(15, GPIO.IN)
GPIO.setup(18, GPIO.IN)
GPIO.setup(17, GPIO.OUT)

# Configuración Hardware SPI.
SPI_PORT = 0
SPI_DEVICE = 0
mcp = Adafruit_MCP3008.MCP3008(SPI_PORT, SPI_DEVICE)

# Configuración DHT.
sensor = Adafruit_DHT.DHT11
pin = 4
buzzer = 17

# Configuración Buzzer
estadoMQ4 = 0
estadoMQ7 = 0
```

```
while True:
    # Lectura del MCP3008 para los canales 0 y 1
    MQ4 = mcp.read_adc(0)
    MQ7 = mcp.read_adc(1)

    # DHT11
    humedad, temperatura = Adafruit_DHT.read_retry(sensor, pin)

    print('Humedad: ', humedad)
    print('Temperatura: ', temperatura)

    # MQ-4
    CH4 = GPIO.input(18)
    print('CH4: ', CH4)
    if GPIO.input(18)==1:
        print("No se detecta CH4")
        sleep(0.2)
    else:
        print("Se detecta CH4")
        sleep(0.1)

    #MQ-7
    CO = GPIO.input(15)
    print('CO: ', CO)
    if GPIO.input(15)==1:
        print("No se detecta CO")
        sleep(0.2)
```

```
# Alarma Buzzer
sleep(0.1)
if humedad > 60:
    GPIO.output(buzzer,True)
    # Alarma riesgo bajo: 1s
    sleep(1)
    GPIO.output(buzzer,False)
    sleep(15)

if temperatura > 30:
    GPIO.output(buzzer,True)
    # Alarma riesgo medio: 2s
    sleep(2)
    GPIO.output(buzzer,False)
    sleep(15)

estadoMQ4 = GPIO.input(18)
if estadoMQ4 ==1:
    GPIO.output(buzzer,True)
    # Alarma riesgo alto: 3s
    sleep(3)
    GPIO.output(buzzer,False)
    sleep(15)

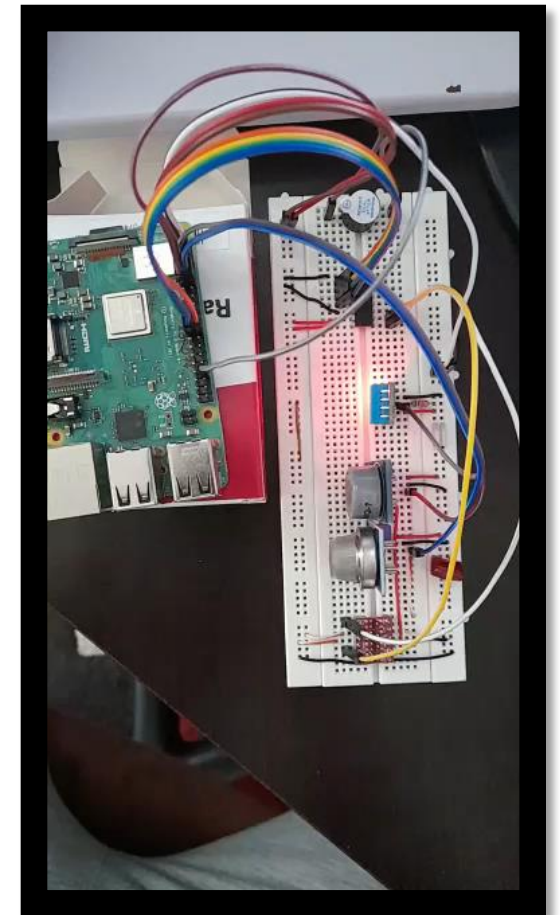
estadoMQ7 = GPIO.input(15)
if estadoMQ7 ==1:
    GPIO.output(buzzer,True)
    # Alarma riesgo alto: 3s
    sleep(3)
    GPIO.output(buzzer,False)
    sleep(15)
```

ACTIVACIÓN REMOTA DE UN ACTUADOR-VIDEO

```
# Almacenamiento de variables
Temperatura = temperatura
Humedad = humedad
Metano = MQ4
Monoxido = MQ7

# Creación del Topic de Thingspeak
topic = "channels/"+1515522+"/publish/"+R3HWFDSTELZGAZP1"

# Creación del mensaje
mensaje = "field1="+str(Temperatura)+"&field2="+str(Humedad)+"&field3="+str(Metano)+"&field4="+str(Monoxido)
+"&field5="+str(now)
try:
    publish.single(topic,payload=mensaje,hostname="mqtt.thingspeak.com",port=1883,tls=None,transport="tcp")
except:
    print("Error 404")
    sleep(15)
```



ANALÍTICA DE DATOS



Regresión lineal



Método de mínimos cuadrados

CÓDIGO ANALÍTICA DE DATOS

```
% Extraer las variables de los datos medidos
temp = Data_Iot(:,1); %temperatura
hume = Data_Iot(:,2); %humedad
meta = Data_Iot(:,3); %metano
mono = Data_Iot(:,4); %monoxido
% Definición del vector de tiempos
time = (1:1:46)';
% Conversión a tablas
temp_s = timetable2table(temp);
hume_s = timetable2table(hume);
meta_s = timetable2table(meta);
mono_s = timetable2table(mono);
% Extraer variable de interés
var_meta = meta_s(:,2);
% Calculo de la pendiente
m = time\var_meta;

% Calculo de los valores para Y
calY = m*time;
calYY = calY(:,1)+755;

% Visualización de las tendencias
scatter(time,var_meta)
hold on
plot(time,calYY)

% Creación del timeStamp
tStamps = datetime('now')-minutes(45):minutes(1):datetime('now');
tStampsF = tStamps';

%% Monoxido
var_mono = mono_s(:,2);

% Calculo de la pendiente
m_mono = time\var_mono;

% Calculo de los valores para Y
calYmono = m_mono*time;
calYYmono = calYmono(:,1)+755;

% Visualización de las tendencias
scatter(time,var_mono)
hold on
plot(time,calYYmono)

%%
% Enviar datos de la tendencia para el Monoxido
thingSpeakWrite(1559284,calYYmono,'WriteKey','TO5W74SMQCXGHJN','TimeStamp',tStampsF);

% Enviar datos de la tendencia para el Metano
thingSpeakWrite(1574509,calYY,'WriteKey','723A1B0DUSR1IKD5','TimeStamp',tStampsF);
```



VIDEO



CONOCE NUESTRO PRODUCTO FINAL

DEMOSTRACIÓN

Web: pima-lite.125mb.com

ThingSpeak: 1515522

Repositorio:
https://github.com/AndresHolguin99/PIMA_LITE



PREGUNTAS

GRACIAS