MATERIA DE SISTEMAS EMBEBIDOS
LABORATORIO INTERFACES Y APRENDIZAJE DE MÁQUINA

Viviana Y. Yucailla-Muzo

19 de enero de 2021

1. Introducción

En los últimos años, el desarrollo de la web semántica y de las interfaces avanzadas para la visualización de datos ha experimentado un gran auge, en paralelo al aumento de información que de forma exponencial ha ido poblando la web. Así, a cada momento, aparecen aplicaciones o interfaces que nos ayudan a buscar, encontrar, visualizar, procesar y comprender esa información y de igual manera crear éstas interfaces a conveniencia del ususario.
A partir de los datos obtenidos se realizará la regresión lineal para hacer una demostración práctica de la sencillez para implementar mejoras basándose en lenguajes de programación abiertos como en este caso es Processing y Arduino.

La regresión lineal es útil cuando se tiene dos variables relacionadas linealmente o cuya relación se puede aproximar a la ecuación de una línea. La forma habitual de verlo es como una nube de puntos donde se calcula una recta que minimiza la distancia a todos los puntos, en este caso en una interfaz realizada en el software Processing.
El mayor problema que hay en el caso de Arduino es que no hay memoria para guardar todas las muestras así que se encontró la solución calculando los valores estadísticos sin guardar todo el histórico de datos.
También como se requiere se pudo calcular la correlación entre variables para obtener una tendencia lineal.

2. Diseño del Sistema

2.1. Diagrama de Flujo

Figura 1: Diagrama de flujo

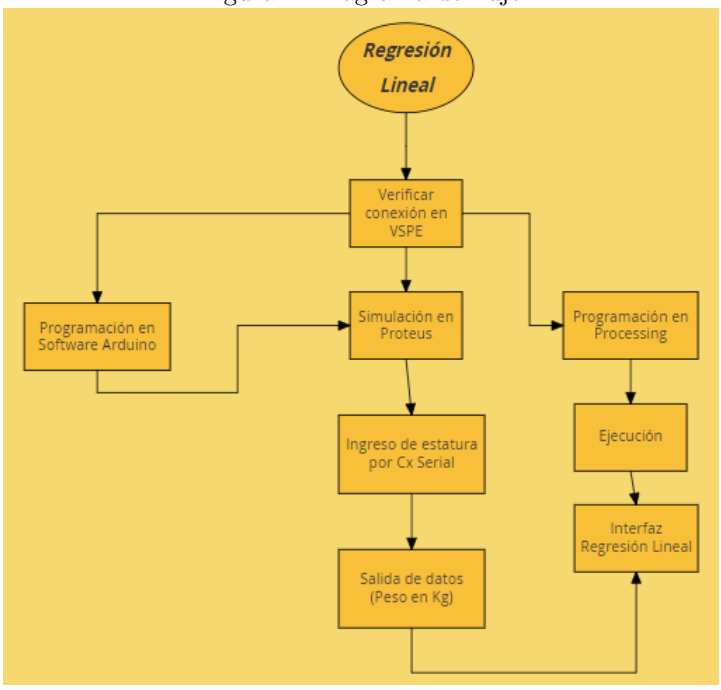
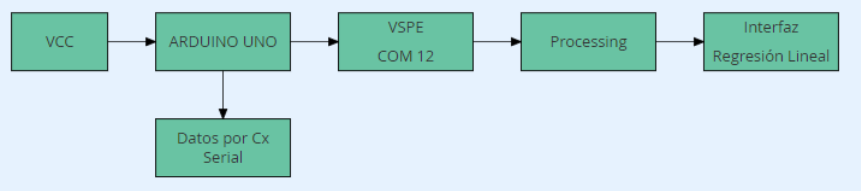


Figura 2: Diagrama de bloques



3. Desarrollo

3.1. Simulación

Figura 3: Simulacion en proteus

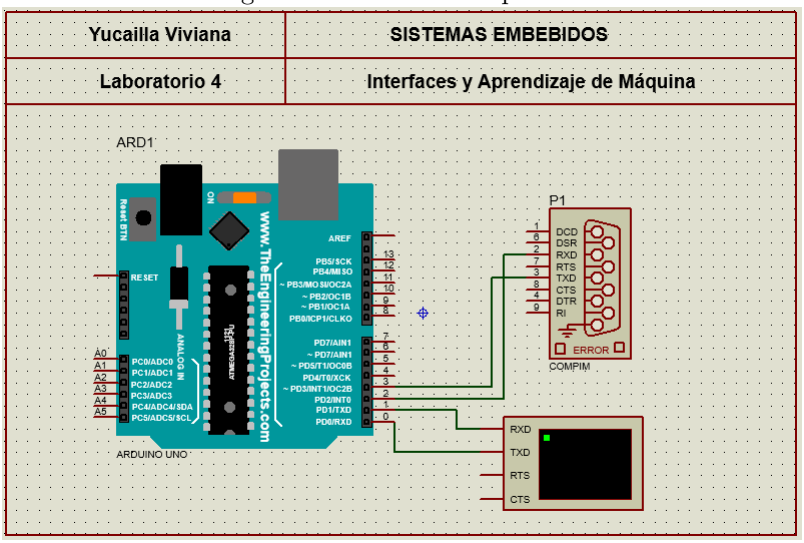
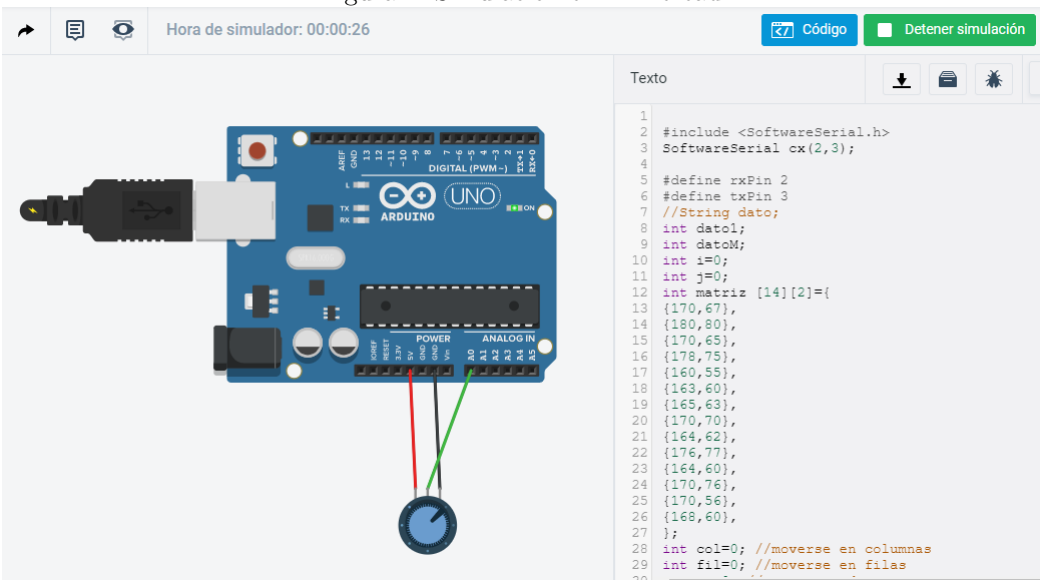


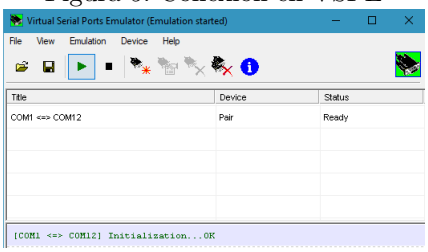
Figura 4: Simulación en Tinkercad



4. Análisis de Resultados

Primero se realizó la conexión en el Virtual Serial Ports Emulator para que no se genere errores en la simulación:

Figura 5: Conexión en VSPE



Se realizó el código en Arduino en donde primero se creó la matriz de los datos con los que se va a trabajar:

Figura 6: Código Arduino matriz

YucaillaV_Lab4_ARDUINO

```
#include <SoftwareSerial.h>
SoftwareSerial cx(2,3);

#define rxPin 2
#define txPin 3
//String dato;
int dato1;
int datoM;
int i=0;
int j=0;
int matriz [14][2]={
{170,67},
{180,80},
{170,65},
{178,75},
{160,55},
{163,60},
{165,63},
{170,70},
{164,62},
{176,77},
{164,60},
{170,76},
{170,56},
{168,60},
};
int col=0; //moverse en columnas
int fil=0; //moverse en filas
int Ex=0; //Sumatoria de x
int Ey=0; //Sumatoria de y
long int Exy=0; //Sumatoria de xy
```

De igual manera se declara las variables:

Figura 7: Código Arduino variables

```
int col=0; //moverse en columnas
int fil=0; //moverse en filas
int Ex=0; //Sumatoria de x
int Ey=0; //Sumatoria de y
long int Exy=0; //Sumatoria de xy
long int Ex2=0; //Sumatoria de x^2
long int Ex_2=0; //Sumatoria de (Ex)^2
int n=14; //tam de muestras
float Bo;//ordenada en el origen
float m; //pendiente
String dato; //recibir estatura
int estatura; //convertir dato
int edad;
float peso;
int ml; //Aux
```

Se prosigue con la función setup en donde se crea el modelo con el cual se va a trabajar en la simulación y processing, también se realiza el ingreso de datos:

Figura 8: Función Setup en Arduino

```
void setup() {
  Serial.begin(9600);
  pinMode(rxPin, INPUT);
  pinMode(txPin, OUTPUT);
  cx.begin(9600);
  for(;fil<n;fil++){
    Ex=Ex+matriz[fil][0];
    Ey=Ey+matriz[fil][1];
    Exy=Exy+(matriz[fil][0]*matriz[fil][1]);
    Ex2=Ex2+pow(matriz[fil][0],2);
    }
  Ex_2=pow(Ex,2);
  Bo=(float(Ex2*Ey)-float(Ex*Exy))/(float(n*Ex2-Ex_2));
  ml=(n*Exy)-(Ex*Ey); //aux de desborde
  m=float(ml)/(float(n*Ex2-Ex_2));
  Serial.println("El modelo es: ");
  Serial.println(String("y= ")+String(m)+String("x")+String(Bo));
  Serial.println("Ingrese su estatura en cm:");
}
```

Se prosigue con la función loop en donde se va activar el ingreso de datos por comunicación serial y la salida de datos del peso con relación a la estatura:

Figura 9: Función Loop en Arduino

```
void loop() {
  if(Serial.available()>0){
    delay(3000);
    dato=Serial.readString();
    estatura=dato.toInt();
    edad=dato.toInt();
    //peso=m*estatura+Bo;
    peso=estatura-100-((estatura-150)/4); // fórmula de Lorentz
    cx.write(estatura);
    Serial.println(String("Su peso es: ")+String(peso)+String("Kg"));
    Serial.println(" ");
    Serial.println("Ingrese su estatura en cm:");
   }
}
```

Algo importante es que se utilizó la fórmula de Lorentz para encontrar el peso según la estatura de la persona, la fórmula que se utilizó es:

PESO = estatura-100-((estatura-150)/4)

Para programar en Processing se comienza declarando las variables a utilizar y de igual manera se importa la biblioteca para conectar con el puerto serial:

Figura 10: Código Processing declaración de variables y matriz

YucaillaV_Lab4_PROCESSING_pde

```
import processing.serial.*;

Serial port; //Objeto para puerto com
int dato=0; // Variable de rx de datos
int aux;

float [][] matriz ={
{(170-154.37)*40.0078064,475-67*5},
{(180-154.37)*40.0078064,475-80*5},
{(170-154.37)*40.0078064,475-65*5},
{(178-154.37)*40.0078064,475-75*5},
{(160-154.37)*40.0078064,475-55*5},
{(163-154.37)*40.0078064,475-60*5},
{(165-154.37)*40.0078064,475-63*5},
{(170-154.37)*40.0078064,475-70*5},
{(164-154.37)*40.0078064,475-62*5},
{(176-154.37)*40.0078064,475-77*5},
{(164-154.37)*40.0078064,475-60*5},
{(170-154.37)*40.0078064,475-76*5},
{(170-154.37)*40.0078064,475-56*5},
{(168-154.37)*40.0078064,475-60*5},
};

```

En la función Setup se comienza a configurar parte de la interfaz ingresando el texto que va aparecer al momento de ejecutar el programa:

Figura 11: Función Setup en Processing

```
void setup() {
  port=new Serial(this,"COM12",9600);
  background(255);
  size(1290,550);
  sep=200;
  sep2=50;
  planoF();
  fill(255,0,0);
  textSize(20);
  text("Puntos del conjunto de entrenamiento correspondientes a la regresión lineal",250,18);
  fill(0);
  text("Eje Altura en cm ",550,520);
  text("E",1250,100);
  text("j",1253,120);
  text("e",1250,140);
  text("P",1250,185);
  text("e",1250,205);
  text("s",1250,225);
  text("o",1250,245);
  text("e",1250,290);
  text("n",1250,310);
  text("K",1250,350);
  text("g",1250,370);
  textSize(13);
  text("Laboratorio 3",20,525);
  text("Yucailla Viviana",20,545);
  fill(#F0C105);
```

En la función draw() se encuentra el código para que aparezca un punto de referencia al peso obtenido al momento de ingresar la estatura en la ventana de cx serial:

Figura 12: Función draw en Processing

```
void draw() {
  if(dato>0){
    y1 = 1.18*dato-133.39;
    fill(255,0,0);
    stroke(0);
    ellipse((dato-154.37)*40.0078064,475-y1*5,10,10);
  }
```

Se creó una función para crear el plano en el que se va a trabajar y va aparecer los puntos del conjunto de entrenamiento correspondientes a la regresión lineal:

Figura 13: Creación del plano en la interfaz Processing

```
void planoF(){
  strokeWeight(1);
  rect(25,25,1200,450);
  for(float i=25;i<=1200;i+=sep){
    line(i,25,i,475);
    k=k+5;
    fill(0);
    textSize(10);
    text(k,i,490);
  }
    for(float j=25;j<=450;j+=sep2){
    line(25,j,1225,j);
    m=m-10;
    fill(0);
    textSize(10);
    text(m,10,j);
  }
    for(int c=0;c<14;c++){
    fill(#E8B936);
    stroke(0);
    ellipse(matriz[c][0],matriz[c][1],10,10);
    }

    for(float v=225;v<=1025;v+=10){
    x++;
    y=1.18*x-133.39;
    fill(0);
```

Para conectar Arduino con Processing y de igual manera los datos ingresados se utiliza la función serialEvent():

Figura 14: Conexión y envío de datos Arduino-Processing

```
void serialEvent(Serial port){

  dato=port.read();
  aux=int(dato); //Convertir de string a int
  println(dato);

}
```

Ya realizado el código en Processing y la simulación en Proteus, se presenta el ingreso por Comunicación Serial y la interfaz de la regresión lineal terminada en donde se puede obervar:

- Puntos del conjunto de entrenamiento de la regresión lineal.
- Tendencia lineal.
- Nombres de los ejes del plano
- Nombre de estudiante.

Figura 15: Ingreso de datos por Cx Serial

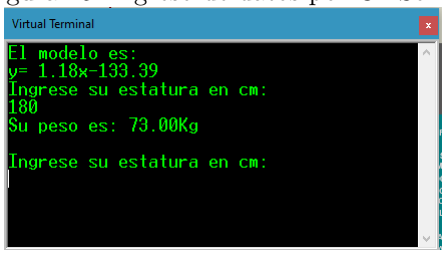
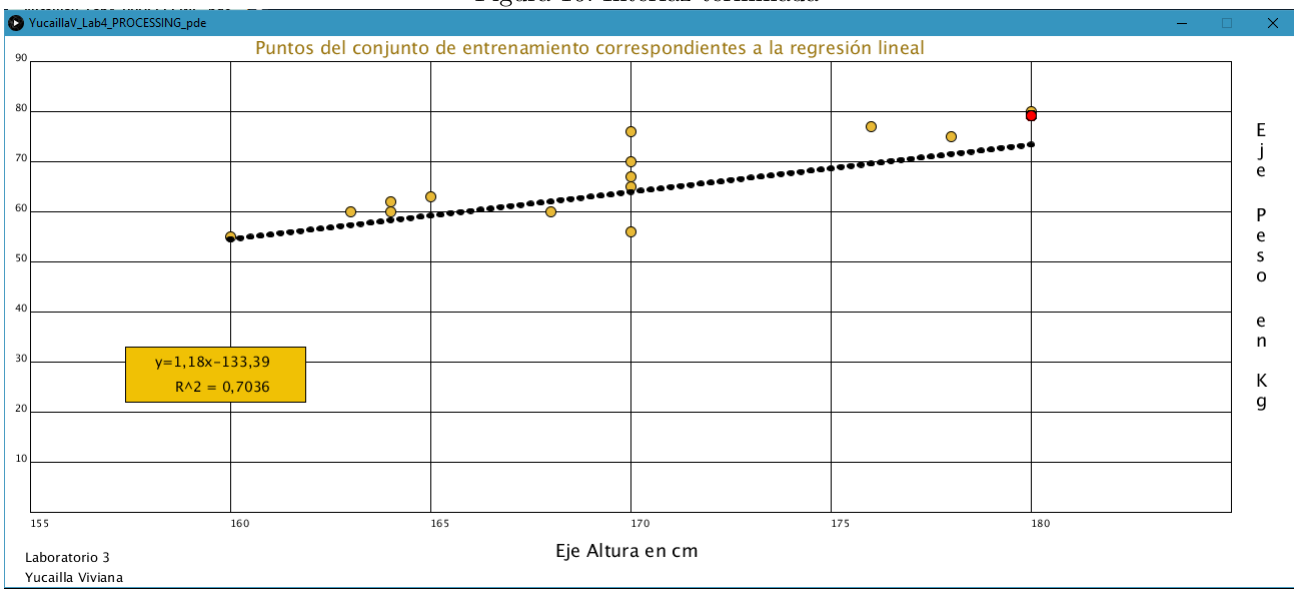


Figura 16: Interfaz terminada



5. Conclusiones y Recomendaciones

- Para la programación en Processing con Arduino y la conexión de datos que se generan, es importante el uso de la función SerialEvent, ya que con la ausencia de la función los datos que se ingresa por comunicación serial no aparecerán en el lenguaje de programación Processing.

- Para el desarrollo de la interfaz se necesita la posible experiencia en el momento de ubicar los gráficos, puntos y en este caso el plano a trabajar para que no haya errores de graficación al momento de ejecutar el programa.

- Se logró desarrollar una interfaz sencilla y simple pero funcional, en el cual de manera didáctica se utiliza la simulación en Proteus para que aparezca el nuevo dato del peso en la interfaz.

- Existe una gran cantidad de herramientas destinadas al desarrollo, en esta práctica se utilizó herramientas que no generarron problemas.

- Se recomienda al momento de crear una interfaz de un plano de regresión lineal, tratar de verificar que no haya errores al momento de que aparezca los puntos de la matriz.

- Se recomienda verificar los datos del peso de la persona en una calculadora para mirar si está ingresada la fórmula correctamente en el código.

6. Link del repositorio en GitHub

A continuación se encuentra el link del repositorio en Github en donde se encuentra la simulación en proteus, el código del Arduino IDE y el código de Processing

https://github.com/Vivi-21/Lab4