**INSTITUTO TECNOLÓGICO DE LEÓN**

**PROYECTO DE MENSAJE EN LCD CON PROGRAMACIÓN EN JAVA.**

**ALUMNOS:**

Roberto Ismael Ramírez Rios.

Paola Castillo Irene.

Angely Ramírez Martínez.

**PROFESOR:**

ING. Levy Rojas Carlos Rafael.

**Fecha de Entrega:** 25-oct-2018.

**INTRODUCCIÓN**

Este proyecto muestra mensajes en una pantalla LCD desde una ventana en java que cualquier usuario puede escribir dentro de la pantalla que hemos creado en este lenguaje de java. Los mensajes, como se puede observar en las imágenes, obviamente son mensajes cortos, esto con la intención de que nuestra pantalla LCD no tenga problemas al momento de mostrarlos.

Las ventanas hechas en java serán ejecutadas únicamente en una computadora u ordenador, ya que hasta el momento solamente hemos creado ese tipo de interfaces, aunque se prevé crearlas dentro del ambiente Android para que puedas escribir mensajes desde tu celular.

Espero este proyecto cumpla con las expectativas de la materia.

**DESARROLLO**

**CÓDIGO.**

* **Arduino -**

#include <LiquidCrystal.h> //librería 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 impresión del mensaje

String Mensaje=""; //variable que obtendrá el mensaje

//temperatura

byte PIN\_SENSOR = A0; //pin donde se conecta el sensor lm35

int dato\_serial = 0; //variable del serial a comunicación

float C; //variable donde se guardara el resultado de la operación

int temp; //escribirá la temperatura

//mili

unsigned long time; //variable de tiempo

unsigned long t = 0; //se inicializa t para repetir los mensajes

int Dt = 100; //variable que se inicializa, serán 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; //operación 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 caracteres 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 posición 0,0

lcd.print(" "); //se imprime espacio

lcd.setCursor(0,1); //se coloca en la posición 0,0

lcd.print(" "); //se imprime espacio

}

lcd.setCursor(0,0); //se coloca en la posición 0,0

lcd.print(line1); //se imprime la línea uno de la pantalla

lcd.setCursor(0,1); //se coloca en la posición 0,1

lcd.print(line2); //se imprime la línea dos de la pantalla

delay(10000); //tiempo de 5000(5 segundos)

if(time-t > Dt){ //condición para que se cicle

if (caracteres>16){ //condición 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 línea, primeros 16 caracteres

lcd.setCursor(0,1); //nos posicionamos en la 0,1

lcd.print(Mensaje.substring(16,caracteres)); //imprime los segundos 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(); //limpia la pantalla

//temperatura

}

void lectura\_dato (void ){ //método para leer el serial, cada carácter

dato\_serial = Serial.read();

}

void comparacion\_dato (void){ //si es vacío escribe el temp

if(dato\_serial == ' '){

Serial.write(temp);

}

}

char Decimal\_to\_ASCII(int entrada){ //método que toma lo enviado por llave y lo convierte a ASCII para poder imprimir en pantalla, contiene todos las letras, mayúsculas, minúsculas y símbolos

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

}

* **Vista\_mensaje –**

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;

//import static mensaje.RelojArduino.serialPort;

//import comfazecast.jSerialComm.SerialPort;

//import jssc.SerialPort;

//-Djava.library.path= "C:\Program Files\Java\jdk1.8.0\_162\jre\lib"

public class vista\_Mensaje extends javax.swing.JFrame {

int caracteres = 32;

private OutputStream Output = null;

SerialPort serialPort;

private final String PORT\_NAME = "COM6";

private static final int TIME\_OUT = 2000;

private static final int DATA\_RATE = 9600;

//TEMP

InputStream in = null;

int temperatura=10;

Thread timer;

public vista\_Mensaje() {

initComponents();

ArduinoConnection();

letras();

//EnviarDatos();

}

public void ArduinoConnection() {

CommPortIdentifier portId = null;

Enumeration portEnum = CommPortIdentifier.getPortIdentifiers();

while (portEnum.hasMoreElements()) {

CommPortIdentifier currPortId = (CommPortIdentifier) portEnum.nextElement();

if (PORT\_NAME.equals(currPortId.getName())) {

portId = currPortId;

break;

}

}

if (portId == null) {

System.exit(ERROR);

return;

}

try {

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());

} catch (IOException e) {

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 guardarÃ¡ 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 razÃ³n la //variable contenga

//mÃ¡s 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 imprimirÃ¡ la cantidad de caracteres disponibles

labelCaracter.setText("Caracteres disponibles: " + (caracteres));

}

}

public void hora(){

Thread thread = new Thread() {

@Override

public void run() {

// wait after connecting, so the bootloader can finish

try {

Thread.sleep(100);

} catch (Exception e) {

}

// enter an infinite loop that sends text to the arduino

PrintWriter output;

try {

output = new PrintWriter(serialPort.getOutputStream());

while (true) {

output.print(new SimpleDateFormat("hh:mm:ss a MMMMMMM dd, yyyy").format(new Date()));

output.flush();

try {

Thread.sleep(100);

} 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">//GEN-BEGIN:initComponents

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.LEADING)

.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.LEADING)

.addComponent(labelCaracter, javax.swing.GroupLayout.PREFERRED\_SIZE, 218, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(layout.createSequentialGroup()

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

.addComponent(btnVer)

.addComponent(btnHora))

.addGap(41, 41, 41)

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

.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.BASELINE)

.addComponent(btnVer)

.addComponent(btnLimpiar))

.addGap(32, 32, 32)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(btnHora)

.addComponent(btnTemperatura))

.addContainerGap(64, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>//GEN-END:initComponents

private void txtMensajeKeyReleased(java.awt.event.KeyEvent evt) {//GEN-FIRST:event\_txtMensajeKeyReleased

letras();

}//GEN-LAST:event\_txtMensajeKeyReleased

private void btnVerActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_btnVerActionPerformed

EnviarDatos(txtMensaje.getText());

txtMensaje.setText("");

letras();

Thread thread = new Thread() {

@Override

public void run() {

try {

Thread.sleep(5000);

} catch (Exception e) {

}

while (true) {

hora();

try {

Thread.sleep(5000);

} catch (Exception e) {

}

}

}

};

thread.start();

}//GEN-LAST:event\_btnVerActionPerformed

private void btnLimpiarActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_btnLimpiarActionPerformed

txtMensaje.setText("");

letras();

}//GEN-LAST:event\_btnLimpiarActionPerformed

private void txtMensajeActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_txtMensajeActionPerformed

EnviarDatos(txtMensaje.getText());

txtMensaje.setText("");

}//GEN-LAST:event\_txtMensajeActionPerformed

private void btnHoraActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_btnHoraActionPerformed

hora();

letras();

}

public void temperatura(){

serialPort.close();

try {

Output.close();

} 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();

{

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//GEN-BEGIN:variables

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;

}

* **Ventana –**

package mensaje;

import gnu.io.CommPortIdentifier;

import gnu.io.PortInUseException;

import gnu.io.SerialPort;

import gnu.io.UnsupportedCommOperationException;

import java.awt.Color;

import java.awt.Graphics;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.IOException;

import java.io.InputStream;

import java.io.OutputStream;

import java.util.Enumeration;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JTextField;

import javax.swing.border.EmptyBorder;

public class Ventana extends JFrame {

///////

private JPanel contentPane;

private JTextField textField;

Enumeration puertos\_libres =null;

CommPortIdentifier port=null;

SerialPort puerto\_ser = null;

OutputStream out = null;

InputStream in = null;

int temperatura=10;

Thread timer;

JLabel lblNewLabel;

JButton btnNewButton,btnNewButton\_1;

public Ventana() {

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 636, 365);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

timer = new Thread(new ImplementoRunnable());

timer.start();

timer.interrupt();

btnNewButton = new JButton("Conectar");

btnNewButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

puertos\_libres = CommPortIdentifier.getPortIdentifiers();

int aux=0;

while (puertos\_libres.hasMoreElements())

{

port = (CommPortIdentifier) puertos\_libres.nextElement();

int type = port.getPortType();

if (port.getName().equals(textField.getText()))

{

try {

puerto\_ser = (SerialPort) port.open("puerto serial", 2000);

int baudRate = 9600; // 9600bps

puerto\_ser.setSerialPortParams(

baudRate,

SerialPort.DATABITS\_8,

SerialPort.STOPBITS\_1,

SerialPort.PARITY\_NONE);

puerto\_ser.setDTR(true);

out = puerto\_ser.getOutputStream();

in = puerto\_ser.getInputStream();

btnNewButton\_1.setEnabled(true);

btnNewButton.setEnabled(false);

timer.resume();

} catch ( IOException e1) {

} catch (PortInUseException e1) {

e1.printStackTrace();

} catch (UnsupportedCommOperationException e1) {

e1.printStackTrace();

}

break;

}

}

}

});

btnNewButton.setBounds(38, 63, 89, 23);

contentPane.add(btnNewButton);

btnNewButton\_1 = new JButton("Desconectar");

btnNewButton\_1.setEnabled(false);

btnNewButton\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

timer.interrupt();

puerto\_ser.close();

btnNewButton\_1.setEnabled(false);

btnNewButton.setEnabled(true);

}

});

btnNewButton\_1.setBounds(205, 63, 128, 23);

contentPane.add(btnNewButton\_1);

textField = new JTextField();

textField.setBounds(38, 32, 86, 20);

contentPane.add(textField);

textField.setColumns(10);

JLabel lblPuertoCom = new JLabel("Puerto COM");

lblPuertoCom.setBounds(37, 11, 90, 14);

contentPane.add(lblPuertoCom);

lblNewLabel = new JLabel("Temperatura");

lblNewLabel.setBounds(80, 124, 128, 24);

lblNewLabel .setFont(new java.awt.Font("Arial", 0, 20));

lblNewLabel .setForeground(Color.blue);

contentPane.add(lblNewLabel);

}public void paint(Graphics g) {

super.paint(g);

g.setColor(Color.blue);

g.fillRect(38, 250-temperatura, 20, temperatura);

try {

Thread.sleep(200);

} catch (InterruptedException e) {

e.printStackTrace();

}}

private class ImplementoRunnable implements Runnable{

int aux;

public void run() {

while(true){

try {

out.write('T');

Thread.sleep(100);

aux = in.read();

if (aux!=2){

temperatura = aux;

lblNewLabel.setText(""+temperatura+" ÂºC");

System.out.println(aux);

}repaint();

} catch (Exception e1) {

}}}}}

* **Ventana2 –**

package mensaje;

import gnu.io.CommPortIdentifier;

import gnu.io.PortInUseException;

import gnu.io.SerialPort;

import gnu.io.UnsupportedCommOperationException;

import java.awt.Color;

import java.awt.Graphics;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.IOException;

import java.io.InputStream;

import java.io.OutputStream;

import java.util.Enumeration;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JTextField;

import javax.swing.border.EmptyBorder;

public class Ventana2 extends JFrame {

//

private JPanel contentPane;

private JTextField textField;

Enumeration puertos\_libres =null;

CommPortIdentifier port=null;

SerialPort puerto\_ser = null;

OutputStream out = null;

InputStream in = null;

int temperatura=10;

Thread timer;

JLabel lblNewLabel;

JButton btnNewButton,btnNewButton\_1;

public Ventana2() {

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 636, 365);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

timer = new Thread(new ImplementoRunnable());

timer.start();

timer.interrupt();

btnNewButton = new JButton("Iniciar");

btnNewButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

puertos\_libres = CommPortIdentifier.getPortIdentifiers();

int aux=0;

while (puertos\_libres.hasMoreElements())

{

port = (CommPortIdentifier) puertos\_libres.nextElement();

int type = port.getPortType();

if (port.getName().equals(textField.getText()))

{

try {

puerto\_ser = (SerialPort) port.open("puerto serial", 2000);

int baudRate = 9600; // 9600bps

puerto\_ser.setSerialPortParams(

baudRate,

SerialPort.DATABITS\_8,

SerialPort.STOPBITS\_1,

SerialPort.PARITY\_NONE);

puerto\_ser.setDTR(true);

out = puerto\_ser.getOutputStream();

in = puerto\_ser.getInputStream();

btnNewButton\_1.setEnabled(true);

btnNewButton.setEnabled(false);

timer.resume();

} catch ( IOException e1) {

} catch (PortInUseException e1) {

e1.printStackTrace();

} catch (UnsupportedCommOperationException e1) {

e1.printStackTrace();

}

break;

}

}

}

});

btnNewButton.setBounds(38, 63, 89, 23);

contentPane.add(btnNewButton);

btnNewButton\_1 = new JButton("Ver");

btnNewButton\_1.setEnabled(false);

btnNewButton\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

timer.interrupt();

puerto\_ser.close();

btnNewButton\_1.setEnabled(false);

btnNewButton.setEnabled(true);

}

});

btnNewButton\_1.setBounds(205, 63, 128, 23);

contentPane.add(btnNewButton\_1);

textField = new JTextField();

textField.setBounds(38, 32, 86, 20);

contentPane.add(textField);

textField.setColumns(10);

JLabel lblPuertoCom = new JLabel("Puerto COM");

lblPuertoCom.setBounds(37, 11, 90, 14);

contentPane.add(lblPuertoCom);

lblNewLabel = new JLabel("Temperatura");

lblNewLabel.setBounds(80, 124, 128, 24);

lblNewLabel .setFont(new java.awt.Font("Arial", 0, 20));

lblNewLabel .setForeground(Color.blue);

//contentPane.add(lblNewLabel);

}public void paint(Graphics g) {

super.paint(g);

g.setColor(Color.blue);

g.fillRect(38, 250-temperatura, 20, temperatura);

try {

Thread.sleep(200);

} catch (InterruptedException e) {

e.printStackTrace();

}}

private class ImplementoRunnable implements Runnable{

int aux;

public void run() {

while(true){

try {

out.write(' ');

Thread.sleep(100);

aux = in.read();

if (aux!=2){

temperatura = aux;

lblNewLabel.setText(""+temperatura+" ÂºC");

System.out.println(aux);

}repaint();

} catch (Exception e1) {

}}}}

// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents

private void initComponents() {

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

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

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

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

.addGap(0, 400, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

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

.addGap(0, 300, Short.MAX\_VALUE)

);

pack();

}// </editor-fold>//GEN-END:initComponents

public static void main(String args[]) {

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new Ventana2().setVisible(true);

}

});

}

// Variables declaration - do not modify//GEN-BEGIN:variables

// End of variables declaration//GEN-END:variables

}

* **Main –**

package mensaje;

public class main {

public static void main(String[] args) {

vista\_Mensaje vm = new vista\_Mensaje();

vm.setVisible(true);

vm.setSize(400,300);

}

}

**CONCLUSIÓN**

Tuvimos bastantes errores al momento de codificar el programa, incluso se nos hizo muy difícil la parte de conectar el Arduino con java. Tuvimos que investigar todos esos temas de conexión para poder terminar el proyecto.

Cuando logramos corregir la mayoría de los errores se nos presentaron algunos otros más como lo que fue la conectividad de los mensajes escritos con la reproducción de los mismos en la pantalla LCD. Sin embargo, después de tanta prueba y error logramos finalizar de una manera correcta nuestro programa.

Aprendimos muchas cosas como fue la conexión entre java y Arduino, la parte grafica de enviar mensajes desde java a la pantalla LCD, pero sobre todo aprendimos a corregir de forma autónoma los errores que se nos mostraban al momento de codificar (como se mencionó anteriormente).