MATERIA

Sistemas Programables

CARRERA

Ingeniería en sistemas Computacionales

PRESENTA:

Jorge Alberto Sanchez Maldonado

NOMBRE DE LA MAESTRA:

Ing. Carlos Rafael Levy Rojas

LEÓN, GUANAJUATO **Periodo: Enero-Junio 2018**

INTRODUCCIÓN

Es un proyecto donde se realiza un programa en java, el cual se conecta a Arduino, donde le enviamos datos como un mensaje, la hora y temperatura, cada uno de estos se ejecuta al oprimir un botón, ya que se realizo una interfaz en java, toda esta información se le manda a Arduino en donde el convertirá a ASCII para poder imprimir en pantalla. Para que esto sea posible se realizo una conexión la cual debe de contar con las librerías rxtx y jserialcomm en el proyecto realizado. En el archivo de Arduino se tiene que contar con la librería LiquidCrystal.h para que sea posible mandar la información de Arduino a la pantalla lcd.

Código Arduino

```
#include <LiquidCrystal.h>      //libreria a utilizar para la pantalla lcd

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);      // pines utilizados en arduino para la pantalla lcd
int imprimir=0;                      // variable para la impresion del mensaje
String Mensaje="";                    //variable que obtendra el mensaje

//temperatura
byte PIN_SENSOR = A0;          //pin donde se conecta el sensor lm35
int dato_serial = 0;                    //variable del serial a comunicacion
float C;                               //variable donde se guardara el resultado de la operacion
int temp;                             //escribira la temperatura


//mili
unsigned long time;                //variable de tirmpo
unsigned long t = 0;               //se inicializa t para repetir los mensajes
int Dt = 100;                   //variable que se inicializa, seran los milisegundos que tardara en repetir

void setup(){
pinMode(10,OUTPUT);                    //pin que se dirige al led
digitalWrite(10,HIGH);                 //escribe
lcd.begin(16, 2);                  //Inicializa la interfaz a la pantalla LCD y especifica las dimensiones (ancho y alto)

Serial.begin(9600);           //Establece la velocidad de datos en bits por segundo
//estaes del reloj
Serial.setTimeout(50);         //establece los milisegundos máximos para esperar datos

}

void loop(){
//reloj
String text = Serial.readString();                    //lee caracteres del buffer serial en una cadena. La función finaliza si se agota el tiempo de espera (ver setTimeout ())
String line1 = text.substring(0, 16);                 //Obtener una subcadena de una cadena
String line2 = text.substring(16, 32);                    //Obtener una subcadena de una cadena
```

```
C = (5.0 * analogRead(PIN_SENSOR) * 100.0)/ 1024;        //operacion para determinar la temperatura

//if(text.length() > 0){

 // }

//  lcd.setCursor(0,0);
 // lcd.print(line1);
 // lcd.setCursor(0,1);
 // lcd.print(line2);


 if ( Serial.available() > 0){            //Obtiene la cantidad de bytes (caracteres) disponibles para leer desde el puerto serie
   //lectura_dato();
   dato_serial = Serial.read();          //la variable lee los datos de serie entrantes
   comparacion_dato();
}
    //

   lcd.setCursor(0,0);                  //nos posicionamos en el 0,0
   lcd.print("C=      Grados");          //imprime el mensaje
   lcd.setCursor(2,0);                  //nos posicionamos en el 2,0
   lcd.print(C);                       //se imprime la temperatura
   temp = C;                           //guardamos C en temp
  //
  //



int cuenta=0;                          //variable para llevar conteo de mensaje
int caracteres=0;                      //caracteres entrantes


while (Serial.available()>0){                                  //mientras haya caractere disponibles para leer desde el puerto serie hara los mandara para convertir a ASCII
Mensaje=Mensaje+Decimal_to_ASCII(Serial.read());
//text = text + Decimal_to_ASCII(Serial.read());
cuenta++;                                      //lleva la cuenta de cada vez que pasa un mensaje en partes

}
caracteres=Mensaje.length();                   //cuenta los caracteres que tiene el mensaje
```

```
delay(10000);                                   //espera un tiempo definido

if(text.length() > 0){                          //verifica si text es mayor que cero para
poder imprimir
  lcd.setCursor(0,0);                           //se coloca en la posicion 0,0
  lcd.print("               ");                 //se imprime espacio
  lcd.setCursor(0,1);                           //se coloca en la posicion 0,0
  lcd.print("               ");                 //se imprime espacio
 }

  lcd.setCursor(0,0);                           //se coloca en la posicion 0,0
  lcd.print(line1);                             //se imprime la linea uno de la pantalla
  lcd.setCursor(0,1);                           //se coloca en la posicion 0,1
  lcd.print(line2);                             //se imprime la linea dos de la pantalla
   delay(5000);                                 //tiempo de 5000(5 segundos)


if(time-t > Dt){                                //condicion para que se cicle
if (caracteres>16){                             //condicion que si es mayor a 16 siga
para imprimir
  if (Mensaje!=""){                             //si es diferente de nulo lo va a imprimir
   lcd.clear();                                 //limpia la pantalla lcd
   lcd.print(Mensaje.substring(0,16));          //imprime la primera linea,
primeros 16 caracteres
   lcd.setCursor(0,1);                          //nos posicionamos en la 0,1
   lcd.print(Mensaje.substring(16,caracteres)); //imprime los degundos 16
caracteres
}
}
else                                            //de lo contrario realiza lo siguiente
{
//
if (Mensaje!=""){                               //limpia e imprime
    t = time;
    lcd.clear();
    lcd.print(Mensaje);
  }
 //
}
}
delay(10000);                                   //espera un tiempo(10 segundos)
Mensaje="";                                     //saca mensaje
 lcd.clear();                                   //limpi pantalla
//temperatura
}
```

```
void lectura_dato (void ){                          //metodo para leer el serial, cada
caractaer
 dato_serial = Serial.read();
}

void comparacion_dato (void){                        //si es vacion escribe el temp
  if(dato_serial == ' '){
      Serial.write(temp);


  }
}


char Decimal_to_ASCII(int entrada){//metodo que toma lo enviado por llava y lo
convierte a ASCII para poder imprimir en pantalla, contiene todos las letras,
mayusculas, minisculas y simbolos
  char salida=' ';
  switch(entrada){
case 32:
salida=' ';
break;
case 33:
salida='!';
break;
case 34:
salida="";
break;
case 35:
salida='#';
break;
case 36:
salida='$';
break;
case 37:
salida='%';
break;
case 38:
salida='&';
break;
case 39:
salida=' ';
break;
case 40:
salida='(';
break;
case 41:
```

```
salida=')';
break;
case 42:
salida='*';
break;
case 43:
salida='+';
break;
case 44:
salida=',';
break;
case 45:
salida='-';
break;
case 46:
salida='.';
break;
case 47:
salida='/';
break;
case 48:
salida='0';
break;
case 49:
salida='1';
break;
case 50:
salida='2';
break;
case 51:
salida='3';
break;
case 52:
salida='4';
break;
case 53:
salida='5';
break;
case 54:
salida='6';
break;
case 55:
salida='7';
break;
case 56:
salida='8';
break;
```

```
case 57:
salida='9';
break;
case 58:
salida=':';
break;
case 59:
salida=';';
break;
case 60:
salida='<';
break;
case 61:
salida='=';
break;
case 62:
salida='>';
break;
case 63:
salida='?';
break;
case 64:
salida='@';
break;
case 65:
salida='A';
break;
case 66:
salida='B';
break;
case 67:
salida='C';
break;
case 68:
salida='D';
break;
case 69:
salida='E';
break;
case 70:
salida='F';
break;
case 71:
salida='G';
break;
case 72:
salida='H';
```

```
break;
case 73:
salida='I';
break;
case 74:
salida='J';
break;
case 75:
salida='K';
break;
case 76:
salida='L';
break;
case 77:
salida='M';
break;
case 78:
salida='N';
break;
case 79:
salida='O';
break;
case 80:
salida='P';
break;
case 81:
salida='Q';
break;
case 82:
salida='R';
break;
case 83:
salida='S';
break;
case 84:
salida='T';
break;
case 85:
salida='U';
break;
case 86:
salida='V';
break;
case 87:
salida='W';
break;
case 88:
```

```
salida='X';
break;
case 89:
salida='Y';
break;
case 90:
salida='Z';
break;
case 91:
salida='[';
break;
case 92:
salida=' ';
break;
case 93:
salida=']';
break;
case 94:
salida='^';
break;
case 95:
salida='_';
break;
case 96:
salida='`';
break;
case 97:
salida='a';
break;
case 98:
salida='b';
break;
case 99:
salida='c';
break;
case 100:
salida='d';
break;
case 101:
salida='e';
break;
case 102:
salida='f';
break;
case 103:
salida='g';
break;
```

```
case 104:
salida='h';
break;
case 105:
salida='i';
break;
case 106:
salida='j';
break;
case 107:
salida='k';
break;
case 108:
salida='l';
break;
case 109:
salida='m';
break;
case 110:
salida='n';
break;
case 111:
salida='o';
break;
case 112:
salida='p';
break;
case 113:
salida='q';
break;
case 114:
salida='r';
break;
case 115:
salida='s';
break;
case 116:
salida='t';
break;
case 117:
salida='u';
break;
case 118:
salida='v';
break;
case 119:
salida='w';
```

```
break;
case 120:
salida='x';
break;
case 121:
salida='y';
break;
case 122:
salida='z';
break;
case 123:
salida='{';
break;
case 124:
salida='|';
break;
case 125:
salida='}';
break;
case 126:
salida='~';
break;
  }
  return salida;              //regresa la salida
}
```

Código java

```
package mensaje;

import gnu.io.CommPortIdentifier;
import gnu.io.SerialPort;
import gnu.io.UnsupportedCommOperationException;
import java.awt.event.ActionEvent;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.io.PrintWriter;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Enumeration;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JFrame;


public class vista_Mensaje extends javax.swing.JFrame {

    int caracteres = 32;              //maximo de caracteres a imprimir
    private OutputStream Output = null;      //variable de conexion
    SerialPort serialPort;                        //serial que hara posible detectar y leer,
(conexion)
    private final String PORT_NAME = "COM6"; //puerto a utilizar
    private static final int TIME_OUT = 2000;
    private static final int DATA_RATE = 9600;  //bits por segundo

    //TEMP
    InputStream in = null;
    int temperatura=10;
    Thread timer;

    public vista_Mensaje() {
        initComponents();
        ArduinoConnection();//se inicializa la conexion
        letras();          //metodo letras para que empieze en el constructor
        //EnviarDatos();
    }

    public void ArduinoConnection() {  //metodo de conexion a arduino
        CommPortIdentifier portId = null;      //identificador de puerto
```

```
        Enumeration portEnum = CommPortIdentifier.getPortIdentifiers();  //obtiene el identificador

        while (portEnum.hasMoreElements()) {        //si tiene mas elementos ejecuta lo siguiente
        CommPortIdentifier currPortId = (CommPortIdentifier) portEnum.nextElement();

        if (PORT_NAME.equals(currPortId.getName())) {  //si el puerto es el mismo conecta
        portId = currPortId;
        break;
            }
        }

        if (portId == null) {       //si no esta manda error y saca del programa
        System.exit(ERROR);
        return;
        }

        try {

            //parametros necesarios para la conexion serial
        serialPort = (SerialPort) portId.open(this.getClass().getName(), TIME_OUT);
        serialPort.setSerialPortParams(DATA_RATE,
        SerialPort.DATABITS_8,
        SerialPort.STOPBITS_1,
        SerialPort.PARITY_NONE);
        Output = serialPort.getOutputStream();

        } catch (Exception e) {

        System.exit(ERROR);
        }
    }

    private void EnviarDatos(String data) {
        try {
        Output.write(data.getBytes());  //envia datos de byte por baty que toma arduino con la funcion
        } catch (IOException e) {       //si no los obtiene manda un error
        System.exit(ERROR);
        }
    }

    public void letras() {
```

```
        caracteres = 32 - txtMensaje.getText().length(); //Indica la cantidad de caracteres
        //disponibles. En el LCD solo se permite imprimir 32 caracteres.

        if (caracteres <= 0) { //Si la cantidad de caracteres se ha agotado...
        labelCaracter.setText("Caracteres disponibles: 0"); //Se imprime que la cantidad de caracteres disponibles es 0
        String cadena = ""; //Se declara la variable que guardará el mensaje a enviar
        cadena = txtMensaje.getText(); //Se asigna el texto del TextField a la variable cadena
        cadena = cadena.substring(0, 32); //Se evita que por alguna razón la variable contenga
        //más de 32 caracteres, utilizando el substring que crea un string a partir de uno mayor.
        txtMensaje.setText(cadena); //se regresa la cadena con 32 caracteres al TextField
        } else {
        //Si la cantidad de caracteres disponibles es ayor a 0 solamente se imprimirá la cantidad de caracteres disponibles
        labelCaracter.setText("Caracteres disponibles: " + (caracteres));
        }
    }

    public void hora(){
        Thread thread = new Thread() {
            @Override
            public void run() {
                try {
                    Thread.sleep(100);  //hilo que hace espera
                } catch (Exception e) {
                }

                PrintWriter output;
                try {
                    //hacemos conexion mediante el puerto
                    output = new PrintWriter(serialPort.getOutputStream());
                    while (true) {
                        //se da el formato de hora desde java para mandar a arduino
                        output.print(new SimpleDateFormat("hh:mm:ss a      MMMMMMM dd, yyyy").format(new Date()));
                        output.flush();
                        try {
                            Thread.sleep(100); //hilo que hara espera
                        } catch (Exception e) {
                        }
                    }
                } catch (IOException ex) {
```

```
Logger.getLogger(vista_Mensaje.class.getName()).log(Level.SEVERE, null, ex);
                }
            }
        };
        thread.start();
    }


    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        txtMensaje = new javax.swing.JTextField();
        labelCaracter = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        btnVer = new javax.swing.JButton();
        btnLimpiar = new javax.swing.JButton();
        btnHora = new javax.swing.JButton();
        btnTemperatura = new javax.swing.JButton();


setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        txtMensaje.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                txtMensajeActionPerformed(evt);
            }
        });
        txtMensaje.addKeyListener(new java.awt.event.KeyAdapter() {
            public void keyReleased(java.awt.event.KeyEvent evt) {
                txtMensajeKeyReleased(evt);
            }
        });

        jLabel2.setText("Escribe un Mensaje");

        btnVer.setText("Ver Mensaje");
        btnVer.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnVerActionPerformed(evt);
            }
        });

        btnLimpiar.setText("Limpiar Mensaje");
        btnLimpiar.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnLimpiarActionPerformed(evt);
```

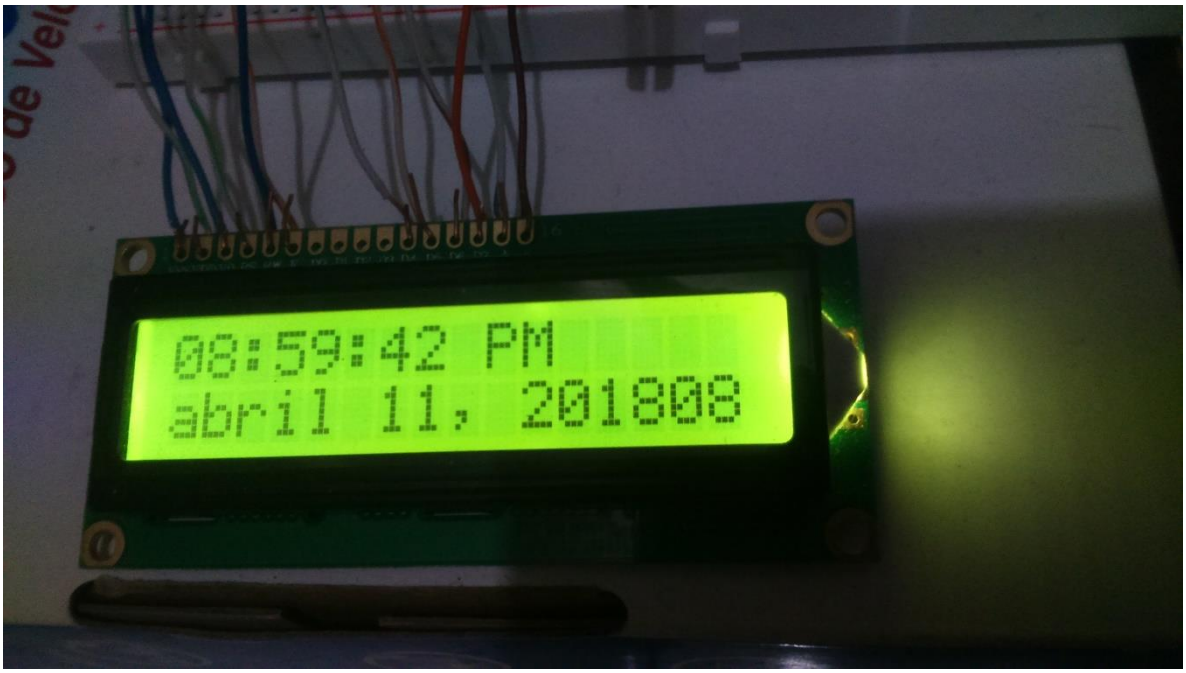
```
        }
    });

    btnHora.setText("Ver Hora");
    btnHora.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnHoraActionPerformed(evt);
        }
    });

    btnTemperatura.setText("Ver Temperatura");
    btnTemperatura.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnTemperaturaActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
            .addGroup(layout.createSequentialGroup()
                .addGap(43, 43, 43)
                .addComponent(txtMensaje,
javax.swing.GroupLayout.PREFERRED_SIZE, 302,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(layout.createSequentialGroup()
                .addGap(139, 139, 139)
                .addComponent(jLabel2,
javax.swing.GroupLayout.PREFERRED_SIZE, 140,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(layout.createSequentialGroup()
                .addGap(73, 73, 73)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
                    .addComponent(labelCaracter,
javax.swing.GroupLayout.PREFERRED_SIZE, 218,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGroup(layout.createSequentialGroup()
```

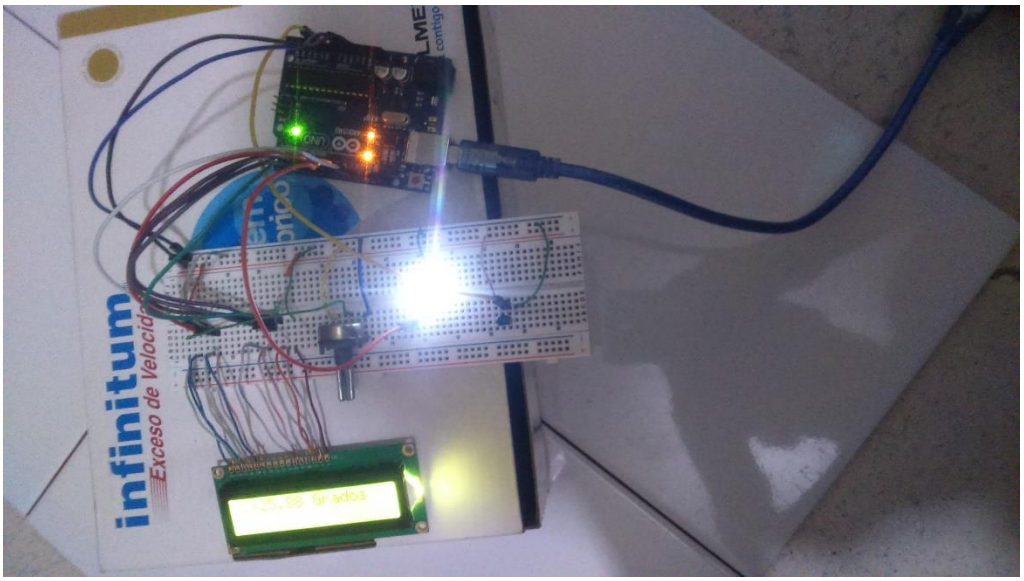
```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
                            .addComponent(btnVer)
                            .addComponent(btnHora))
                        .addGap(41, 41, 41)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
                            .addComponent(btnTemperatura)
                            .addComponent(btnLimpiar))))))
                .addContainerGap(55, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(26, 26, 26)
                .addComponent(jLabel2)
                .addGap(41, 41, 41)
                .addComponent(txtMensaje,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addComponent(labelCaracter,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    21,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BAS
ELINE)
                    .addComponent(btnVer)
                    .addComponent(btnLimpiar))
                .addGap(32, 32, 32)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BAS
ELINE)
                    .addComponent(btnHora)
                    .addComponent(btnTemperatura))
                .addContainerGap(64, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>

    private void txtMensajeKeyReleased(java.awt.event.KeyEvent evt) {
```

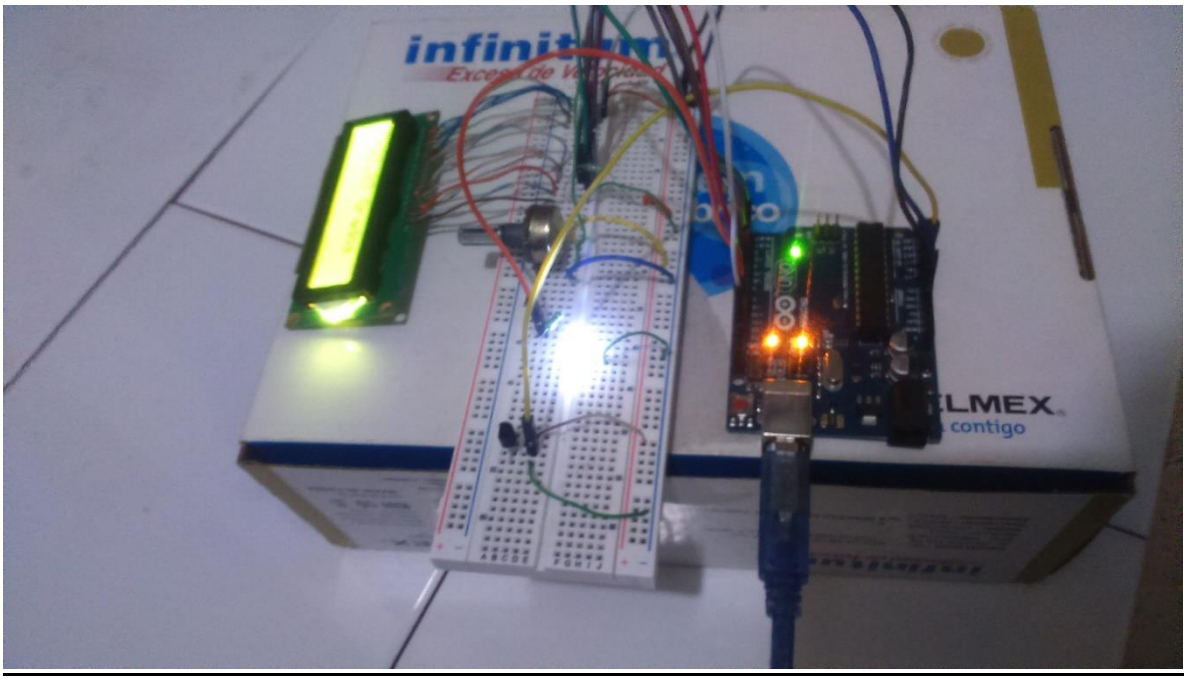
```
        letras();         //llama el metodo letras para contar cuantos caracteres de han escrito en el textfield
    }

    private void btnVerActionPerformed(java.awt.event.ActionEvent evt) {
        EnviarDatos(txtMensaje.getText()); //envia los datos que se ingresaron en el textfield
        txtMensaje.setText("");           //deja el campo en blanco

        letras();                        //llama al metodo letras
        hora();                          //lama al metodo horas
          Thread thread = new Thread() {
           @Override
           public void run() {
               try {
                   Thread.sleep(5000);
               } catch (Exception e) {
               }
                   while (true) {

                       try {
                           Thread.sleep(5000);
                       } catch (Exception e) {
                       }
                   }
          }
        };
        thread.start();


    }

    private void btnLimpiarActionPerformed(java.awt.event.ActionEvent evt) {
        txtMensaje.setText("");  //limpia el campo y llama metodo
        letras();
    }

    private void txtMensajeActionPerformed(java.awt.event.ActionEvent evt) {
        EnviarDatos(txtMensaje.getText());   //envia mensaje
        txtMensaje.setText("");        //deja el campo en blanci
    }

    private void btnHoraActionPerformed(java.awt.event.ActionEvent evt) {
        hora();                          //llama la hora
        EnviarDatos(txtMensaje.getText());   //envia informacion
        txtMensaje.setText("");              //limpia textfield
        letras();                        //llama al metodo letras para el mensaje
```

```
    }

    public void temperatura(){

        serialPort.close();   //cierra conexion
        try {
            Output.close();           //cierra conexion
        } catch (IOException ex) {
            Logger.getLogger(vista_Mensaje.class.getName()).log(Level.SEVERE,
null, ex);
        }
            //this.dispose();
//            Ventana2 v= new Ventana2();
//            v.setVisible(true);
//            v.setSize(400,300);
    }



    private void btnTemperaturaActionPerformed(java.awt.event.ActionEvent evt) {
        temperatura();      //llama a temperatura
    }


    public static void main(String args[]) {

        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new vista_Mensaje().setVisible(true);



            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton btnHora;
    private javax.swing.JButton btnLimpiar;
    private javax.swing.JButton btnTemperatura;
    private javax.swing.JButton btnVer;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel labelCaracter;
    private javax.swing.JTextField txtMensaje;
    // End of variables declaration
}
```

Imágenes


08:59:42 PM
abril 11, 201808


infinitum
Exceso de Velocid


infinitum
Excede lo increíble
LMEX
contigo