



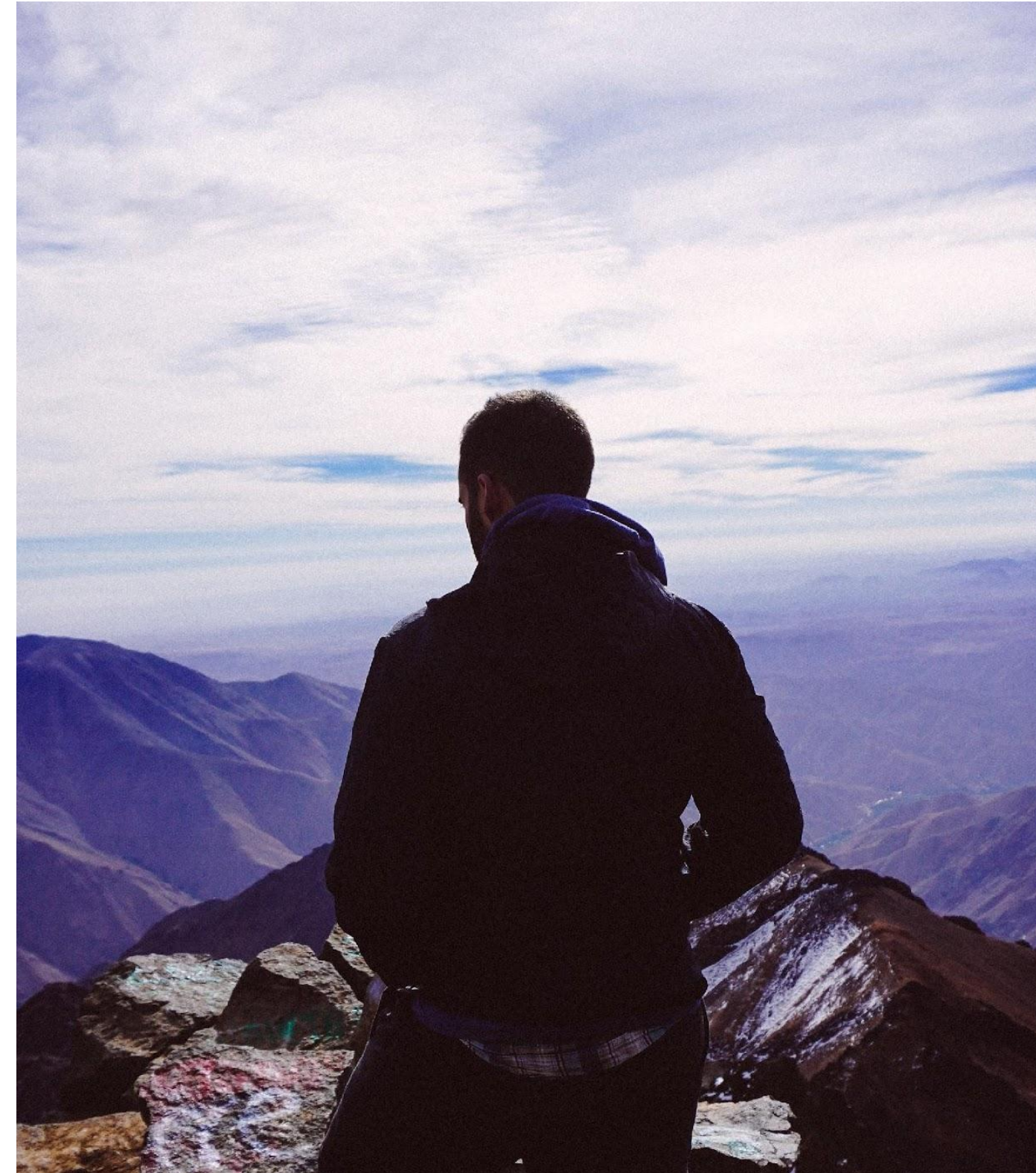
# Reto IOT

Análisis de la calidad del aire

# Etapas

## del Reto IOT

- Inicio
- Planeación
- Ejecución
- Monitoreo y control
- Cierre





# Inicio

Identificación del problema

1

# Mala Calidad de Aire

- ✓ Una manera de detectar calidad del aire
- ✓ Recolección de datos con un modelo físico
- ✓ Información debe ser guardada en una base de datos

Para mejorar la vida de las personas, queremos buscar una manera de recopilar datos sobre la calidad de aire y subir los datos a internet para facilitar el acceso



# ¿Cómo resolvemos este problema?

- ✓ MQ135 (Sensores de aire)
- ✓ Arduino
- ✓ Base de datos

El proyecto va a consistir en usar varias piezas electrónicas y servicios en línea para completar, en este caso planeamos conectar unos pequeños sensores de calidad de aire a un arduino y finalmente mandar los datos a una base de datos en línea.







# Planeación

# 2

(plan de Miro)

# Plan miro

### Equipo 3

[illegible]



# Ejecución

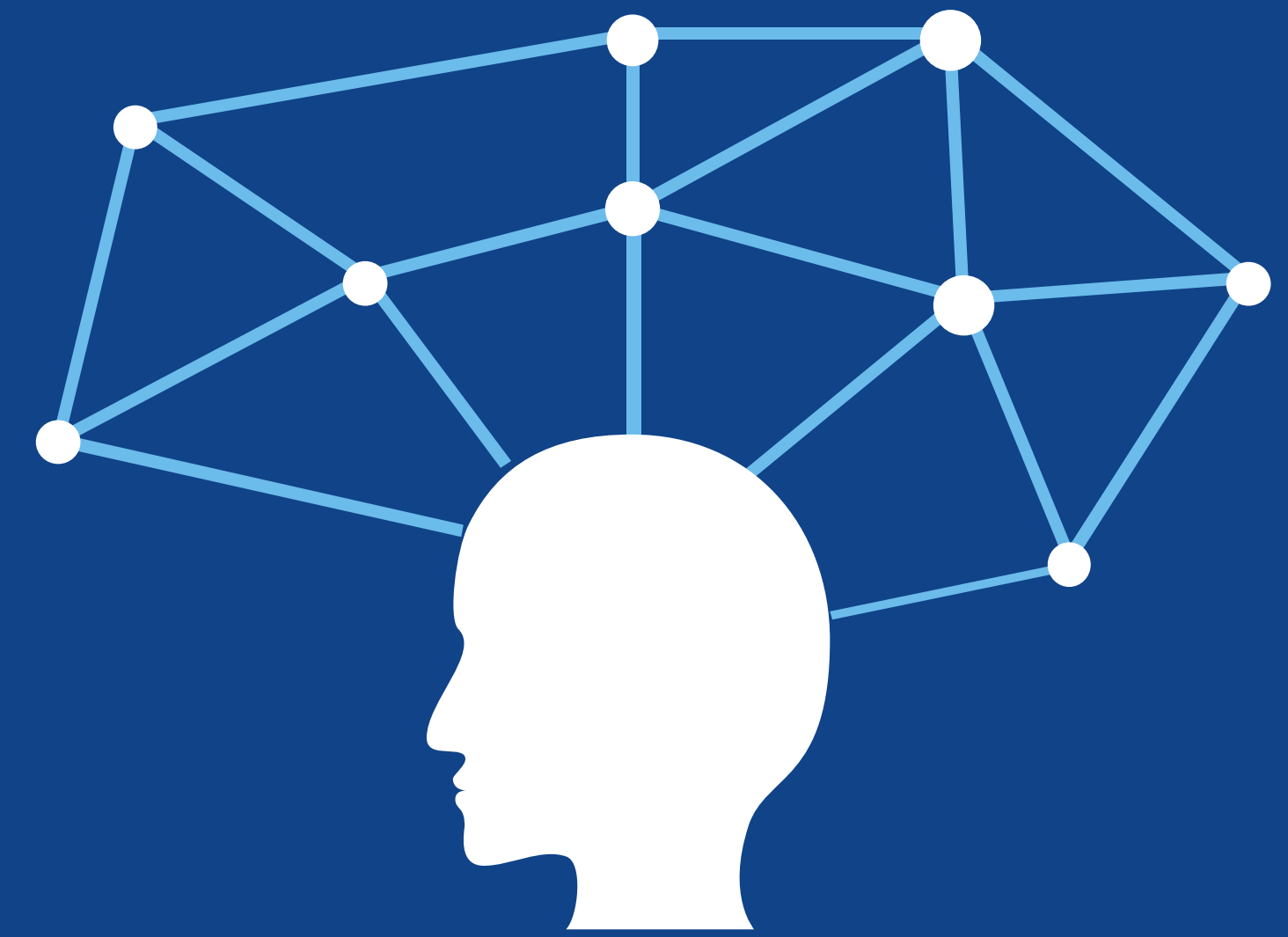
Y pruebas físicas

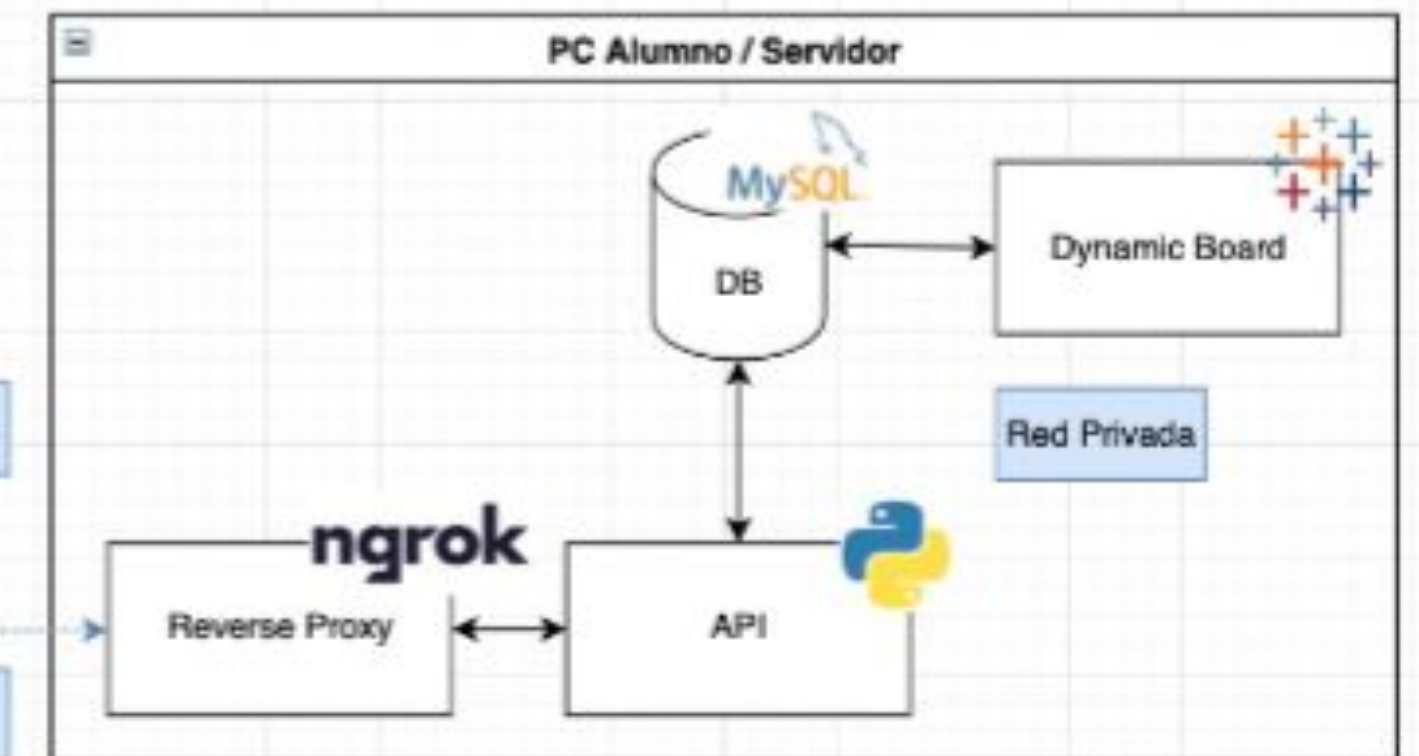
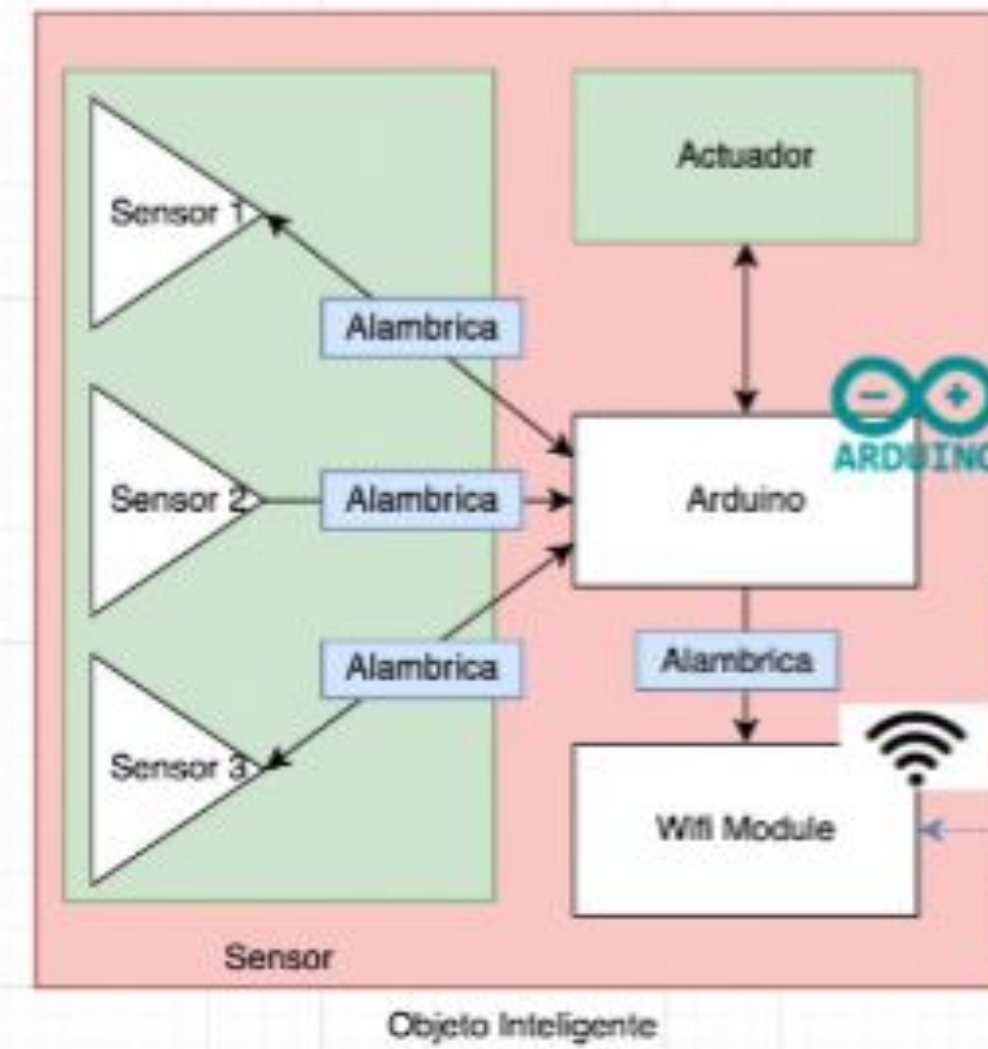
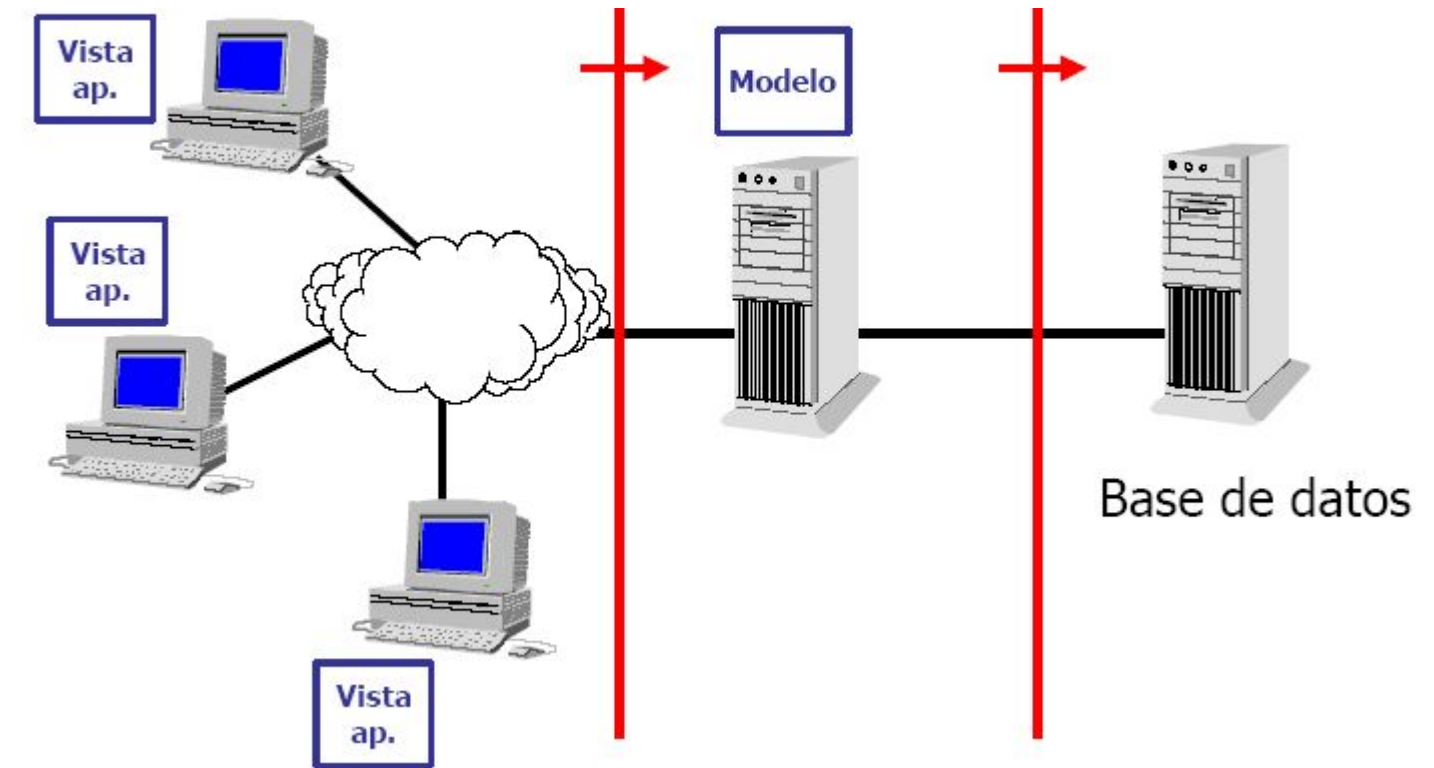
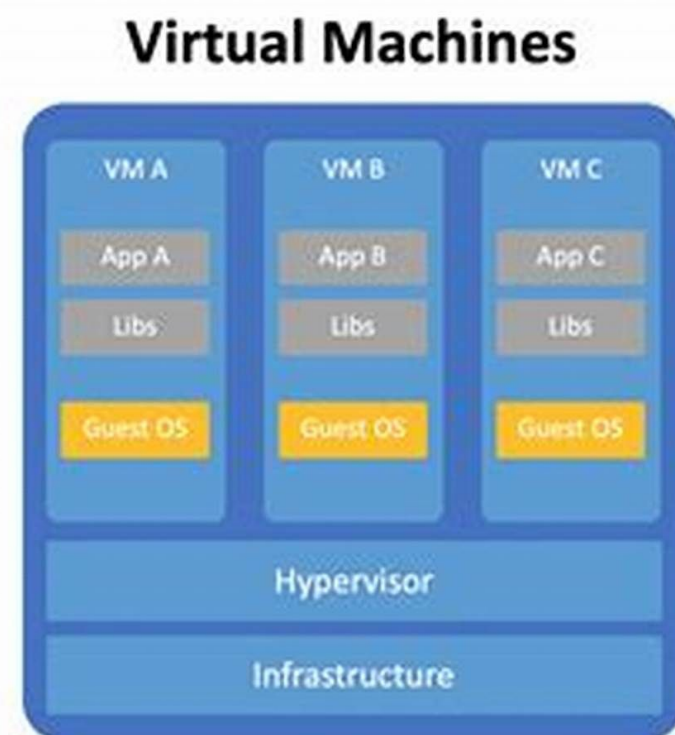
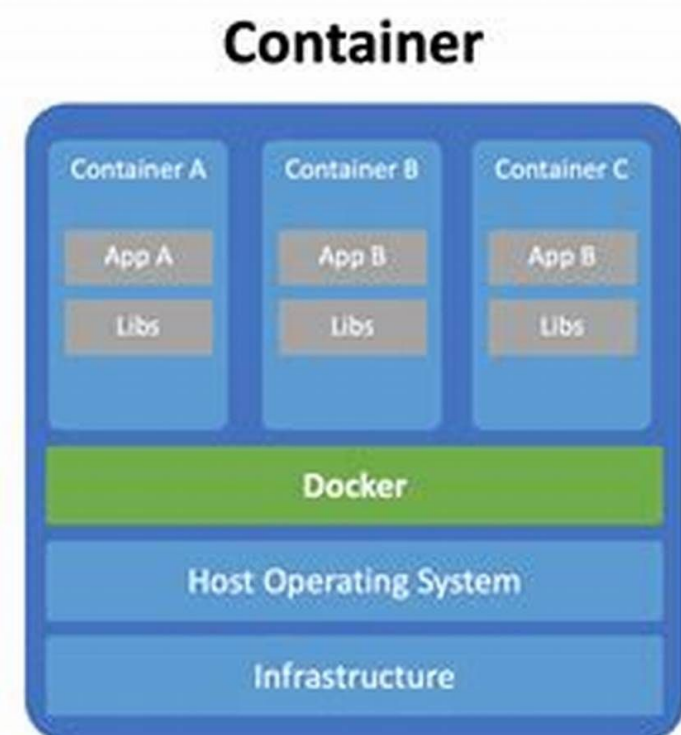
3



# Software

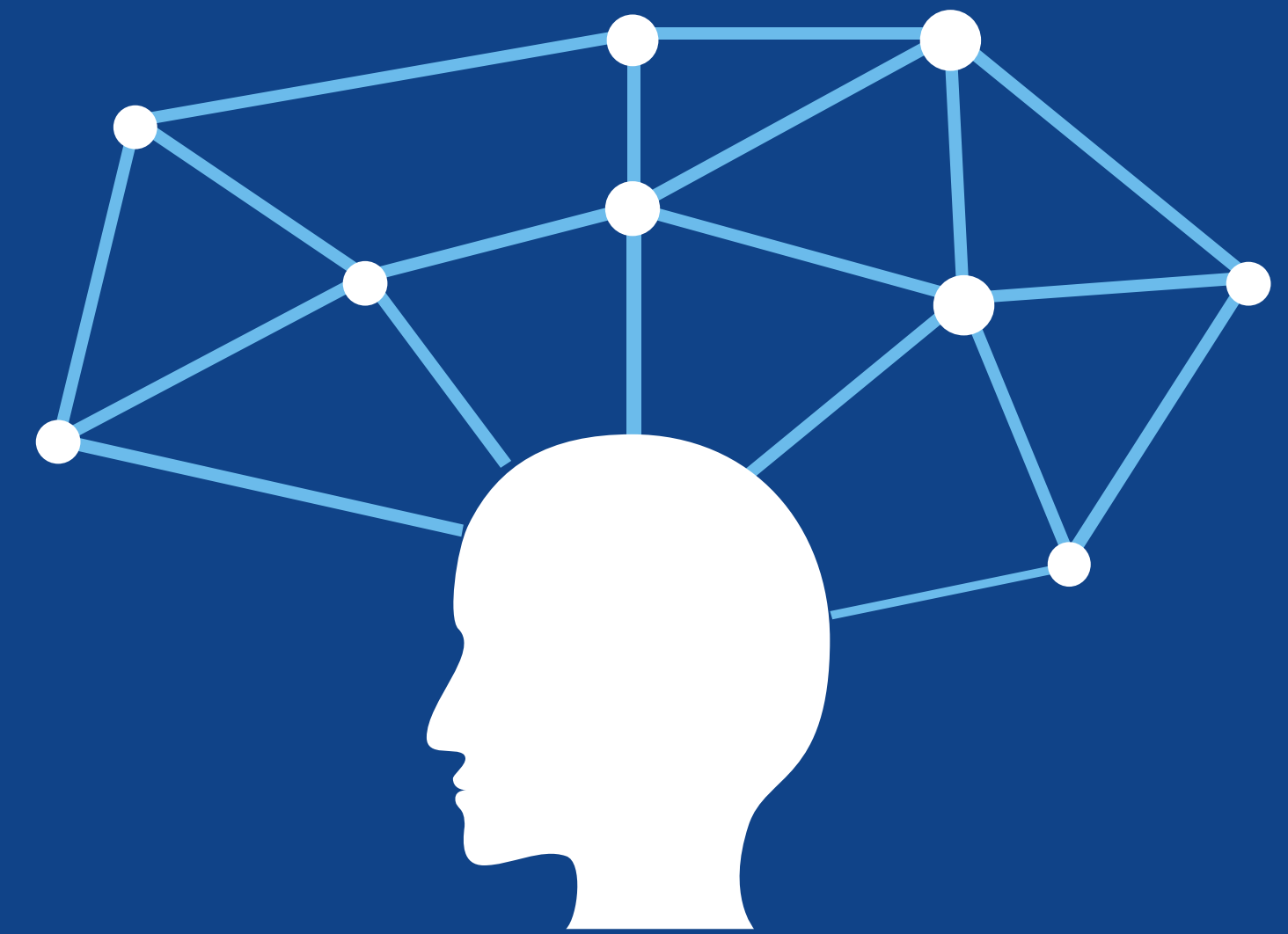
- Análisis y diseño de base de datos (modelo BDD)
- Arquitectura y diseño de redes IOT
- Sensores. actuadores y objetos inteligentes
- Transferencia de datos por IOT



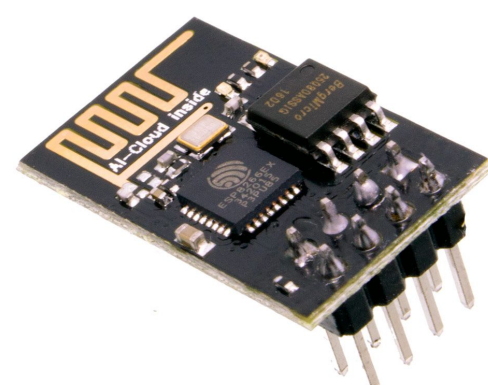
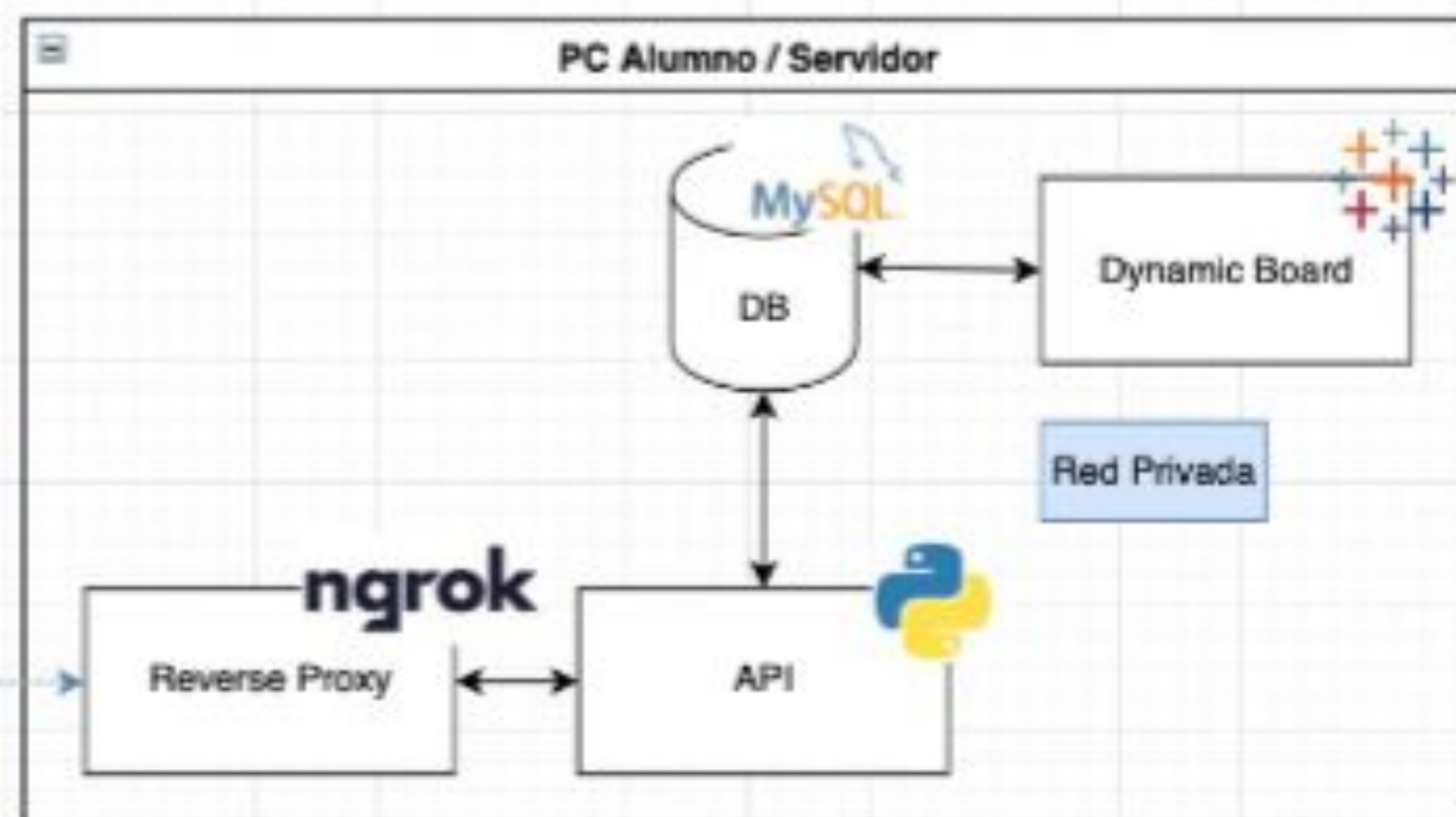
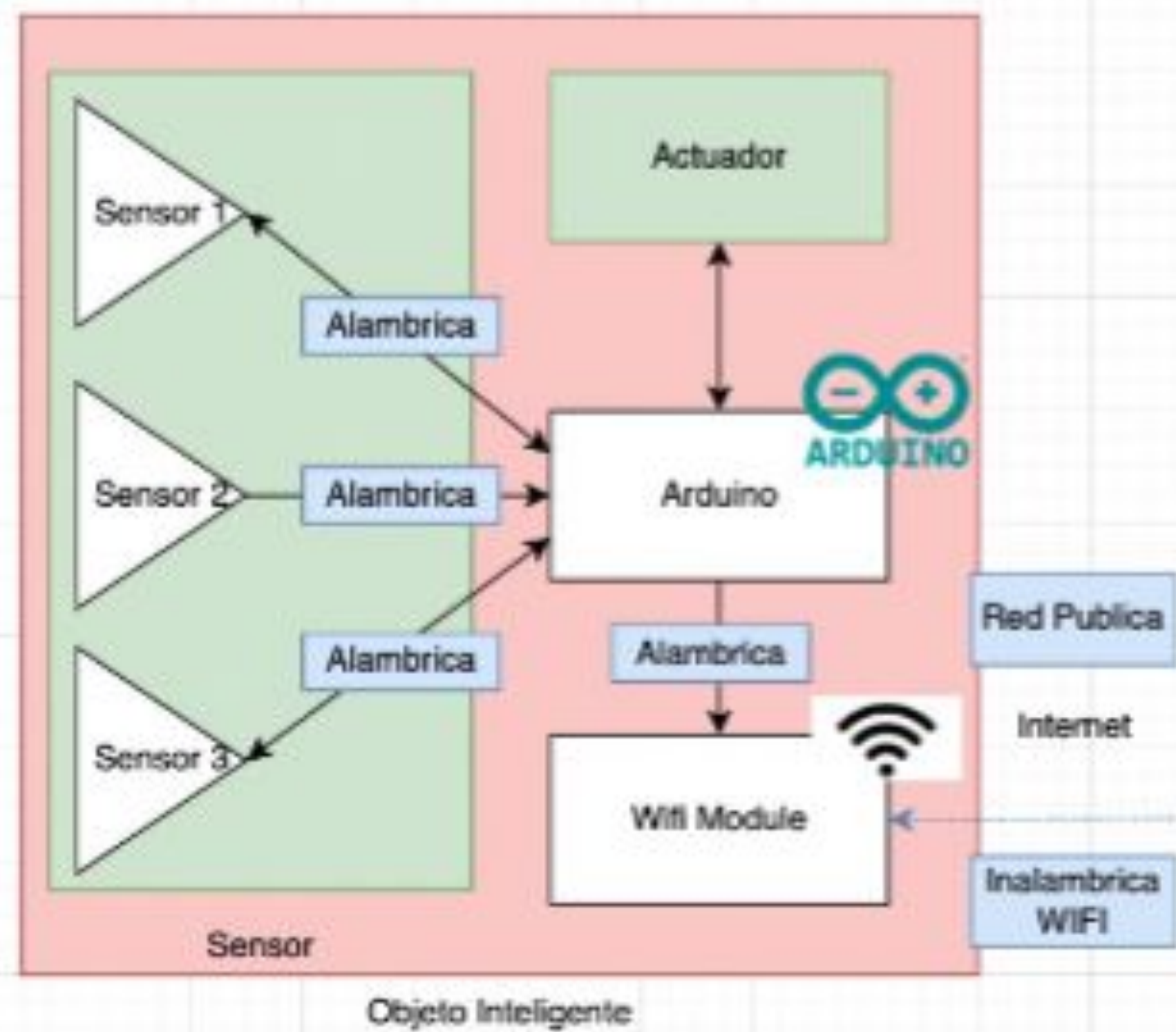
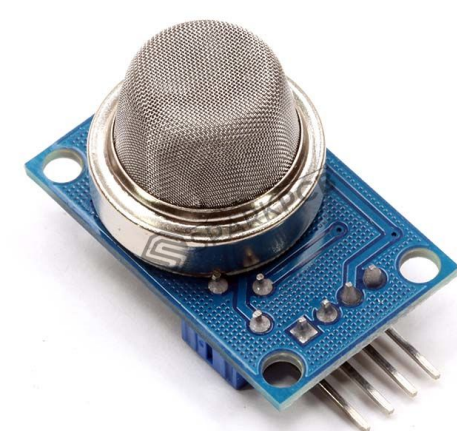
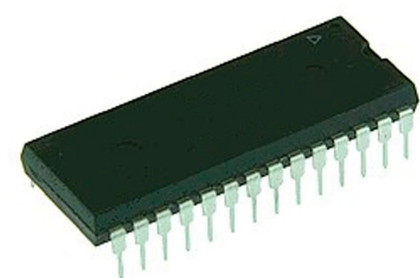


# Hardware

- Sensores MQ 135
- Arduino
- Modulo ESP8266 (wifi)
- Circuito integrado
- Circuitos









# Prototipo físico

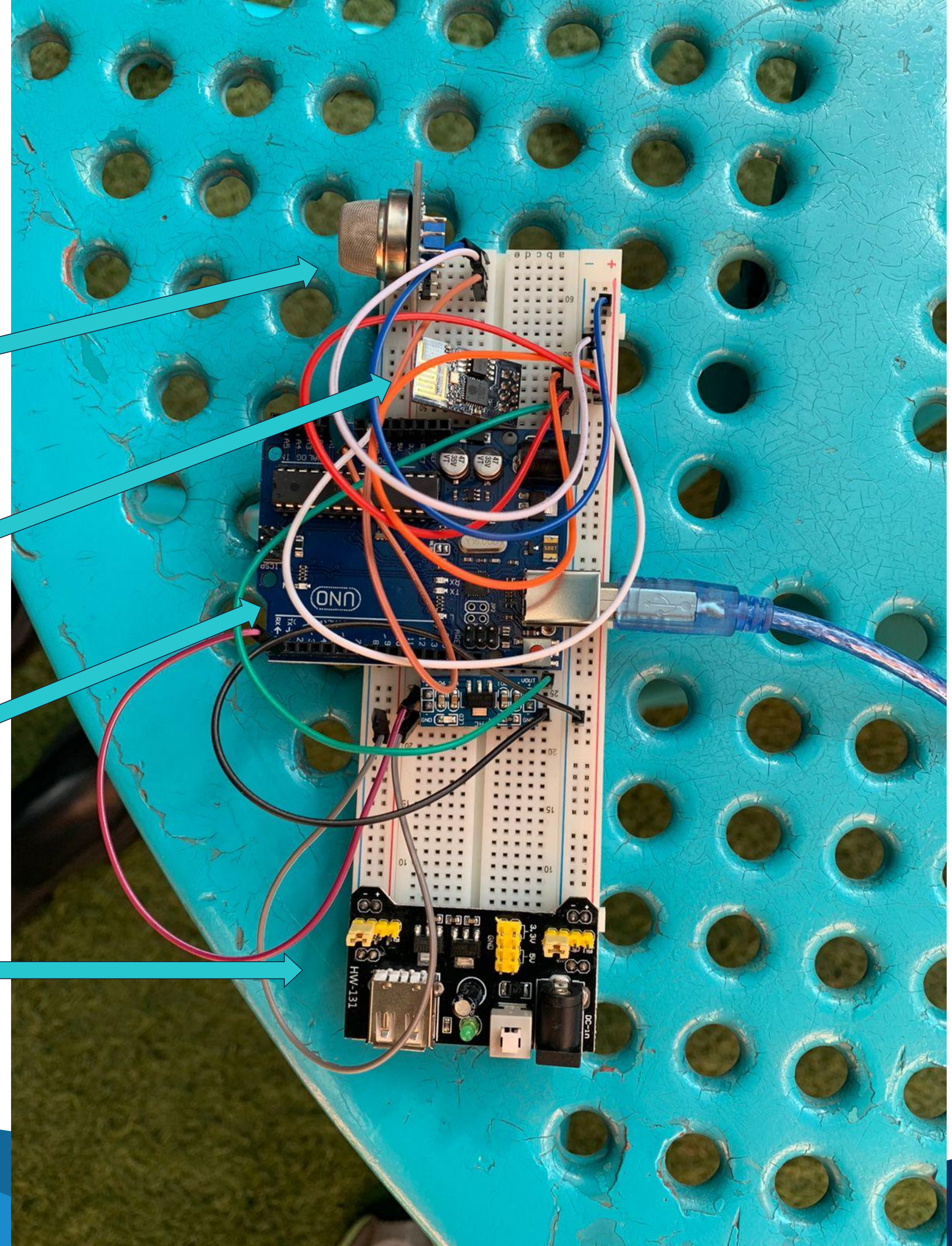
(Componentes principales)

Sensor de aire

Modulo WIFI ESP2866

Arduino UNO

Módulo Fuente De Voltaje Mb102 Para Protoboard





# Código Arduino

```

arduino1
#include <ArduinoHttpClient.h>
#include <ArduinoJson.h>
#include <WiFiNINA.h>

char ssid[] = "SoteloNetwork";
char password[] = "lturrlaga";

int status = WL_IDLE_STATUS;

char server[] = "f466-177-247-99-79.ngrok.io"; //Always modify when re-run ngrok

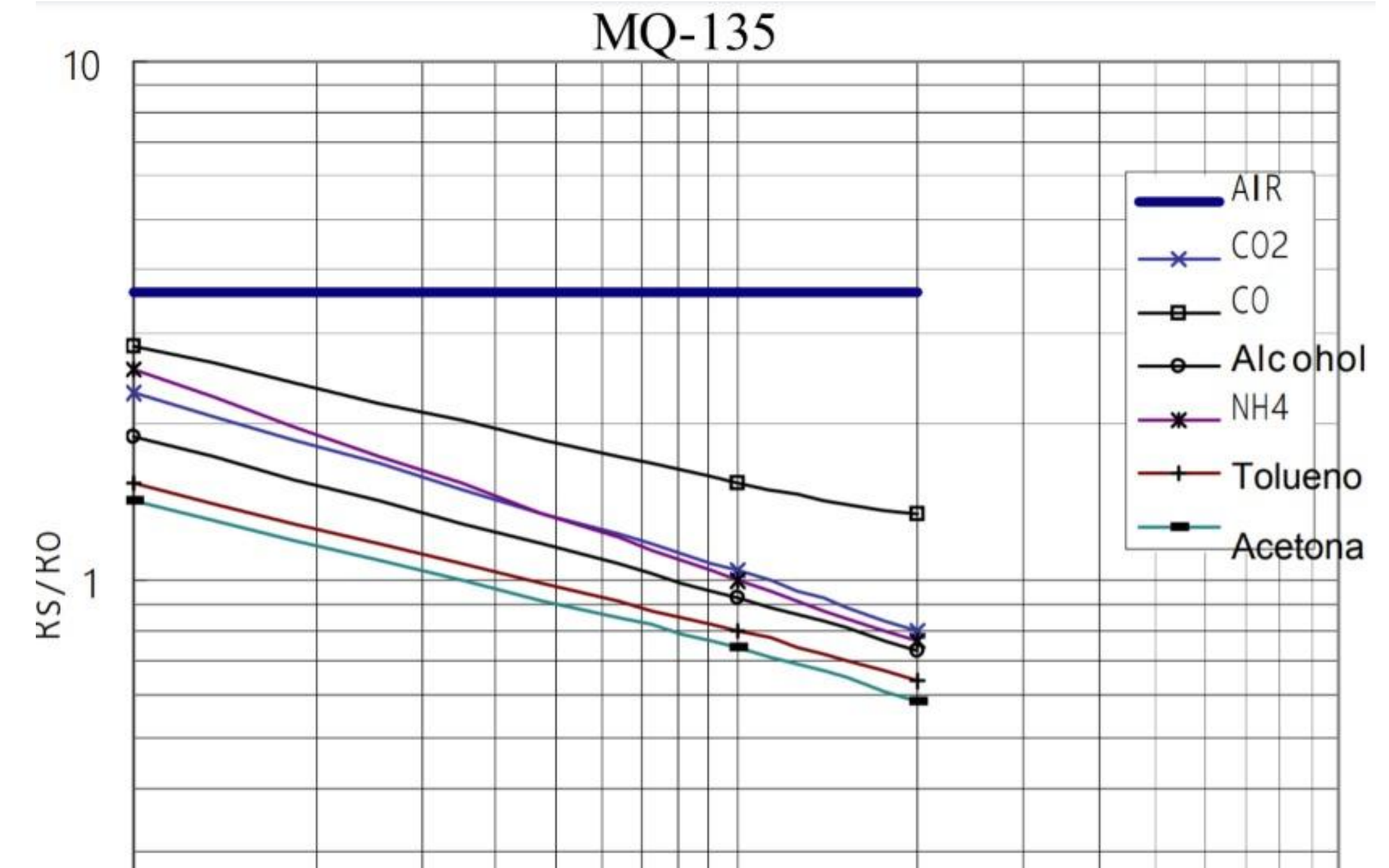
WiFiClient client;

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

  while (status != WL_CONNECTED) {
    Serial.println("Attempting to connect to Network: ");
    Serial.println(ssid);
    status = WiFi.begin(ssid,password);
    delay (1000);
  }

  Serial.print("Connected to SSID: ");
  Serial.println(WiFi.SSID());
  IPAddress ip = WiFi.localIP();
  IPAddress gateway = WiFi.gatewayIP();
  Serial.print("IP Address: ");
  Serial.println(ip);

```



ppm	Rs /Ro
200	0.8
100	1.1
90	1.15
80	1.2
70	1.25
60	1.3
50	1.45
40	1.55
30	1.75
20	1.85
17	2
10	2.3

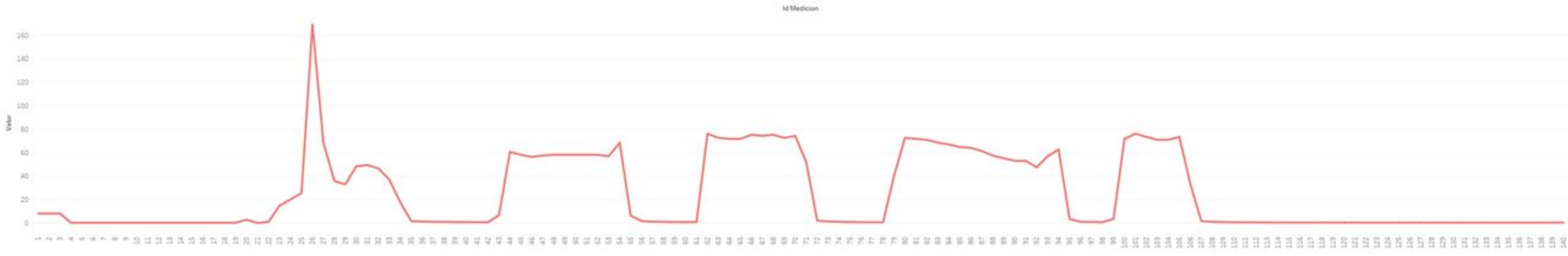
$$y = 105.68x + 219.8$$



# Código API

```
app.py  X
app.py
1  from flask import Flask, jsonify, request
2  from flask_sqlalchemy import SQLAlchemy
3  from flask_expects_json import expects_json
4  from datetime import datetime, timezone
5  import os
6
7  app = Flask(__name__)
8  #Configura el SQLALCHEMY con Flask
9  app.config['SQLALCHEMY_DATABASE_URI'] = 'mysql+mysqlconnector://root:SuperPWD@localhost/IOT'
10 app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
11
12 #Crea objeto SQLALCHEMY
13 db = SQLAlchemy(app)
14
15 # arreglo = [{"id":0, "data":1}, {"id":1,"data":2}, {"id":2,"data":3}]
16 schema_parametros = {
17     'type': 'object',
18     'properties': {
19         'unidades': {'type': 'string'},
20         'alertaAlto': {'type': 'number'},
21         'alertaBajo': {'type': 'number'},
22         'valorMaximo': {'type': 'number'},
23         'valorMinimo': {'type': 'number'},
24         'nombre': {'type': 'string'}
25     },
26     'required': ['unidades', 'alertaAlto', 'alertaBajo', 'valorMaximo', 'valorMinimo', 'nombre']
27 }
28
29
30 schema_dispositivos = {
31     'type': 'object',
32     'properties': {
33         'numSerie': {'type': 'string'}
```

Datos de CO2 capturados

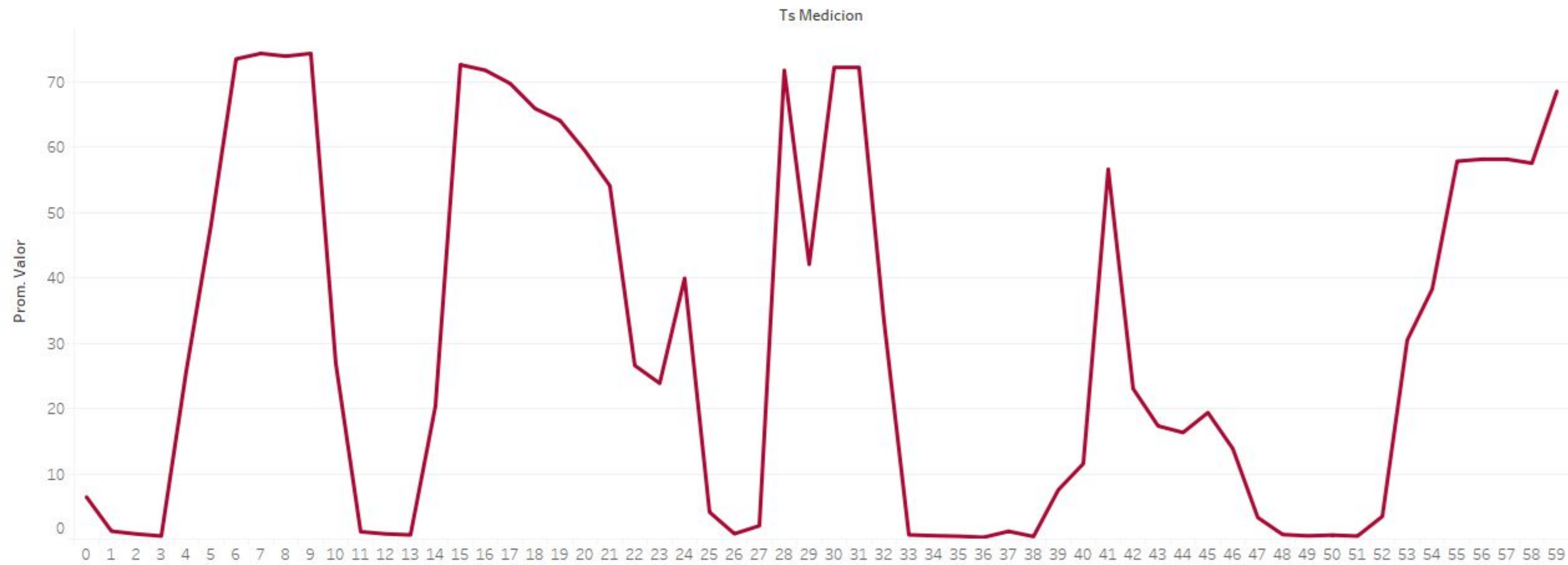


La tendencia de Valor para Id Medicion



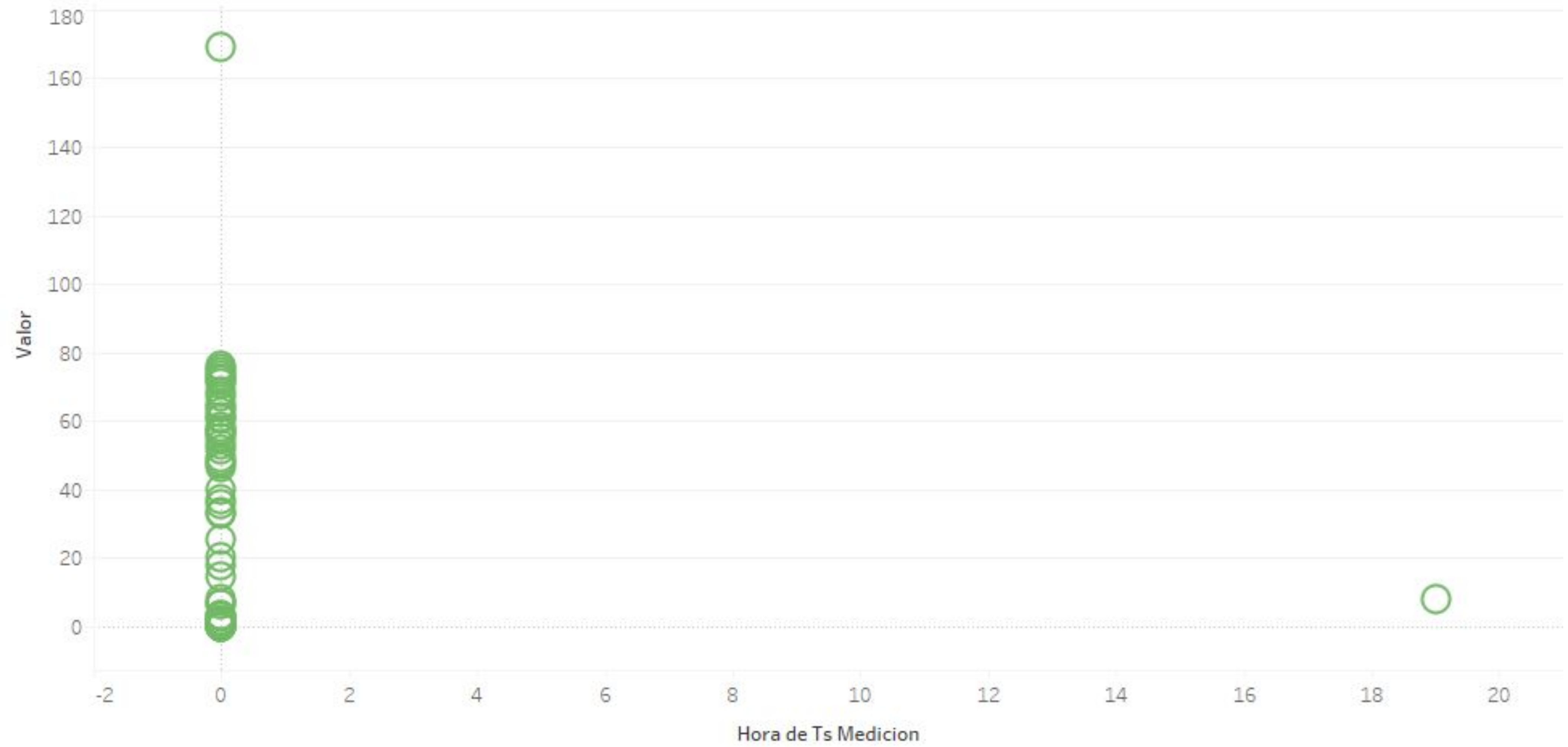


## Niveles de CO2 por segundo



La tendencia de promedio de Valor para Ts Medicion segundo.

## Datos capturados Max y Min



Ts Medicion hora vs. Valor.





# Monitoreo y control

4

Project Charter y RAID's



# Project charter

Objective	Stakeholders			
Calcular la calidad del aire de cierta zona para crear un plan de acción para mejorar o mantener esos datos para alertar sobre la presencia de algún gas nocivo.	Las personas viviendo en el área analizada.			
	Expertos de la calidad de aire.			
	Organizaciones que cuidan al medio ambiente.			
In Scope	Risks / Issues / Challenges / Dependencies			
Obtención de datos de la calidad del aire, datos que serán almacenados y leídos en la red, con los sensores indicados. El tráfico de datos será a través del internet a un Reverse Proxy.	Falta de conocimiento del programa			
	No capture cierta información al momento de pasarlo a la nube			
	Alteración de los datos por causas externas.			
Out of Scope	#	Key Milestones	Due Date	Status
Tener dos sensores al mismo tiempo. Tener una capa protectora Hacer móviles el protoboard		Creación del sistema	Semana 6	●
		Programación del sistema	Semana 6	●
		Captura de datos	24 horas después de haber corrido	●
Assumptions		Almacenamiento en la nube.	Semana 7	●
Condiciones del clima y alrededores adecuados Programa corre como debiera, sin errores y sin pérdida de información (incluyendo los errores de hardware).		Análisis de datos (Tableau)	Semana 7	●
		Corrección de programa	Durantes las pruebas.	●
		Creación de red domótica	Semana 8	●
		Verificación de datos en tiempo real	antes de la entrega	●
Legend: Completed ● On Track ● At Risk ● Delayed ● Not Started ●				



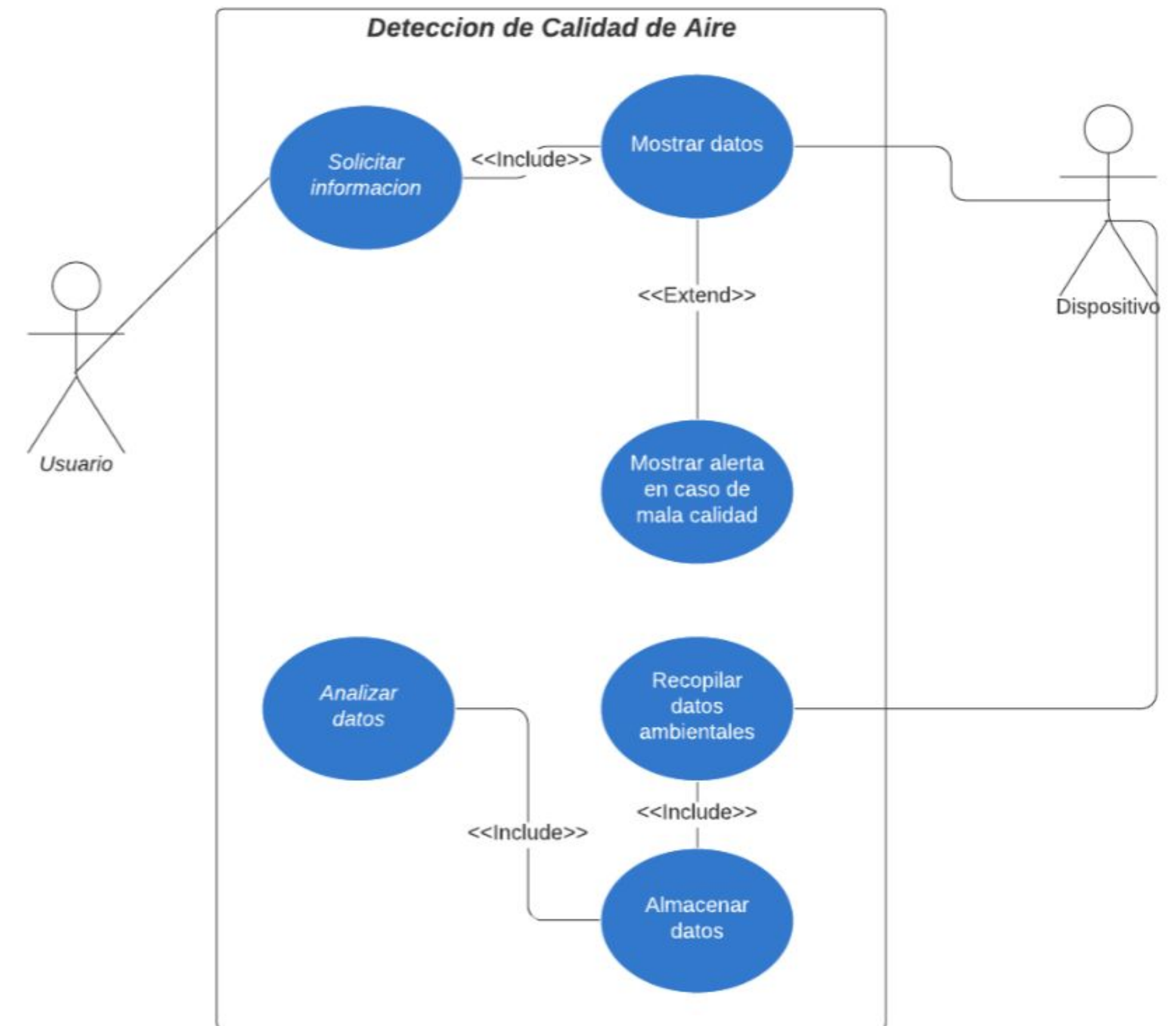
# Project charter

## a) Restricciones del producto

- i) Rapidez: Se tiene planeado que el dispositivo esté activo continuo, pero los sensores pueden tardan en tomar las capturas.
- ii) Facilidad de uso: Que no sea intuitivo.
- iii) Fiabilidad: Problemas añadidos por la saturación de datos y la red que afecten las actualizaciones del servicio y los tiempos de espera.

## b) organizacionales o externos

- i) El tráfico de datos será a través del internet a un Reverse Proxy





# Cierre

Conclusiones

5