MATERIA DE SISTEMAS EMBEBIDOS
PROYECTO 1B

Wilmer D. Farinango-Tallana
Pablo S. Ulcuango-Necpas

16 de junio de 2021

1. Introducción

En el presente documento se presenta evidencia de la utilización de criterios como: la adquisición de datos mediante sensores de gas (MQ153 Y MQ7), en periodos de 5 min con estados de programación SLEEP. Con la adquisición de datos se procede a tener una base de datos con los que se pasaran por filtros, suavizado de señales, convolución y realacion señal-suido (SNR), mediante la codificación en el entorno de arduino. Dicha base de datos también será utilizada para el entrenamiento, aprendizaje de maquinas, introduciendo los criterios de algoritmos de aprendizaje como KNN y Bayes.
Todo el proceso mencionado se reflejará como resultado final en una interfaz diseñada en el software Processing, en donde, se presentan los datos conseguidos por los sensores y evaluando si dichos datos son de un entorno de aire limpio o un aire contaminado.

2. RESUMEN

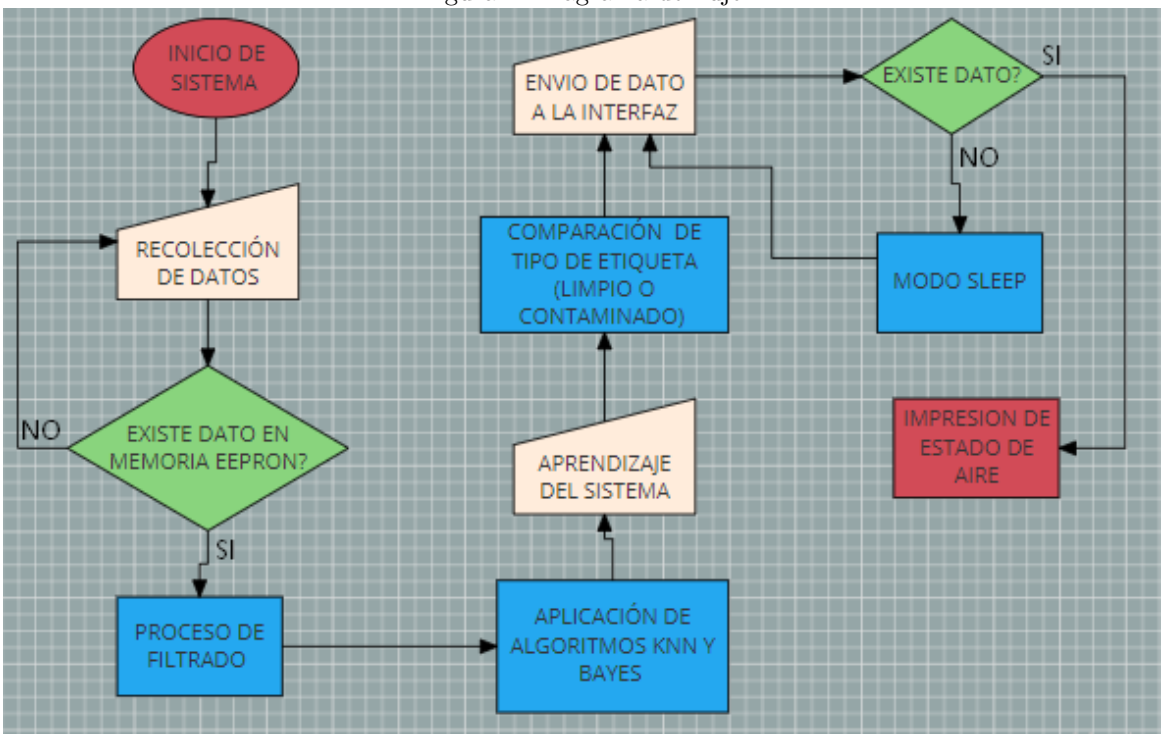
La contaminación del aire es una gran Riesgo para la salud, además de resultar en un gran daño económico. A En la actualidad, la contaminación del aire se controla principalmente con alta precisión. Equipos, pero a una resolución espacial extremadamente baja. Superando la desventaja de la baja resoluci´on espacial, y la construcción de una red inalámbrica de sensores, puede abrir puertas a nuevas aplicaciones, tanto para comprender mejor las fuentes y los efectos sobre la salud de la contaminación del aire. Revisamos proyectos de vanguardia seleccionados para suministrar al lector con una visión general de las aplicaciones actuales que fueron habilitadas por el uso de nuevos sensores, redes de sensores inalámbricos (WSNW) y tecnologías de la información. Evaluaremos críticamente lo espacial y resolución temporal, la precisión del sensor y los resultados. Mostraremos que el paso crucial, aún por hacer, es el desarrollo de sensores fiables. Esto puede habilitar el avance de las aplicaciones existentes. Al introducir inteligente nodos, las capacidades de la Internet de las cosas (IoT) pueden desencadenar nuevos casos de uso. Concluiremos con una visión de futuro. Trabajo, presentando nuevas aplicaciones para proteger mejor a los ciudadanos de los efectos adversos para la salud de la contaminación del aire y para mejorar entender esos efectos. Mostraremos que estas aplicaciones. Requiere sensores confiables y WSNWs. La contaminación del aire es una gran Riesgo para la salud, además de resultar en un gran daño económico. A En la actualidad, la contaminación del aire se controla principalmente con alta precisión. Equipos, pero a una resolución espacial extremadamente baja. Superando la desventaja de la baja resolución espacial, y la construcción de una red inalámbrica de sensores, puede abrir puertas a nuevas aplicaciones, tanto para comprender mejor las fuentes y los efectos sobre la salud de la contaminación del aire. Revisamos proyectos de vanguardia seleccionados para suministrar al lector con una visión general de las aplicaciones actuales que fueron habilitadas por el uso de nuevos sensores, redes de sensores inalómbricos (WSNW) y tecnologías de la información. Evaluaremos críticamente lo espacial y resolución temporal, la precisión del sensor y los resultados. Mostraremos que el paso crucial, aún por hacer, es el desarrollo de sensores fiables. Esto puede habilitar el avance de las aplicaciones existentes. Al introducir inteligente nodos, las capacidades de la Internet de las cosas (IoT) pueden desencadenar nuevos casos de uso. Concluiremos con una visión de futuro. Trabajo, presentando nuevas aplicaciones para proteger mejor a los ciudadanos de los efectos adversos para la salud de la contaminación del aire y para mejorar entender esos efectos. Mostraremos que estas aplicaciones. Requiere sensores confiables y WSNWs.

3. Diseño del Sistema

3.1. Diagrama de Flujo

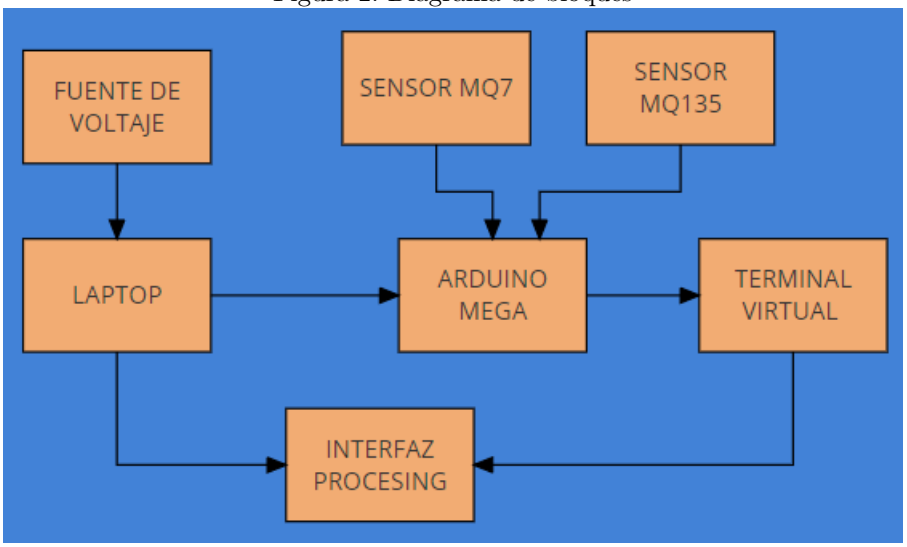
Ingresar su diagrama de flujo realizado en cualquier programa.

Figura 1: Diagrama de flujo



3.2. Ingrese su diagrama de bloques

Figura 2: Diagrama de bloques



4. Desarrollo

El sistemas esta basado en cinco aspectos principales:La adquisicion de datos mediante sensores de detección gases denomidados NOx MQ135 y CO MQ7, por consiguiente, dichos datos serán almacenado en un Arduino

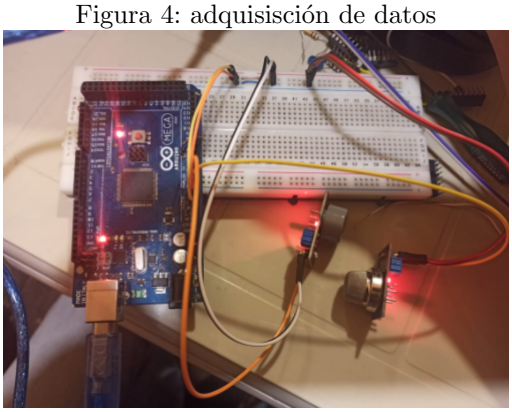
MEGA. El segundo proceso es el tratamiento de los datos adquiridos utilizando el criterio de filtrado de señales, en este caso se utiliza el filtro Gausiano y el filtro de mediana. EL tercer aspecto se encarga de el aprendizaje de maquinas con la utilización de algoritmos de clasificación los cuales son KNN y Bayes, por consiguiente, cada dato que se ingrese y ya sea tatado mediante filtro los algoritmos se encargaran de almacenarlos y evaluarlos para determinar si pertenecen a la etiqueta 1 o 2. El cuarto aspecto se viene enfocado a al diseño de la interfaz en el entorno del software de programación Processing, en el cual se mostrará el valor de adquisición de los datos de cada sensor y el tipo de aire (Limpio o Contaminado) que representan dichos datos. Y el último pero no menos importante es el diseño de el hardware del sistema, diseñado en la herramienta de fUSION 360, en donde se crea la carcasa para introducir el arduino, los sensores ed forma que el proyecto tenga un terminado más estético.

4.1. Adquisición de Datos

Para realizar la adquisisción de datos se realiza el armado del sistema electrónico el que está contituido por los sensores ya mencionados conectados al arduino y los mismos conectados a los pines análogos del Arduino MEGA. Una vez realizado el circuito se procede a recolectar datos en ambientes de aire limpio y aire contaminado. Esto se realiza mediante código de arduino con condiciones de que se tome 100 instancias en rangos de 5 minutos cada uno, mientras pase el periodo de tiempo se configura al arduino que pase a modo SLEEP.

Figura 3: circuito del sistema

Figura 4: adquisisción de datos



4.2. Filtrado de Datos

4.2.1. Codigo principal

```
/*
   MQ-7 con filtro de Gauuss
*/
float myArray[] = {336, 336, 336, 336, 336, 336, 336, 336, 334, 334, 334, 334, 334, 334,
    334, 334, 334, 334, 334, 334, 333, 334, 334, 333, 333, 333, 333, 333, 334, 334, 333,
    333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333,
    333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333,
    333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333,
    333, 333, 333, 333, 333, 333, 333, 333, 333, 332, 332, 333, 333, 332, 333, 332, 333,
```

```
    332, 361, 361, 361, 361, 361, 361, 361, 361, 361, 361, 361, 361, 360, 360, 360,
    360, 360, 360, 360, 360, 360, 360, 360, 360, 359, 359, 359, 359, 359, 359, 358,
    358, 358, 358, 358, 357, 357, 356, 357, 357, 357, 357, 357, 357, 356, 356, 356, 356,
    356, 355, 355, 355, 355, 355, 355, 354, 354, 354, 354, 353, 353, 353, 353, 352, 352,
    352, 352, 352, 352, 351, 351, 351, 351, 351, 351, 351, 350, 350, 350, 350, 350, 350,
    350, 350, 350, 349, 349, 349, 349, 349, 349, 349, 349, 349, 349, 348, 348, 348,};
float kernel[] = {0.05, 0.24, 0.4, 0.24, 0.05};
float G[] = {336, 336, 336, 336, 336, 336, 336, 336, 334, 334, 334, 334, 334, 334, 334,
    334, 334, 334, 334, 334, 333, 334, 334, 333, 333, 333, 333, 333, 334, 334, 333, 333,
    333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333,
    333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333,
    333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333, 333,
    333, 333, 333, 333, 333, 333, 333, 333, 332, 332, 333, 333, 332, 333, 332, 333, 332,
    361, 361, 361, 361, 361, 361, 361, 361, 361, 361, 361, 361, 361, 360, 360, 360, 360,
    360, 360, 360, 360, 360, 360, 360, 360, 360, 359, 359, 359, 359, 359, 359, 358, 358,
    358, 358, 358, 357, 357, 356, 357, 357, 357, 357, 357, 357, 356, 356, 356, 356, 356,
    355, 355, 355, 355, 355, 355, 354, 354, 354, 354, 353, 353, 353, 353, 352, 352, 352,
    352, 352, 352, 351, 351, 351, 351, 351, 351, 351, 350, 350, 350, 350, 350, 350, 350,
    350, 350, 349, 349, 349, 349, 349, 349, 349, 349, 349, 349, 348, 348, 348,};
int N = 200, Nk = 5;
float den = 0, c = 0, zj = 0;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  den = 0;
  for (int i = 0; i < Nk; i++) {
    den = den + kernel[i];
  }
  for (int i = 0; i < N - Nk; i++) {
    for (int i2 = 0; i2 < Nk; i2++) {
      c = c + kernel[i2] * myArray[i2 + i];
    }
    c = c / den;
    Serial.print(c);
    Serial.print(",");
    Serial.println(myArray[i + 2] + 70);
    c = 0;
  }
  SNR();
}
void SNR() {
  //Promedio se al original v1
  float sum = 0;
  for (int i = 0; i < 90; i++) {
    sum = sum + G[i];
  }
  float promedio = sum / 90;
  float v1 = ((promedio * 5) / 1023);
  Serial.println(v1);
  Serial.print(",");
  //Promedio se al Suavizada (v2) metodo de Gauss
  float sum2 = 0;
  for (int i = 0; i < 90; i++) {
    sum2 = sum2 + G[i + 2];
  }
  float promedio2 = sum2 / 90;
  float v2 = ((promedio2 * 5) / 1023);
  Serial.println(v2);
  Serial.print(",");
  //Metrica SNR m=20*log(v2/v1)
  float metrica = 20 * log(v2 / v1);
  Serial.println(metrica);
  Serial.print(",");
}
```

4.3. Algoritmos de clasificación

```
/*
  *Knn medici n de datos en tiempo real
 */
#include "datos.h"
#include <avr/power.h>
```

```
#include <avr/sleep.h>
#include <avr/wdt.h>
#include <MsTimer2.h>

int respuesta;
int knn (int k, int etiquetas, int tam_col, int col_fil);
int cont=0;
int MQ135 = 0;
int MQ7 = 1;
int on=0;
int tiempo=10;
int cont1=0;
int j=0;
int k=0;
int envio=0;
int aux=0;

void setup() {
  Serial.begin(9600);
  MsTimer2::set(1000,reloj);
  set_sleep_mode(SLEEP_MODE_STANDBY);
  sleep_enable();
  Serial.println("INICIO DEL SISTEMA");
  Serial.println("#  MQ135  MQ7     Tipo de Aire");
}

void loop() {
  MQ135 = analogRead(0);
  MQ7 = analogRead(1);
  delay(1000);
  respuesta=knn(3,2,3,160);
  switch(respuesta){
    case 1:
      cont++;
      for(;k<1;k++){
        Serial.println(String(cont)+String("    ")+String(MQ135)+String("    ")+String(MQ7)+
            String("    ")+String("Aire Contaminado"));
        delay(250);
        MQ135=Serial.read();
        Serial.write(MQ135);
        MsTimer2::start();
        tiempo=0;
        on=0;
      }
    break;
    case 2:
      cont++;
      for(;k<1;k++){
        Serial.println(String(cont)+String("    ")+String(MQ135)+String("    ")+String(MQ7)+
            String("    ")+String("Aire Limpio"));
        delay(250);
        MQ135=Serial.read();
        Serial.write(MQ135);
        MsTimer2::start();
        tiempo=0;
        on=0;
      }
    break;
    }
}

int knn (int k, int etiquetas, int tam_col, int tam_fil){
int col;
int fil=0;
int i=0;
int j;
float aux;
float aux_etiqueta;
float potencia;
float raiz;
int label;
String salida="";
float datos_prueba [3]={MQ135,MQ7,2.0};
 float matriz_k [3][k];
 for(;i<k;i++){
  matriz_k[0][i]=i+1.00;
```

```
  matriz_k[1][i]=0;
  matriz_k[2][i]=2500.0+i;
  }
  i=0;
  float matriz_eti[2][etiquetas];
  for(;i<etiquetas;i++){
      matriz_eti[0][i]=i+1.0;
      matriz_eti[1][i]=0.0;
    }
   for(;fil<tam_fil;fil++){
      for(col=0;col<tam_col-1;col++){
          potencia=potencia+pow(matriz[fil][col]-datos_prueba[col],2);
        }
        raiz=sqrt(potencia);
        potencia=0;
        if(raiz<matriz_k[2][k-1]){
          matriz_k[2][k-1]=raiz;
          matriz_k[1][k-1]=matriz[fil][tam_col-1];
          for(i=0;i<k;i++){
             for(j=i+1;j<k;j++){
               if(matriz_k[2][i]>matriz_k[2][j]){
                  aux=matriz_k[2][i];
                  matriz_k[2][i]=matriz_k[2][j];
                  matriz_k[2][j]=aux;
                  aux_etiqueta=matriz_k[1][i];
                  matriz_k[1][i]=matriz_k[1][j];
                  matriz_k[1][j]=aux_etiqueta;
                }
              }
            }
        }
    }
    for(i=0;i<etiquetas;i++){
      for(j=0;j<k;j++){
          if(matriz_eti[0][i]==matriz_k[1][j]){
            matriz_eti[1][i]++;
          }
        }
      }
    for(i=0;i<etiquetas-1;i++){
       if(matriz_eti[1][i]<matriz_eti[1][i+1])
        label=(int)matriz_eti[0][i+1];
        else
        label=(int)matriz_eti[0][i];
      }
 return label;
};

void reloj() {
  cont1++;
  power_adc_disable();
  if(cont1==tiempo){
    power_adc_enable();
    cont1=0;
  }
  if(cont1>=10){
    wdt_enable(WDTO_8S);
    sleep_mode();
  }
}
```

4.4. Interfas (Processing)

```
/*
 Interfaz grafica - PROYECTO FINAL
 integrantes:
 Ulcualngo Pablo
 Farinango Wilmer
 */


import processing.serial.*;
Serial port;
int dato;
int dato1;
```

```
int n=0;
int dato3=0;
PImage img;
PImage img2;
PImage img3;

void setup () {
  size(720, 640);
  background(255);
  port= new Serial(this, "COM5", 9600);
  img = loadImage("limpio.jpg");
  img2 = loadImage("Contaminado.png");
  img3 = loadImage("fondo.png");
  port.bufferUntil('\n');
}

void draw() {
  fill(255);
  strokeWeight(3);
  img3.resize(640, 640);
  image(img3, 40, 6);

  //cuadro titulo
  stroke (250, 1, 1); //color borde rojo
  strokeWeight (0); //grosor 5
  fill(#E9502B);// color relleno
  rect (50, 20, 620, 130, 15);// rectangulo redondeado

  // cuando principal
  stroke (25); //color borde rojo
  strokeWeight (1); //grosor 5
  fill(#15B09F);// color relleno
  rect(50, 200, 630, 200, 10);// rectangulo redondeado

  textSize(25);
  fill(255);
  text("PROYECTO FINAL", 150, 60);
  textSize(20);
  fill(0);
  text("Integrantes: ", 85, 85);
  text("- Farinango Wilmer", 85, 105);
  text("- Ulcuango Pablo", 85, 125);
  // cuadro de texto mq135
  fill(255);
  strokeWeight(0);
  rect(70, 260, 180, 80, 10);
  // cuadro de texto mq7
  fill(255);
  strokeWeight(0);
  rect(275, 260, 180, 80, 10);
  // cuadro de texto tipo de aire
  fill(255);
  strokeWeight(0);
  rect(480, 260, 180, 80, 10);

  textSize(18);
  fill(255);
  text("SENSOR MQ 135", 90, 240);

  textSize(18);
  fill(255);
  text("SENSOR MQ 7", 300, 240);

  textSize(18);
  fill(255);
  text("TIPO DE AIRE", 500, 240);

  textSize(18);
  fill(0);
  text(dato, 130, 310);
  text(dato1, 340, 310);

  if (dato3==2) {
    text("Aire limpio", 490, 310);
    img.resize(450, 200);
    image(img, 140, 420);
```

```
  } else if (dato3==1) {
    text("Aire contaminado", 490, 310);
    img2.resize(450, 200);
    image(img2, 140, 420);
  }
}

void serialEvent(Serial port) {
  dato=port.read();
  if (dato<70) {
    dato1=dato*(333/65);
    dato3=2;
  } else {
    dato1=dato*(356/79);
    dato3=1;
  }
}
```

4.5. Diseño de Hardware 3D (FUSION 360)

4.6. Simulación

Figura 5: Simulacion en FUSION 360

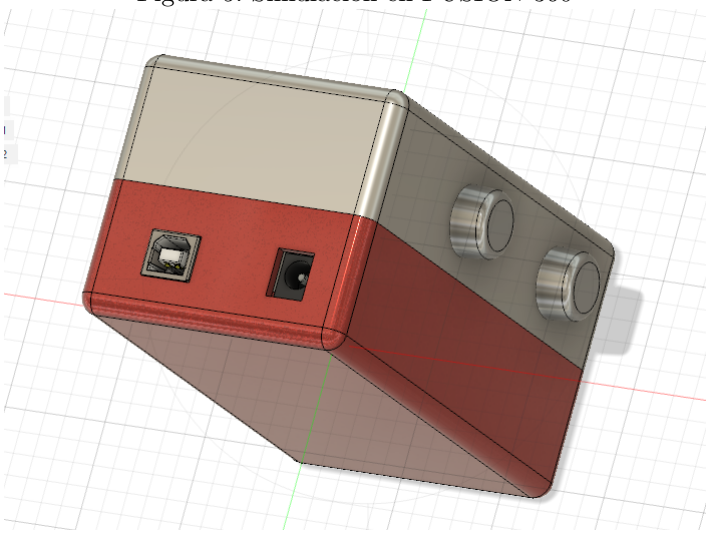


Figura 6: Interfaz gráfica

5. Análisis de Resultados

Figura 7: MQ 135 Filtro Gauss

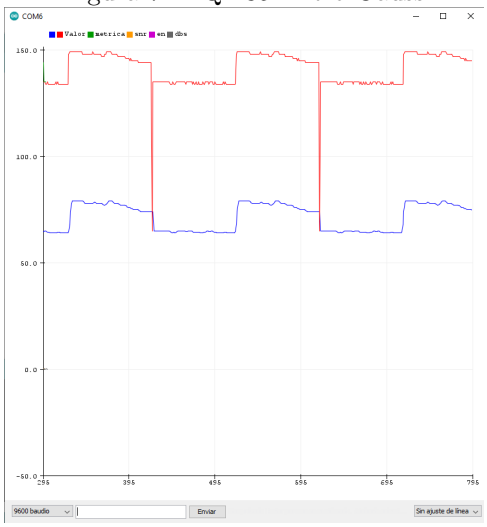


Figura 8: MQ 135 Filtro GaussRuido

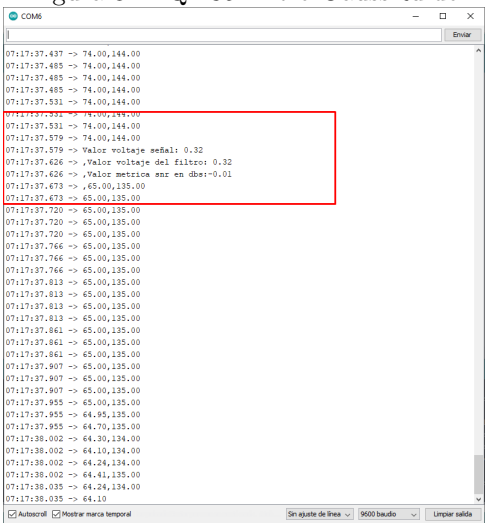


Figura 9: MQ 135 Filtro Mediana

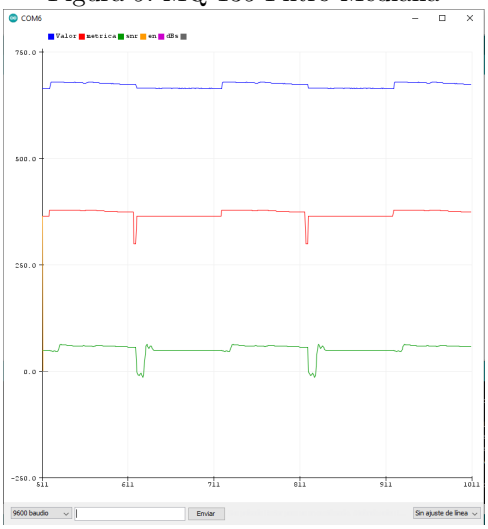


Figura 10: MQ 135 Filtro MedianaRuido

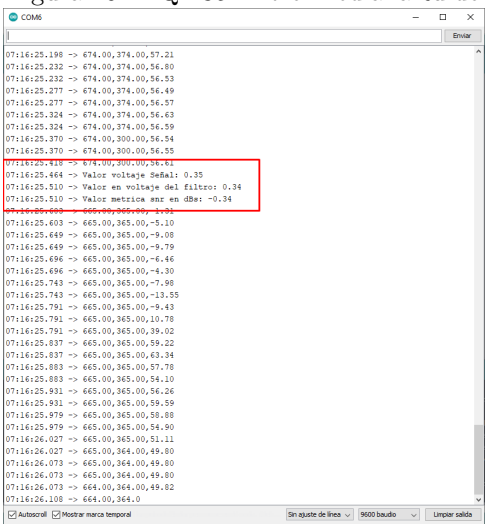


Figura 11: MQ 7 Filtro Gauss

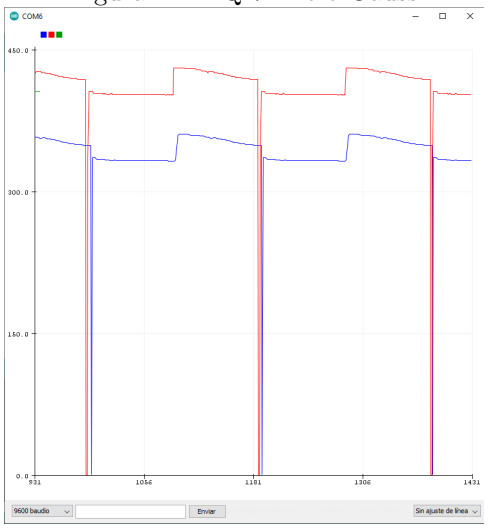


Figura 12: MQ 7 Filtro GaussRuido

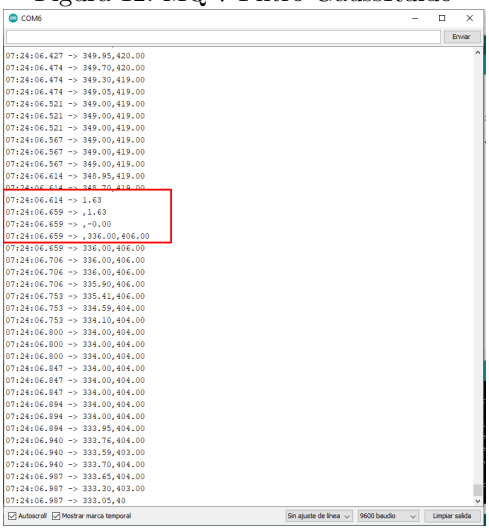


Figura 13: MQ 7 Filtro Mediana

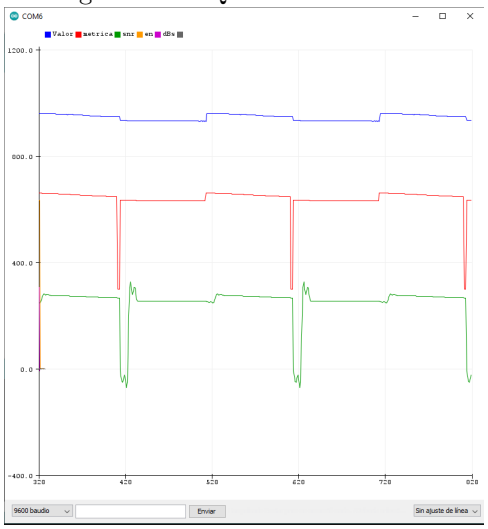
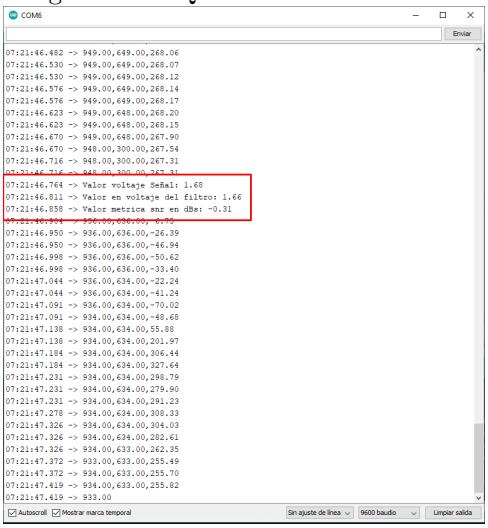


Figura 14: MQ7 Filtro MedianaRuido



6. Conclusiones y Recomendaciones

- El aprendizaje de maquinas depende fundamentalmente de la adquisición de datos para generar o construir un base de datos de donde pueda acceder a información para analizar y clasificar determinadas instancias, en este caso el análisis del tipo de aire si es contaminado o limpio. entre mayor se la cantidad de datos mejor será la clasificación.
- La selección de filtros para el proceso del proyecto fue mediante la comparación del filtro de Gauss y el filtro Mediana. El mas conveniente fue el filtro de Gauss debido a su parámeto de señal a ruido es muy baja, teniendo una relación de 0.01 a diferencia de el filtro mediana que tiene 0.032.
- La matriz de confusión está diseñada de (100,3) ya que son 10 instancias(datos) que se requieren tomar y 2 de las tres columnas son de los datos de los sensores y la ultima de las colimnas es para la asignación del as etiquetas. Esta matriz es la encargada de que mediante los datos de entrenamiento realice el análisis de datos de pruebas y presentará los resultados en la interfaz de procesing.
- Para que la toma de datos mediante los sensores sea optima se debe investigar primero el punto de funcionamiento óptimo de los disositivos y el rango que cubren para detectar los gases, que tienen a proxmiadamente un rango de detección de 10500 ppm . Como la adquisición de datos se los realizó dentro de un domicilio el rango de los componentes no influyo en mayor cantidad.
- Tomar precauciones en el caso de que la adquisición de datos sea extensa, debido que, si se propasa más de 48 horas aquiriendo datos el sensor sufirá sobrecalentamiento.
- Una vez adquirido los datos es recomendable almacenarlos en una hoja de excel, para que se pueda tener de forma ordenada y se pueda realizar operaciones de forma sintetisada y rápida por la extensa cantidad de datos.