

CARRO SEGUIDOR DE RUTA

Laboratorio II de Electrónica



Universidad
del Cauca

Presentado por:

Jhareb Sánchez Peláez

Edwin Stiven Molina Arias

Cristian Daniel Benavides Ospina

Presentado a:

Fabio Hernán Realpe Martinez

FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES

UNIVERSIDAD DEL CAUCA

Popayán

INTRODUCCIÓN

Se presenta el diseño e implementación de un prototipo planteado en el "*laboratorio II de electrónica*", el cual se llevará a cabo a lo largo del semestre. El prototipo se basa en el desarrollo de un auto seguidor de ruta el cuál usa algoritmo programados en el software de desarrollo *arduino* y *java*. A través del IDE de Arduino, se realiza el algoritmo encargados de darle funcionalidad a todo el sistema hardware (sensores, motores, placa e integrado) del prototipo, y a través de un compilador, en este caso *NetBeans*, se desarrolla la parte software tal como la interfaz gráfica a partir de la cuál se define la ruta y el envío de la misma hacía la placa base del prototipo *ESP8266*.

OBJETIVOS

Objetivo general:

Diseñar e implementar un vehículo funcional el cuál permita seguir una trayectoria definida a través de una interfaz gráfica.

Objetivos específicos:

- Realizar una respectiva investigación de documentación bibliográfica.
- Realizar un previo análisis para conocer los elementos necesarios para desarrollar el prototipo.
- Adquirir los elementos y dispositivos necesarios para la implementación del prototipo inicial.
- Implementar el auto con un algoritmo que permita controlar el auto mediante control remoto para verificar la funcionalidad de todos los dispositivos.
- Desarrollar el código en Arduino que permita ejecutar y calibrar las posibles direcciones que tomará el vehículo.
- Diseñar la interfaz gráfica en java, en la cual se logre definir la ruta deseada.
- Conectar la interfaz gráfica con el módulo principal del prototipo para enviar la ruta y realizarla posteriormente.

Semana 1: 21 de abril del 2022

CARRO SEGUIDOR DE RUTA

Inicialmente se adquirieron los siguientes componentes para el montaje del carro

- Un chasis del carro.
- Una rueda loca un broche con plug de 9V.
- Un porta pilas AAX4.
- Dos motorreductores de 6V.
- Un L293D.
- Un HC-06.
- Un Arduino Mega 2560.
- Una protoboard mediana.
- Tornillos.
- Cables dupont macho macho.
- Alambre de cobre.
- Una batería alcalina de 9V.
- 4 baterías alcalinas AA.

La implementación del proyecto inició con la construcción de la base del carro compuesta por el chasis ,las ruedas, los encoders y los motorreductores ;ya montada esta parte se procedió a comprobar el funcionamiento de los motorreductores y se confirmó que funcionaban correctamente puesto que las ruedas del vehículo se movían con velocidad y sin inconveniente.

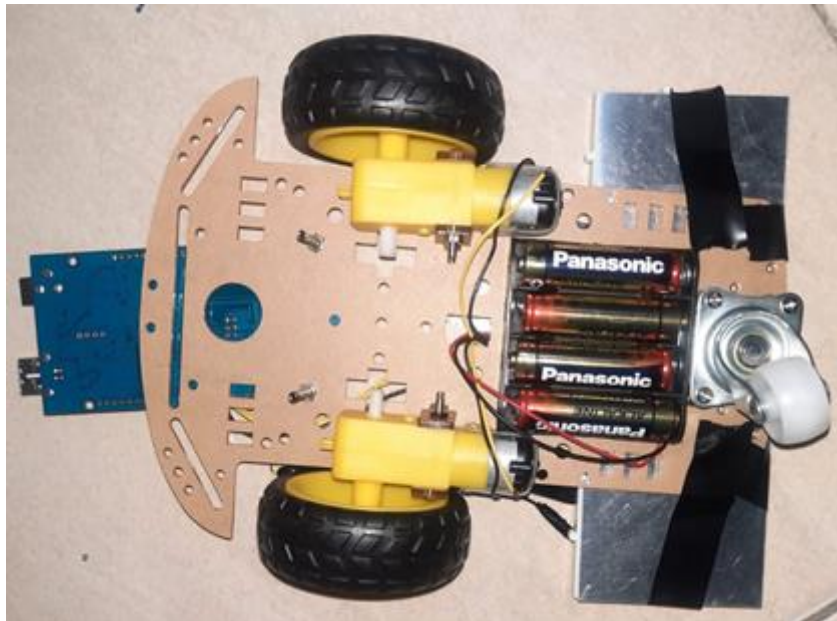


Fig 1. Parte posterior del carro

A continuación se realizaron las conexiones de arduino con el driver L293D , el módulo bluetooth HC 06, los motorreductores y las baterías.

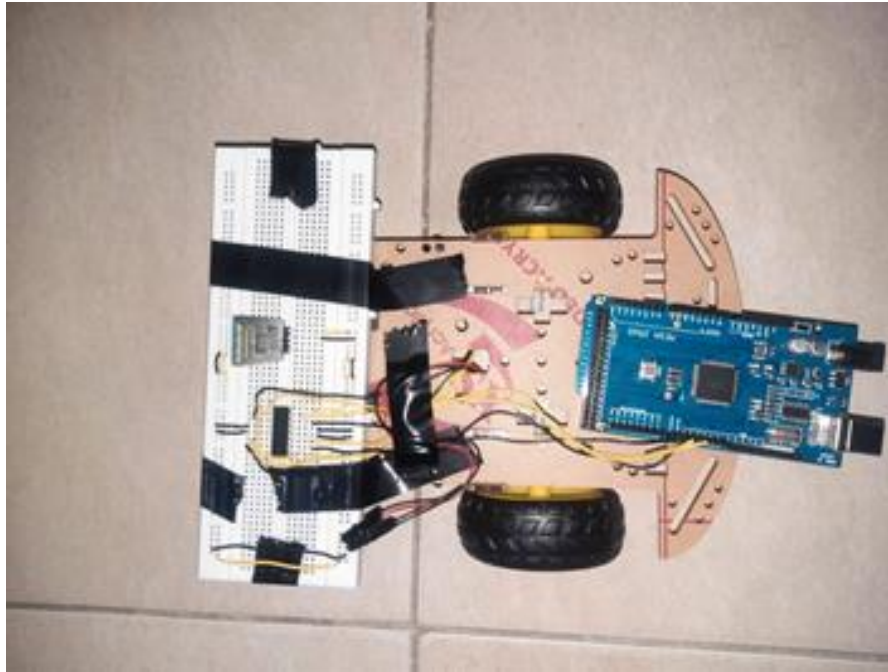


Fig 2. Parte superior del carro

Aunque se realizaron las conexiones en ocasiones oportunidades no se logró encender el módulo bluetooth, por ello pensamos que es necesario medir en el laboratorio el consumo de los componentes utilizados para averiguar si el problema de las conexiones eléctricas radica en que nos estamos suministrando suficiente corriente a los componentes o existe algún tipo de daño en los mismos o problemas de conexión con los cables.

Semana 2: 5 de mayo del 2022

Con el propósito de probar el correcto funcionamiento de los componentes y la correcta conexión de todo el sistema se utilizó el siguiente código que corresponde a un carro manejado por control remoto mediante una aplicación móvil.

```
#include <SoftwareSerial.h>
#include <Servo.h>

// CONEXIONES PARA EL BLUETOOTH.
int bluetoothTx = 2;
int bluetoothRx = 3;
SoftwareSerial bluetooth(bluetoothTx, bluetoothRx);
// MOTOR 1.
int Motor1A = 5;
int Motor1B = 6;

// MOTOR 2.
int Motor2A = 9;
int Motor2B = 10;
void setup ()
{
  bluetooth.begin(115200);
  bluetooth.print("$$$");
  delay(100);
  bluetooth.println("U,9600,N");
  bluetooth.begin(9600);
}
```

```

pinMode( Motor1A, OUTPUT );
pinMode( Motor2A, OUTPUT );
pinMode( Motor1B, OUTPUT );
pinMode( Motor2B, OUTPUT );

digitalWrite( Motor1A, LOW );
digitalWrite( Motor2A, LOW );
digitalWrite( Motor1B, LOW );
digitalWrite( Motor2B, LOW );
}

int flag1 = -1;
int flag2 = -1;

void loop()
{
  if(blueetooth.available())
  {
    char toSend = (char)blueetooth.read();
    if(toSend == 'S')
    {

      flag1 = 0;
      flag2 = 0;

      digitalWrite( Motor1A, LOW);
      analogWrite( Motor1B, LOW);

      digitalWrite( Motor2A, LOW);
      analogWrite( Motor2B, LOW);

    }
    if( toSend == 'F' || toSend == 'G' || toSend == 'I')
    {
      if (flag1 != 1)
      {
        // ESTOS HARÁN QUE VAYA PARA ADELANTE EL CARRITO.
        flag1 = 1;
        digitalWrite( Motor1A, HIGH);
        analogWrite( Motor1B, 0 );
        digitalWrite( Motor2A, HIGH);
        analogWrite( Motor2B, 0 );
      }
    }
    if(toSend == 'B' || toSend == 'H' || toSend == 'J')
    {
      if(flag1 != 2)
      {
        // ESTOS HARAN LA REVERSA DEL CARRITO.
        flag1 = 2;
        digitalWrite( Motor1B, HIGH);
        analogWrite( Motor1A, 0 );
        digitalWrite( Motor2B, HIGH);
        analogWrite( Motor2A, 0 );
      }
    }
    if(toSend == 'L' || toSend == 'G' || toSend == 'H')
    {
      if(flag2 != 1)
      {
        // ESTOS HARAN QUE GIRE HACIA LA IZQUIERDA.
        flag2 = 1;
        digitalWrite( Motor2B, HIGH);
        analogWrite( Motor2A, 0 );
        digitalWrite( Motor1A, HIGH);
        analogWrite( Motor1B, 0 );
      }
    }
    else
    if(toSend == 'R' || toSend == 'I' || toSend == 'J')
    {
      if(flag2 != 2)
      {
        // ESTOS HARÁN QUE GIRE HACIA LA DERECHA.
        flag2 = 2;
        digitalWrite( Motor1B, HIGH);
        analogWrite( Motor2A, 0 );
        digitalWrite( Motor2A, HIGH);
        analogWrite( Motor2B, 0 );
      }
    }
    else
    {
      if(flag2 != 3)
      {

```

```

flag2 = 3;
digitalWrite ( Motor2A, LOW);
analogWrite ( Motor2B, LOW);
digitalWrite ( Motor2B, LOW);
analogWrite ( Motor2A, LOW);
}
}
}
}

```

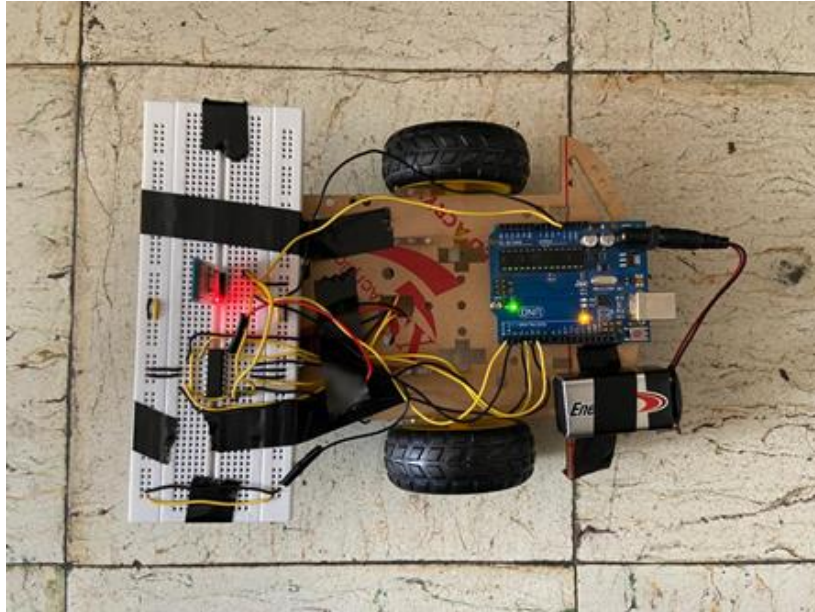


Fig 3. Carro funcional

Semana 3: 12 de mayo del 2022

Finalmente decidimos usar una tarjeta arduino ESP 8266 para controlar el carro mediante wifi, la implementación fue exitosa puesto el carro se manejó mediante control en página web ubicado en la ip local generada y el puerto 10.

Con respecto al código lo primero que se incluye es la librería proporciona las rutinas específicas wifi de ESP8266 a las que llamamos para conectarse a la red.

```
#include <ESP8266WiFi.h>
```

A continuación, se agregan los datos necesarios para realizar la conexión

```

const char* ssid  = "Rickys_home";    // SSID
const char* password = "12345678";    // Password
const char* host = "192.168.20.85";    // Dirección IP local o remota, del Servidor Web
const int  port = 80;                  // Puerto, HTTP es 80 por defecto, cambiar si es necesario.
const int  watchdog = 2000;           // Frecuencia del Watchdog
unsigned long previousMillis = millis();

```


Y se le dieron los siguientes nombres a los puertos de la ESP8266 los que están conectados los sensores de velocidad

```
#define BUTTON 12 //D6  
#define BUTTON2 14 //D5
```

Posteriormente se definieron los nombres para los puertos de la ESP8266 a los que están conectados los motores y que permitirá controlarlos.

```
//MOTOR DERECHA  
int OUTPUT4= 0; //D1  
int OUTPUT3= 2; //D2  
//MOTOR IZQUIERDA  
int OUTPUT2= 5;  
int OUTPUT1= 4;
```

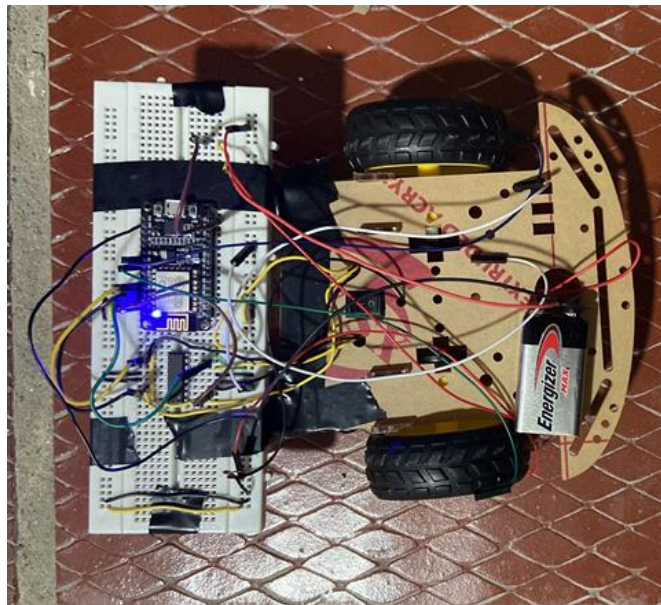


Fig 4. Carro usando la tarjeta arduino ESP 8266

Semana 4: 19 de Mayo del 2022

Se implementaron dos sensores de velocidad usando específicamente el LM393, se realizaron las modificaciones en el código y las respectivas conexiones físicas con el arduino y la protoboard.

Inicialmente se definen los puertos de la tarjeta ESP8266 en los cuales se conectan los sensores de velocidad

```
#define BUTTON 12 //D6  
#define BUTTON2 14 //D5
```

A continuación, se inicializaron las variables en 0 que permitirán llevar el conteo de las veces que el sensor detecte movimiento mediante las ranuras del disco codificado.

```
char dato;
volatile int DER=0;
volatile int IZQ=0;
```

Y finalmente se realizaron las interrupciones que permitirán realizar el conteo cada vez que los sensores detectan la rotación del disco codificado, este valor guardado en las funciones IZQ y DER permitirán controlar la distancia recorrida por el carro y los ángulos de giro que se necesiten.

```
//funciones de la interrupción
void IRAM_ATTR RUEDADER(){
    DER++; //INCREMENTAR +1
}

void IRAM_ATTR RUEDA IZQ(){
    IZQ; //INCREMENTAR +1
}
```

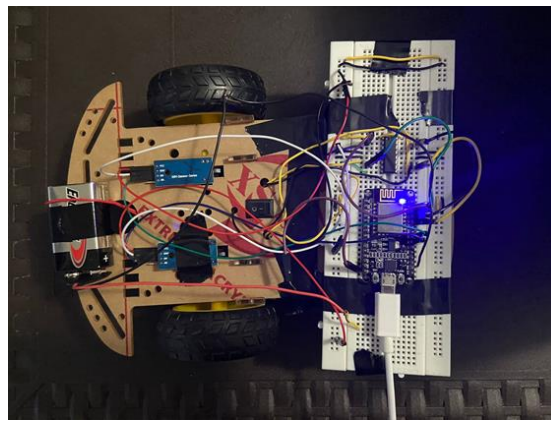


Fig 5 . Carro implementado con los sensores de velocidad herradura Lm393

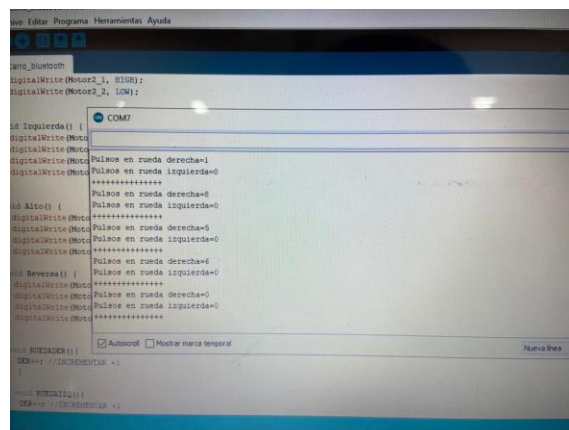


Fig 6. Prueba de los sensores de velocidad en arduino

Se comprobó el correcto funcionamiento de los sensores puesto que el monitor serial muestra las revoluciones que realizan las ruedas del carro cada segundo.

Semana 5: 26 de mayo del 2022

Se logró realizar la comunicación entre arduino, la base de datos y java que permitirá posteriormente enviar la ruta generada desde la aplicación en java hasta la tarjeta ESP 8266 mediante wifi, para ello se utilizó phpMyAdmin.

Inicialmente se configura la estructura de la base de datos de la siguiente manera.

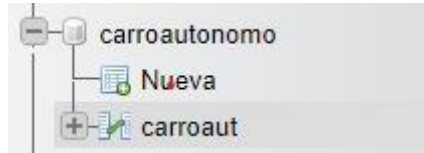


Fig 7. Configuración de la base de datos primera parte

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1 Id	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
<input type="checkbox"/>	2 Enable	int(11)			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	3 Ruta	varchar(100)	latín1_swedish_ci		No	Ninguna			Cambiar Eliminar Más

Fig 8. Configuración de la base de datos primera parte

.Cómo se muestra a continuación para probar que se pudo llevar a cabo la comunicación enviamos la orden ADELANTE la cual se recibió correctamente y la ejecutaba correctamente cada cierto periodo de tiempo.

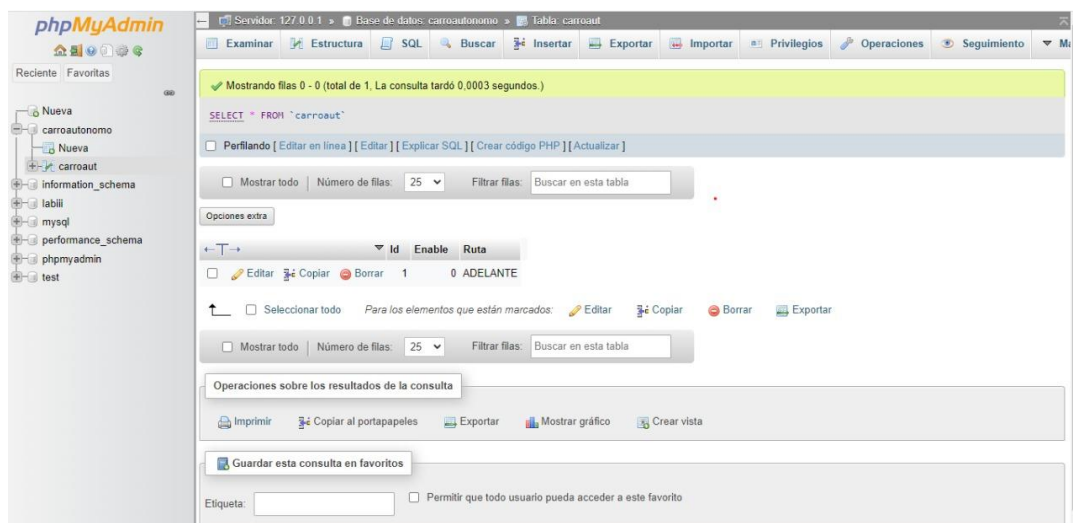


Fig 9. Prueba de funcionamiento de la base de datos

Semana 6: 2 de junio del 2022

Para la construcción de la interfaz que permitirá al usuario ingresar la ruta que el auto recorrerá se utilizó el siguiente código compuesto por 3 clases.

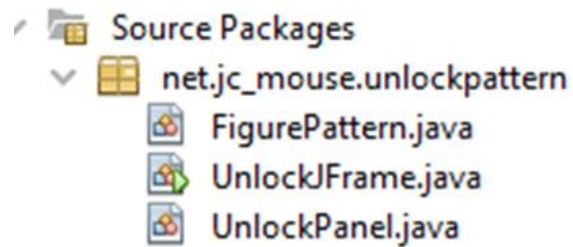


Fig 10. Estructura de la interfaz de java

FigurePattern: Es la figura circular que forma la matriz 3x3

```

1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to
edit this template
4   */
5  package net.jc_mouse.unlockpattern;
6  import java.awt.BasicStroke;
7  import java.awt.Color;
8  import java.awt.Graphics2D;
9  import java.awt.Point;
10 import java.awt.Rectangle;
11 import java.awt.geom.Ellipse2D;
12 /**
13  * @web https://www.jc-mouse.net/
14  * @author Mouse
15  */
16 public class FigurePattern {
17     //identificador/valor de objeto (NO debe repetirse)
18     private int key;
19     //coordenadas
20     private int x          = 0;
21     private int y          = 0;
22     //bandera
23     private boolean selected = false;
24     //area sensible a eventos del raton
25     private Rectangle area   = new Rectangle();
26
27     /**Constructor de clase */
28     public FigurePattern() {}
29
30     /**
31      * Constructor de clase
32      * @param k clave de celda
33      * @param x coordenada X de objeto
34      * @param y coordenada Y de objeto
35      */
36     public FigurePattern(int k, int x, int y){
37         this.key = k;
38         this.x = x;
39         this.y = y;
40         //area interna del circulo
41         area.setBounds(x+10, y+10 , 60, 60);
42     }

```

```

43
44     public void draw(Graphics2D g2){
45         //fondo
46         g2.setColor( new Color(0,0,0) );
47         g2.fill( new Ellipse2D.Double(x+10, y+10 , 60, 60 ) );
48         //borde exterior
49         g2.setStroke(new BasicStroke( 4 ));
50         g2.setColor( (selected)?new Color(0,204,0):new
Color(153,153,153) );
51         g2.draw( new Ellipse2D.Double(x+10, y+10 , 60, 60 ) );
52         //circulo interno
53         g2.setColor( new Color(255,255,255) );
54         g2.fill( new Ellipse2D.Double(x+30, y+30 , 20, 20 ) );
55     }
56
57     public int getKey() {
58         return key;
59     }
60
61     public void setKey(int key) {
62         this.key = key;
63     }
64
65     public boolean isSelected() {
66         return selected;
67     }
68
69     public void setSelected(boolean selected) {
70         this.selected = selected;
71     }
72
73     public Rectangle getArea() {
74         return area;
75     }
76
77     public void setArea(Rectangle area) {
78         this.area = area;
79     }
80
81     /**
82      * retorna el punto medio de la figura
83      * @return Point
84      */
85     public Point getCenterPoint(){
86         Point p = new Point( x+40, y+40);
87         return p;
88     }
89
90     /**
91      * Metodo statico que determina si un punto se encuentra dentro un
area rectangular
92      * @param r rectangle
93      * @param p point
94      * @return boolean
95      */
96     public static boolean insideArea(Rectangle r,Point p) {
97         return r.contains(p);
98     }
99
100 }//end:FigurePattern

```

UnlockPanel: Un JPanel donde pintar las figuras circulares e implementar los eventos del ratón para pintar el patrón de desbloqueo.

Esta clase que descende de un JPanel será donde se pinte la matriz con las 9 figuras circulares, así mismo implementa métodos para que el usuario pueda arrastrando el cursor del mouse dibujar el patrón de desbloqueo.

El método checkUnlockPattern() se encarga de verificar si el patrón introducido por el usuario es correcto, de ser cierto, abrirá un nuevo JFrame (tu aplicación) que en nuestro ejemplo se encuentra en el paquete «demo» y se llama «AppDemoJFrame», caso contrario, de ser un patrón incorrecto, muestra un mensaje de error «Access denied».

```
import java.awt.event.MouseListener;
import java.awt.event.MouseMotionListener;
import java.awt.geom.Rectangle2D;
import java.util.ArrayList;
import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JPanel;
import net.jc_mouse.demo.AppDemoJFrame;

/**
 * @web https://www.jc-mouse.net/
 * @author Mouse
 */
public class UnlockPanel extends JPanel implements MouseListener, MouseMotionListener {

    private JFrame parent;
    private String myKey="125895";//patron de desbloqueo
    //array para las 9 figuras que forman la matriz
    private ArrayList<FigurePattern> figurePatternArr = new ArrayList();
    //array para las figuras que conforman el patron desbloqueo
    private ArrayList<FigurePattern> pattern= new ArrayList();
    //String para guardar el patron generado por el usuario
    private String password = "";
    //para mostrar imagen de "acceso denegado"
    private final Image image = new ImageIcon(getClass().getResource("/net/jc_mouse/unlockpattern/access_denied.png")).getImage();
    private boolean showError=false;

    /**
     * Constructor de clas
     * @param parent JFrame que contiene el JPanel
     */
    public UnlockPanel(JFrame parent){
        this.parent = parent;
        //se agregan 9 circulos al contenedor
        figurePatternArr.add(new FigurePattern(1,0,0));
        figurePatternArr.add(new FigurePattern(2,80,0));
        figurePatternArr.add(new FigurePattern(3,160,0));

        figurePatternArr.add(new FigurePattern(4,0,80));
        figurePatternArr.add(new FigurePattern(5,80,80));
        figurePatternArr.add(new FigurePattern(6,160,80));

        figurePatternArr.add(new FigurePattern(7,0,160));
        figurePatternArr.add(new FigurePattern(8,80,160));
        figurePatternArr.add(new FigurePattern(9,160,160));

        //Eventos del raton
        UnlockPanel.this.addMouseListener( UnlockPanel.this );
        UnlockPanel.this.addMouseMotionListener( UnlockPanel.this);
    }

    @Override
    public void paintComponent(Graphics g){
        Graphics2D g2 =(Graphics2D) g;
        g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,RenderingHints.VALUE_ANTIALIAS_ON);
        g2.setColor( new Color(0,0,0) );
        g2.fill(new Rectangle2D.Double(0,0,getWidth(),getHeight()));
    }
}
```

```

//dibuja las lineas del patron de desbloqueo entre los circulos
Point point = null;
for(FigurePattern figurePattern: pattern){
    g2.setStroke( new BasicStroke( 20 ) );
    g2.setColor( new Color(204,204,204) );
    if(point!=null)
        g2.drawLine(point.x, point.y, figurePattern.getCenterPoint().x, figurePattern.getCenterPoint().y);
    point = figurePattern.getCenterPoint();
}

//dibuja los circulos para los patrones de desbloqueo
if(figurePatternArr!=null)
    figurePatternArr.stream().forEach((b) -> {
        b.draw(g2);
    });

//muestra error
if(showError){
    g2.drawImage(image, (getWidth()-220)/2, (getHeight()-93)/2, 220, 93, null);
}
}

@Override
public void mouseClicked(MouseEvent e) {}

@Override
public void mousePressed(MouseEvent e) {}

@Override
public void mouseReleased(MouseEvent e) {
    checkUnlockPattern();
}

@Override
public void mouseEntered(MouseEvent e) {}

@Override
public void mouseExited(MouseEvent e) {}

@Override
public void mouseDragged(MouseEvent e) {
    //cuando se arrastra el cursor del mouse
    figurePatternArr.stream().filter((p) -> (
        FigurePattern.insideArea(p.getArea(),e.getPoint()) ))forEach((p) -> {
        if(this.pattern.isEmpty()){//Si esta vacio se añade objeto
            pattern.add(p);//añade a array
            p.setSelected(true);//cambia color
            password +=p.getKey();//concatena valor de figura
        }else{//Si ya existen objetos en el array
            //se comprueba que objeto nuevo no se repita con el ultimo añadido
            if(pattern.get(pattern.size()-1).getKey() != p.getKey()){
                pattern.add(p);//añade a array
                p.setSelected(true);//cambia color
                password +=p.getKey(); //concatena valor de figura
            }
        }
    });
    repaint();
}

@Override
public void mouseMoved(MouseEvent e) { showError=false; repaint();}

/**
 * metodo para comprobar que patron de desbloqueo es correcto
 * de ser asi -> abre nuevo formulario
 * caso contrario -> muestra mensaje de error
 */
public void checkUnlockPattern(){
    if(password.equals(myKey)){//si patron de desbloqueo es correcto
        parent.setVisible(false);//se oculta
        //se muestra aplicacion cliente
        AppDemoJFrame app = new AppDemoJFrame();
        app.setVisible(true);
    }else{

```

```
//end
```

Onlooker name: Elor name principaladnas se maestra todo el sistema

}

Mediante la siguiente matriz 3x3 el usuario podrá generar la ruta que recorre el carro.

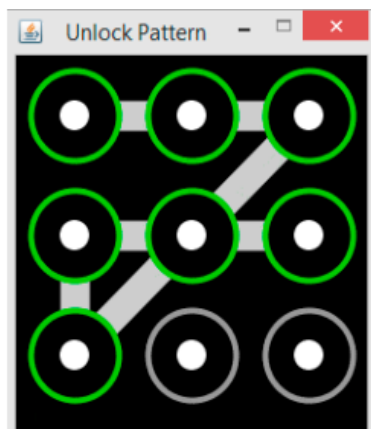


Fig 11. Patrón obtenido de internet

Hasta el momento se logra detectar un vector con el orden de las posiciones designadas para cada círculo que compone el panel de la siguiente manera:

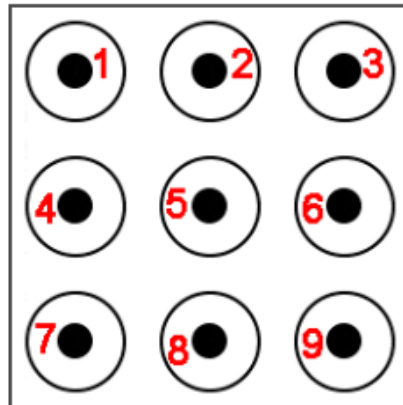


Fig 12. Definición de los puntos

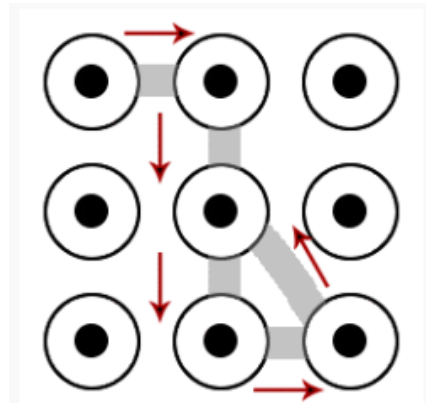


Fig 13. Vector de puntos

Por lo que el vector de ruta enviada en este caso será: [1 2 5 8 9 5]

Semana 7: 16 de junio del 2022

Con el objetivo de que el carro pueda hacer un recorrido más complejo se amplió a 4x4 la matriz generadora de ruta logrando así el siguiente resultado.



Fig 14. Interfaz modificada 4*4 para el prototipo

Para ello se tuvo que aumentar en la clase UnlockJFrame el tamaño de la interfaz.

```

24 private void initComponents() {
25     setTitle("Unlock Pattern");
26     setResizable(false);
27     setSize(new Dimension(340,360));
28     setPreferredSize(new Dimension(340,360));
29     unlockPanel = new UnlockPanel(this);
30     setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
31     getContentPane().add(unlockPanel);
32     pack();
33 }

```

y en la clase UnlockPanel agregamos las nuevas componentes de las matriz 4x4

```

76 public UnlockPanel(JFrame parent){
77     this.parent = parent;
78     //se agregan 9 circulos al contenedor
79     figurePatternArr.add(new FigurePattern(11,0,0));
80     figurePatternArr.add(new FigurePattern(12,80,0));
81     figurePatternArr.add(new FigurePattern(13,160,0));
82     figurePatternArr.add(new FigurePattern(14,240,0));
83
84     figurePatternArr.add(new FigurePattern(15,0,80));
85     figurePatternArr.add(new FigurePattern(16,80,80));
86     figurePatternArr.add(new FigurePattern(17,160,80));
87     figurePatternArr.add(new FigurePattern(18,240,80));
88
89     figurePatternArr.add(new FigurePattern(19,0,160));
90     figurePatternArr.add(new FigurePattern(20,80,160));
91     figurePatternArr.add(new FigurePattern(21,160,160));
92     figurePatternArr.add(new FigurePattern(22,240,160));
93
94     figurePatternArr.add(new FigurePattern(23,0,240));
95     figurePatternArr.add(new FigurePattern(24,80,240));
96     figurePatternArr.add(new FigurePattern(25,160,240));
97     figurePatternArr.add(new FigurePattern(26,240,240));

```

Debido a que el propósito es lograr que el carro recorra una ruta tomando como punto de partida un mismo punto en la matriz generadora de ruta y finalizando el recorrido siempre al momento en el que llega a un punto de destino se programó el generador de ruta para que solo se envíe la ruta cuando se cumplen las dos condiciones mencionadas anteriormente.

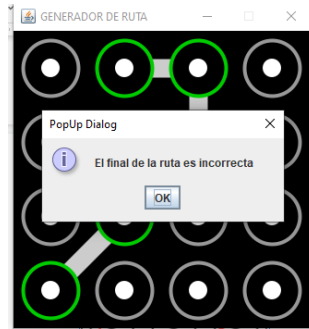


Fig 15. Punto final incorrecto

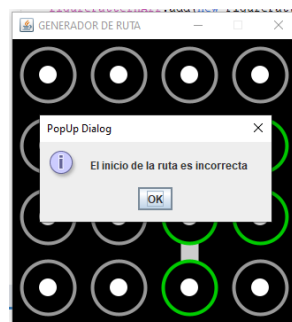


Fig 16. Punto inicial incorrecto

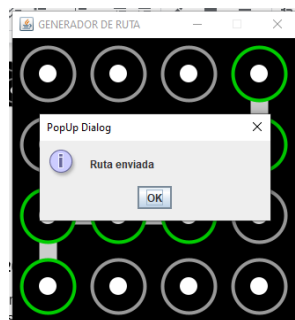


Fig 17. Ruta correctamente enviada

Para ello se modificó el siguiente método del código.

```

180     public void checkUnlockPattern () {
181         PreparedStatement ps;
182         String sql;
183         Ruta=password.toCharArray();
184         Longitud=Ruta.length;
185         Longitud--;
186         Finall = Character.getNumericValue(Ruta[Longitud-1]);

```

```

187         Final2 = Character.getNumericValue(Ruta[Longitud]);
188         Inicial1 = Character.getNumericValue(Ruta[0]);
189         Inicial2 = Character.getNumericValue(Ruta[1]);
190         if(Inicial1==2 && Inicial2==3){
191             if(Final1==1 && Final2==4)
192             {
193
194                 JOptionPane.showMessageDialog(null,
195                     "Ruta enviada",
196                     "PopUp Dialog",
197                     JOptionPane.INFORMATION_MESSAGE);
198
199                 taru=password;
200
201                 taruconvertida = conversorDeRuta(taru, 4);
202                 RutaLista();
203
204
205
206                 try {
207                     Class.forName("com.mysql.cj.jdbc.Driver");
208                 } catch (ClassNotFoundException ex) {
209
210                     Logger.getLogger(UnlockPanel.class.getName()).log(Level.SEVERE, null, ex);
211                 }
212
213                 try {
214
215                     con=DriverManager.getConnection(url,usuario,clave);
216                     stmt=con.createStatement();
217
218                     stmt.executeUpdate("DELETE FROM carroaut");
219
220                     for(int i=0;i<EnviarRuta.size();i++)
221                     {
222                         sql = "INSERT INTO carroaut
223                             VALUES(?,?,?)";
224
225                         ps = con.prepareStatement(sql);
226                         id=i+1;
227                         ps.setInt(1,id);
228                         ps.setInt(2, 0);
229                         ps.setString(3,EnviarRuta.get(i));
230                         ps.executeUpdate();
231                         //stmt.executeUpdate("INSERT INTO
232                         carroaut VALUES(null,0,i)");
233                         System.out.println(EnviarRuta.get(i));
234                     }
235
236                 } catch (SQLException ex) {
237
238                     Logger.getLogger(UnlockPanel.class.getName()).log(Level.SEVERE, null, ex);
239                 }
240
241                 } else{
242                     //showError=true;
243                     JOptionPane.showMessageDialog(null,
244                         "El final de la ruta es incorrecta",
245                         "PopUp Dialog",
246                         JOptionPane.INFORMATION_MESSAGE);
247                 }

```

```

243         } else{
244             JOptionPane.showMessageDialog(null,
245                 "El inicio de la ruta es incorrecta",
246                 "PopUp Dialog",
247                 JOptionPane.INFORMATION_MESSAGE);
248         }

```

Semana 8: 23 de junio del 2022

Se iniciaron las pruebas del movimiento del vehículo, mediante el código de arduino los encoders y los sensores de velocidad se logra controlar la distancia que recorrerá el carro en cada paso de la ruta y el ángulo de cada uno de los giros que realice, teniendo en cuenta se establece que el carro debe recorrer 38 cm en línea recta al recorrer 72 divisiones del encoder usado:

```

        if(secuencias>auxruta){
            if (ruta[auxruta].indexOf("ADELANTE")>0){
                if (IZQ>=72 && DER>=72){

```

Fig 18. Código Arduino

53,74 cm en diagonal al recorrer 57 divisiones del encoder

```

        }
    }else if (ruta[auxruta].indexOf("ADELANTEH")>0){

        if (IZQ>=57 && DER>=57){
            parar();

```

Fig 19. Código Arduino

45° con al recorrer 4 divisones del encoder

```

        }
    }else if (ruta[auxruta].indexOf("DERECHA45")>0){

        if (IZQ>=4 && DER>=4){

```

Fig 20. Código Arduino

90° con al recorrer 8 divisiones del encoder

```

        }
    }else if (ruta[auxruta].indexOf("DERECHA90")>0){

        if (IZQ>=8 && DER>=8){

```

Fig 21. Código Arduino

135° con al recorrer 12 divisiones del encoder

```
}else if (ruta[auxruta].indexOf("DERECHA135")>0){  
    if(IZQ>=12 && DER>=12){
```

Fig 22. Código Arduino

180° con al recorrer 16 divisiones del encoder

```
}else if (ruta[auxruta].indexOf("180")>0){  
    if(IZQ>=16 && DER>=16){
```

Fig 23. Código Arduino

Semana 9: 7 de Julio del 2022

Continuando con la fase de pruebas se programaron dos rutas paso a paso en java y se enviaron al carro mediante la base de datos, el carro recibió correctamente las rutas y ejecutó sin embargo se demoraba bastante el programa de arduino en procesarlas.

Semana 10: 14 de Julio del 2022

Debido a que en el método implementado anteriormente era necesario programar cada ruta pasa por paso lo cual era ineficiente debido a que son cientos las posibles combinaciones que se pueden generar con una matriz 4x4 se requirió implementar un nuevo método el cual recibe el vector que genera en primera instancia la matriz generadora de ruta y transforma la ruta de tal forma que el programa solo necesita saber el punto de inicial y final de cada paso de la ruta de esta forma se evita tener que programar cada una de las rutas puesto que el código creado es capaz de adaptarse a múltiples casos.

```
259 public void RutaLista(){  
260     RutaNueva=taruconvertida.toCharArray();  
261     for(int i=0; i < (taruconvertida.length()-1) ; i+=2) {  
262  
263         int a=Character.getNumericValue(RutaNueva[i])*10;  
264         int b=Character.getNumericValue(RutaNueva[i+1]);  
265         int c=a+b;  
266         System.out.println(c);  
267         switch(c) {  
268             case 51:
```



```

269     case 73:
270     case 15:
271     case 37: EnviarRuta.add("ADELANTE");break;
272
273     case 62:
274     case 84:
275     case 26:
276     case 48: EnviarRuta.add("AVANZANDO");break;
277
278     case 52:
279     case 74:
280     case 16:
281     case 38: EnviarRuta.add("DERECHA45");EnviarRuta.add("AVANZANDO");break;
282
283     case 63:
284     case 85:
285     case 27:
286     case 41: EnviarRuta.add("DERECHA45");EnviarRuta.add("ADELANTE");break;
287
288     case 53:
289     case 75:
290     case 17:
291     case 31: EnviarRuta.add("RIGHT90");EnviarRuta.add("ADELANTE");break;
292
293     case 64:
294     case 86:
295     case 28:
296     case 42: EnviarRuta.add("RIGHT90");EnviarRuta.add("AVANZANDO");break;
297
298
299     case 54:
300     case 76:
301     case 18:
302     case 32: EnviarRuta.add("CERTO135");EnviarRuta.add("AVANZANDO");break;
303
304     case 65:
305     case 87:
306     case 21:
307     case 43: EnviarRuta.add("CERTO135");EnviarRuta.add("ADELANTE");break;
308
309     case 58:
310     case 72:
311     case 14:
312     case 36: EnviarRuta.add("IZQUIERDA45");EnviarRuta.add("AVANZANDO");break;
313
314     case 61:
315     case 83:
316     case 25:
317     case 47: EnviarRuta.add("DERECHA45");EnviarRuta.add("ADELANTE");break;
318
319     case 57:
320     case 71:
321     case 13:
322     case 35: EnviarRuta.add("LEFT90");EnviarRuta.add("ADELANTE");break;
323
324     case 68:
325     case 82:
326     case 24:
327     case 46: EnviarRuta.add("LEFT90");EnviarRuta.add("AVANZANDO");break;
328
329     case 56:
330     case 78:
331     case 12:
332     case 34: EnviarRuta.add("DEIXE135");EnviarRuta.add("AVANZANDO");break;
333
334     case 67:
335     case 81:
336     case 23:
337     case 45: EnviarRuta.add("DEIXE135");EnviarRuta.add("ADELANTE");break;
338
339     case 66:
340     case 88:
341     case 22:
342     case 44: EnviarRuta.add("180");EnviarRuta.add("AVANZANDO");break;
343
344     case 55:
345     case 77:
346     case 11:
347     case 33: EnviarRuta.add("180");EnviarRuta.add("ADELANTE");break;
348
349     }
350 }
351 }
352
353 public String conversorDeRuta(String ruta, int columnas) {
354     String rutaConvertida = "5";

```

```

355 char[] aCaracteres = ruta.toCharArray();
356
357 for (int i = 0; i < ruta.length()-2; i+=2) {
358     int a = aCaracteres[i]*10+aCaracteres[i+1]-(aCaracteres[i+2]*10+aCaracteres[i+3]);
359
360     if(a==1) {
361         rutaConvertida += "7";if(i+1<ruta.length()-3){rutaConvertida+="3";}
362     }else if(a==1) {
363         rutaConvertida += "3";if(i+1<ruta.length()-3){rutaConvertida+="7";}
364     }else if(a==columnas) {
365         rutaConvertida += "5";if(i+1<ruta.length()-3){rutaConvertida+="1";}
366     }else if(a==columnas) {
367         rutaConvertida += "1";if(i+1<ruta.length()-3){rutaConvertida+="5";}
368     }else if(a==columnas-1) {
369         rutaConvertida += "4";if(i+1<ruta.length()-3){rutaConvertida+="8";}
370     }else if(a==columnas+1) {
371         rutaConvertida += "8";if(i+1<ruta.length()-3){rutaConvertida+="4";}
372     }else if(a==columnas+1) {
373         rutaConvertida += "6";if(i+1<ruta.length()-3){rutaConvertida+="2";}
374     }else if(a==columnas-1) {
375         rutaConvertida += "2";if(i+1<ruta.length()-3){rutaConvertida+="6";}
376     }
377 }
378 return rutaConvertida;
379 }
380
381 }//end

```

Semana 11: 21 de Julio del 2022

Se realizaron los últimos ajustes en las variables del código para definir la distancia que el carro recorre en línea recta y diagonal con el objetivo de que la ruta que recorre el carro coincida con la medida de las baldosas del piso de prueba para poder observar mejor que el carro efectivamente recorre la ruta que se le asignó como dichas baldosas miden 38cmX38xm el carro se configuró para recorrer 38 cm hacia adelante y 53.74 cm en diagonal, para ello se modificaron los valores de las variables IZQ y DER (que controlan la distancia recorrida y el ángulo de giro) de la siguiente sección del código de esta manera.

```

if(rutaC.toInt() != 1){
    if (consul==1){
        if(secuencias>auxruta){
            if (ARUTA[auxruta]=="ADELANTE"){
                if(IZQ>=60 && DER>=60){
                    parar();
                    auxruta++;
                    delay(200);
                    IZQ=0;
                    DER=0;
                }else{
                    if(IZQ < DER || DER>=60){
                        adelanteruedaizquierda();

                    }else if(IZQ > DER || IZQ>=60){
                        adelanteruedaderecha();
                    }else if(IZQ == DER){
                        adelante();
                        Serial.print("Pulsos en rueda derecha=");
                        Serial.println(DER);
                        Serial.print("Pulsos en rueda izquierda=");
                        Serial.println(IZQ);
                        Serial.println("+++++++");
                    }
                }
            }else if (ARUTA[auxruta]=="AVANZANDO"){

                if(IZQ>=57 && DER>=57){
                    parar();
                    auxruta++;
                    delay(500);
                    IZQ=0;
                    DER=0;
                }else{
                    if(IZQ < DER || DER>=57){
                        adelanteruedaizquierda();
                    }else if (IZQ > DER || IZQ>=57){
                        adelanteruedaderecha();
                    }
                }
            }
        }
    }
}

```

```

    }else{

        Serial.print("Pulsos en rueda derecha=");
        Serial.println(DER);
        Serial.print("Pulsos en rueda izquierda=");
        Serial.println(IZQ);
        Serial.println("+++++++");
        adelante();
    }
}

}else if (ARUTA[auxruta]=="DERECHA45"){

    if(IZQ>=9 && DER>=9){
        parar();
        auxruta++;
        delay(500);
        IZQ=0;
        DER=0;
    }else{
        if(IZQ < DER || DER>=9){
            adelanteruedaizquierda();
        }else if (IZQ > DER || IZQ>=9){
            atrasruedaderecha();
        }else{

            Serial.print("Pulsos en rueda derecha=");
            Serial.println(DER);
            Serial.print("Pulsos en rueda izquierda=");
            Serial.println(IZQ);
            Serial.println("+++++++");
            derecha();
        }
    }
}

}else if (ARUTA[auxruta]=="RIGHT90"){

    if(IZQ>=19 && DER>=19){
        parar();
        auxruta++;
        delay(500);
        IZQ=0;
        DER=0;
    }else{
        if(IZQ < DER || DER>=19){
            adelanteruedaizquierda();
        }else if (IZQ > DER || IZQ>=19){
            atrasruedaderecha();
        }else{

            Serial.print("Pulsos en rueda derecha=");
            Serial.println(DER);
            Serial.print("Pulsos en rueda izquierda=");
            Serial.println(IZQ);
            Serial.println("+++++++");
            derecha();
        }
    }
}

}else if (ARUTA[auxruta]=="CERTO135"){

    if(IZQ>=27 && DER>=27){
        parar();
        auxruta++;
        delay(500);
        IZQ=0;
        DER=0;
    }else{
        if(IZQ < DER || DER>=27){
            adelanteruedaizquierda();
        }else if (IZQ > DER || IZQ>=27){
            atrasruedaderecha();
        }else{

            Serial.print("Pulsos en rueda derecha=");
            Serial.println(DER);
            Serial.print("Pulsos en rueda izquierda=");
            Serial.println(IZQ);
            Serial.println("+++++++");
            derecha();
        }
    }
}

}else if (ARUTA[auxruta]=="IZQUIERDA45"){

    if(IZQ>=9 && DER>=9){
        parar();
        auxruta++;
        delay(500);
    }
}

```

```

        IZQ=0;
        DER=0;
    }else{
        if (IZQ < DER || DER>=9){
            atrasruedaizquierda();
        }else if (IZQ > DER || IZQ>=9){
            adelanteruedaderecha();
        }else{

            Serial.print("Pulsos en rueda derecha=");
            Serial.println(DER);
            Serial.print("Pulsos en rueda izquierda=");
            Serial.println(IZQ);
            Serial.println("+++++++");
            izquierda();
        }
    }

}

}else if (ARUTA[auxruta]=="LEFT90"){

    if (IZQ>=18 && DER>=18){
        parar();
        auxruta++;
        delay(500);
        IZQ=0;
        DER=0;
    }else{
        if (IZQ < DER || DER>=18){
            atrasruedaizquierda();
        }else if (IZQ > DER || IZQ>=18){
            adelanteruedaderecha();
        }else{

            Serial.print("Pulsos en rueda derecha=");
            Serial.println(DER);
            Serial.print("Pulsos en rueda izquierda=");
            Serial.println(IZQ);
            Serial.println("+++++++");
            izquierda();
        }
    }

}

}else if (ARUTA[auxruta]=="DEIXEI135"){

    if (IZQ>=27 && DER>=27){
        parar();
        auxruta++;
        delay(500);
        IZQ=0;
        DER=0;
    }else{
        if (IZQ < DER || DER>=27){
            atrasruedaizquierda();
        }else if (IZQ > DER || IZQ>=27){
            adelanteruedaderecha();
        }else{

            Serial.print("Pulsos en rueda derecha=");
            Serial.println(DER);
            Serial.print("Pulsos en rueda izquierda=");
            Serial.println(IZQ);
            Serial.println("+++++++");
            izquierda();
        }
    }

}

}else if (ARUTA[auxruta]=="180"){

    if (IZQ>=36 && DER>=36){
        parar();
        auxruta++;
        delay(500);
        IZQ=0;
        DER=0;
    }else{
        if (IZQ < DER || DER>=36){
            atrasruedaizquierda();
        }else if (IZQ > DER || IZQ>=36){
            adelanteruedaderecha();
        }else{

            Serial.print("Pulsos en rueda derecha=");
            Serial.println(DER);
            Serial.print("Pulsos en rueda izquierda=");
            Serial.println(IZQ);
            Serial.println("+++++++");
            izquierda();
        }
    }

}

```

```

    }
}

```

Se presentó el prototipo funcional y para ello el docente ingreso la siguiente ruta en la interfaz generadora de ruta en java.



Fig 24. Ruta generada para la presentación del prototipo funcional

Semana 12: 28 de Julio del 2022

Con el propósito de disminuir el tiempo que tarda el carro en ejecutar la ruta enviada se optó por pasar el método que se usaba en java para controlar el carro al código de arduino.

```

if(rutaC.toInt() != 1){
    if (consul==1 && HacerRuta==0){
        char ruta[rutaC.length()+1];
        ruta.toCharArray(ruta, rutaC.length()+1);
        delay(2000);

        for(int i=1; i < (rutaC.length()-1) ; i+=2) {

            int a=String(ruta[i]).toInt()*10;
            int b=String(ruta[i+1]).toInt();
            int c=a+b;
            Serial.println(c);
            switch(c) {
                case 51:
                case 73:
                case 15:
                case 37: ARUTA[HacerRuta]="ADELANTE";HacerRuta+=1;break;

                case 62:
                case 84:
                case 26:
                case 48: ARUTA[HacerRuta]="AVANZANDO";HacerRuta+=1;break;

                case 52:
                case 74:
                case 16:
                case 38: ARUTA[HacerRuta]="DERECHA45";ARUTA[HacerRuta+1]="AVANZANDO";HacerRuta+=2;break;

                case 63:

```

```

case 85:
case 27:
case 41: ARUTA[HacerRuta]="DERECHA45";ARUTA[HacerRuta+1]="ADELANTE";HacerRuta+=2;break;

case 53:
case 75:
case 17:
case 31: ARUTA[HacerRuta]="RIGHT90";ARUTA[HacerRuta+1]="ADELANTE";HacerRuta+=2;break;

case 64:
case 86:
case 28:
case 42: ARUTA[HacerRuta]="RIGHT90";ARUTA[HacerRuta+1]="AVANZANDO";HacerRuta+=2;break;

case 54:
case 76:
case 18:
case 32: ARUTA[HacerRuta]="CERTO135";ARUTA[HacerRuta+1]="AVANZANDO";HacerRuta+=2;break;

case 65:
case 87:
case 21:
case 43: ARUTA[HacerRuta]="CERTO135";ARUTA[HacerRuta+1]="ADELANTE";HacerRuta+=2;break;

case 58:
case 72:
case 14:
case 36:
ARUTA[HacerRuta]="IZQUIERDA45";ARUTA[HacerRuta+1]="AVANZANDO";HacerRuta+=2;break;

case 61:
case 83:
case 25:
case 47: ARUTA[HacerRuta]="IZQUIERDA45";ARUTA[HacerRuta+1]="ADELANTE";HacerRuta+=2;break;

case 57:
case 71:
case 13:
case 35: ARUTA[HacerRuta]="LEFT90";ARUTA[HacerRuta+1]="ADELANTE";HacerRuta+=2;break;

case 68:
case 82:
case 24:
case 46: ARUTA[HacerRuta]="LEFT90";ARUTA[HacerRuta+1]="AVANZANDO";HacerRuta+=2;break;

case 56:
case 78:
case 12:
case 34: ARUTA[HacerRuta]="DEIXEI135";ARUTA[HacerRuta+1]="AVANZANDO";HacerRuta+=2;break;

case 67:
case 81:
case 23:
case 45: ARUTA[HacerRuta]="DEIXEI135";ARUTA[HacerRuta+1]="ADELANTE";HacerRuta+=2;break;

case 66:
case 88:
case 22:
case 44: ARUTA[HacerRuta]="180";ARUTA[HacerRuta+1]="AVANZANDO";HacerRuta+=2;break;

case 55:
case 77:
case 11:
case 33: ARUTA[HacerRuta]="180";ARUTA[HacerRuta+1]="ADELANTE";HacerRuta+=2;break;

}
}
}

```


CÓDIGO FINAL DE ARDUINO

```
#include <ESP8266WiFi.h>

//const char* ssid = "StivenMolina"; // SSID
//const char* password = "12345678"; // Password
//const char* ssid = "PISO-1"; // SSID
//const char* password = "torreal000"; // Password
//const char* host = "192.168.1.108"; // Dirección IP local o remota, del Servidor Web
const char* ssid = "StivenMolina"; // SSID
const char* password = "12345678"; // Password
const char* host = "192.168.104.134"; // Dirección IP local o remota, del Servidor Web
const int port = 80; // Puerto, HTTP es 80 por defecto, cambiar si es necesario.
const int watchdog = 2000; // Frecuencia del Watchdog
unsigned long previousMillis = millis();

#define BUTTON 12 //D6
#define BUTTON2 14 //D5

String ARUTA[1000];
String line;
String ADELANTE="ADELANTE";
int consul;
int auxruta=0;
int secuencias=0;
int HacerRuta=0;
String rutaC;

char dato;
volatile int DER=0;
volatile int IZQ=0;

//MOTOR DERECHA
int OUTPUT4= 5; //D1
int OUTPUT3= 4; //D2
//MOTOR IZQUIERDA
int OUTPUT2= 0;
int OUTPUT1= 2;

//funciones de la interrupcion
void IRAM_ATTR RUEDADER(){
  DER++; //INCREMENTAR +1
}

void IRAM_ATTR RUEDAIZQ(){
  IZQ++; //INCREMENTAR +1
}

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

  pinMode(OUTPUT1, OUTPUT);
  pinMode(OUTPUT2, OUTPUT);
  pinMode(OUTPUT3, OUTPUT);
  pinMode(OUTPUT4, OUTPUT);
  pinMode(BUTTON, INPUT);
  pinMode(BUTTON2, INPUT);

  digitalWrite (BUTTON, LOW);
  digitalWrite (BUTTON2, HIGH);

  Serial.println();
  Serial.print("CONECTANDO WIFI: ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi conectado");
  Serial.println("Dirección IP: ");
  Serial.println(WiFi.localIP());

  //pinMode(BUTTON, INPUT);
  attachInterrupt(BUTTON, RUEDADER, CHANGE); //INT0 EN PIN12
  //pinMode(BUTTON2, INPUT);
  attachInterrupt(BUTTON2, RUEDAIZQ, CHANGE); //INT0 EN PIN13

  consul=0;
  espera(5); //espera 5 segundos
}
```

```

void loop() {

  if(consul==0){
    consulta();
    Serial.print(line);
    consul=1;
    auxruta=0;
    Serial.print("Consulta terminada");
    IZQ=0;
    DER=0;
    rutaC=line;
    secuencias=rutaC.length();
  }

  if(rutaC.toInt() != 1){
    if (consul==1 && HacerRuta==0){
      char ruta[rutaC.length()+1];
      rutaC.toCharArray(ruta, rutaC.length()+1);
      delay(2000);

      for(int i=1; i < (rutaC.length()-1) ; i+=2) {

        int a=String(ruta[i]).toInt()*10;
        int b=String(ruta[i+1]).toInt();
        int c=a+b;
        Serial.println(c);
        switch(c) {
          case 51:
          case 73:
          case 15:
          case 37: ARUTA[HacerRuta]="ADELANTE";HacerRuta+=1;break;

          case 62:
          case 84:
          case 26:
          case 48: ARUTA[HacerRuta]="AVANZANDO";HacerRuta+=1;break;

          case 52:
          case 74:
          case 16:
          case 38: ARUTA[HacerRuta]="DERECHA45";ARUTA[HacerRuta+1]="AVANZANDO";HacerRuta+=2;break;

          case 63:
          case 85:
          case 27:
          case 41: ARUTA[HacerRuta]="DERECHA45";ARUTA[HacerRuta+1]="ADELANTE";HacerRuta+=2;break;

          case 53:
          case 75:
          case 17:
          case 31: ARUTA[HacerRuta]="RIGHT90";ARUTA[HacerRuta+1]="ADELANTE";HacerRuta+=2;break;

          case 64:
          case 86:
          case 28:
          case 42: ARUTA[HacerRuta]="RIGHT90";ARUTA[HacerRuta+1]="AVANZANDO";HacerRuta+=2;break;

          case 54:
          case 76:
          case 18:
          case 32: ARUTA[HacerRuta]="CERTO135";ARUTA[HacerRuta+1]="AVANZANDO";HacerRuta+=2;break;

          case 65:
          case 87:
          case 21:
          case 43: ARUTA[HacerRuta]="CERTO135";ARUTA[HacerRuta+1]="ADELANTE";HacerRuta+=2;break;

          case 58:
          case 72:
          case 14:
          case 36: ARUTA[HacerRuta]="IZQUIERDA45";ARUTA[HacerRuta+1]="AVANZANDO";HacerRuta+=2;break;

          case 61:
          case 83:
          case 25:
          case 47: ARUTA[HacerRuta]="IZQUIERDA45";ARUTA[HacerRuta+1]="ADELANTE";HacerRuta+=2;break;

          case 57:
          case 71:
          case 13:
          case 35: ARUTA[HacerRuta]="LEFT90";ARUTA[HacerRuta+1]="ADELANTE";HacerRuta+=2;break;

          case 68:
          case 82:
          case 24:

```

```

case 46: ARUTA[HacerRuta]="LEFT90";ARUTA[HacerRuta+1]="AVANZANDO";HacerRuta+=2;break;

case 56:
case 78:
case 12:
case 34: ARUTA[HacerRuta]="DEIXEI135";ARUTA[HacerRuta+1]="AVANZANDO";HacerRuta+=2;break;

case 67:
case 81:
case 23:
case 45: ARUTA[HacerRuta]="DEIXEI135";ARUTA[HacerRuta+1]="ADELANTE";HacerRuta+=2;break;

case 66:
case 88:
case 22:
case 44: ARUTA[HacerRuta]="180";ARUTA[HacerRuta+1]="AVANZANDO";HacerRuta+=2;break;

case 55:
case 77:
case 11:
case 33: ARUTA[HacerRuta]="180";ARUTA[HacerRuta+1]="ADELANTE";HacerRuta+=2;break;

    }
  }
}

if(rutaC.toInt() != 1){
  if (consul==1){
    if(secuencias>auxruta){
      if (ARUTA[auxruta]=="ADELANTE"){
        if (IZQ>=60 && DER>=60){
          parar();
          auxruta++;
          delay(200);
          IZQ=0;
          DER=0;
        }else{
          if (IZQ < DER || DER>=60){
            adelanteruedaizquierda();

          }else if (IZQ > DER || IZQ>=60){
            adelanteruedaderecha();
          }else if (IZQ == DER){
            adelante();
            Serial.print("Pulsos en rueda derecha=");
            Serial.println(DER);
            Serial.print("Pulsos en rueda izquierda=");
            Serial.println(IZQ);
            Serial.println("+++++++");
          }
        }
      }
    }else if (ARUTA[auxruta]=="AVANZANDO"){

      if (IZQ>=57 && DER>=57){
        parar();
        auxruta++;
        delay(500);
        IZQ=0;
        DER=0;
      }else{
        if (IZQ < DER || DER>=57){
          adelanteruedaizquierda();
        }else if (IZQ > DER || IZQ>=57){
          adelanteruedaderecha();
        }else{
          Serial.print("Pulsos en rueda derecha=");
          Serial.println(DER);
          Serial.print("Pulsos en rueda izquierda=");
          Serial.println(IZQ);
          Serial.println("+++++++");
          adelante();
        }
      }
    }
  }else if (ARUTA[auxruta]=="DERECHA45"){

    if (IZQ>=9 && DER>=9){
      parar();
      auxruta++;
      delay(500);
      IZQ=0;
      DER=0;
    }else{
      if (IZQ < DER || DER>=9){

```

```

        adelanteruedaizquierda();
    }else if (IZQ > DER || IZQ>=9){
        atrasruedaderecha();
    }else{

        Serial.print("Pulsos en rueda derecha=");
        Serial.println(DER);
        Serial.print("Pulsos en rueda izquierda=");
        Serial.println(IZQ);
        Serial.println("+++++++");
        derecha();
    }
}
}else if (ARUTA[auxruta]=="RIGHT90"){

    if(IZQ>=19 && DER>=19){
        parar();
        auxruta++;
        delay(500);
        IZQ=0;
        DER=0;
    }else{
        if(IZQ < DER || DER>=19){
            adelanteruedaizquierda();
        }else if (IZQ > DER || IZQ>=19){
            atrasruedaderecha();
        }else{

            Serial.print("Pulsos en rueda derecha=");
            Serial.println(DER);
            Serial.print("Pulsos en rueda izquierda=");
            Serial.println(IZQ);
            Serial.println("+++++++");
            derecha();
        }
    }
}

}else if (ARUTA[auxruta]=="CERTO135"){

    if(IZQ>=27 && DER>=27){
        parar();
        auxruta++;
        delay(500);
        IZQ=0;
        DER=0;
    }else{
        if(IZQ < DER || DER>=27){
            adelanteruedaizquierda();
        }else if (IZQ > DER || IZQ>=27){
            atrasruedaderecha();
        }else{

            Serial.print("Pulsos en rueda derecha=");
            Serial.println(DER);
            Serial.print("Pulsos en rueda izquierda=");
            Serial.println(IZQ);
            Serial.println("+++++++");
            derecha();
        }
    }
}

}else if (ARUTA[auxruta]=="IZQUIERDA45"){

    if(IZQ>=9 && DER>=9){
        parar();
        auxruta++;
        delay(500);
        IZQ=0;
        DER=0;
    }else{
        if(IZQ < DER || DER>=9){
            atrasruedaizquierda();
        }else if (IZQ > DER || IZQ>=9){
            adelanteruedaderecha();
        }else{

            Serial.print("Pulsos en rueda derecha=");
            Serial.println(DER);
            Serial.print("Pulsos en rueda izquierda=");
            Serial.println(IZQ);
            Serial.println("+++++++");
            izquierda();
        }
    }
}

}else if (ARUTA[auxruta]=="LEFT90"){

```

```

if(IQ>=18 && DER>=18){
    parar();
    auxruta++;
    delay(500);
    IQ=0;
    DER=0;
}else{
    if(IQ < DER || DER>=18){
        atrasruedaizquierda();
    }else if (IQ > DER || IQ>=18){
        adelanteruedaderecha();
    }else{

        Serial.print("Pulsos en rueda derecha=");
        Serial.println(DER);
        Serial.print("Pulsos en rueda izquierda=");
        Serial.println(IQ);
        Serial.println("+++++++");
        izquierda();
    }
}

}else if (ARUTA[auxruta]=="DEIXE1135"){

    if(IQ>=27 && DER>=27){
        parar();
        auxruta++;
        delay(500);
        IQ=0;
        DER=0;
    }else{
        if(IQ < DER || DER>=27){
            atrasruedaizquierda();
        }else if (IQ > DER || IQ>=27){
            adelanteruedaderecha();
        }else{

            Serial.print("Pulsos en rueda derecha=");
            Serial.println(DER);
            Serial.print("Pulsos en rueda izquierda=");
            Serial.println(IQ);
            Serial.println("+++++++");
            izquierda();
        }
    }

}else if (ARUTA[auxruta]=="180"){

    if(IQ>=36 && DER>=36){
        parar();
        auxruta++;
        delay(500);
        IQ=0;
        DER=0;
    }else{
        if(IQ < DER || DER>=36){
            atrasruedaizquierda();
        }else if (IQ > DER || IQ>=36){
            adelanteruedaderecha();
        }else{

            Serial.print("Pulsos en rueda derecha=");
            Serial.println(DER);
            Serial.print("Pulsos en rueda izquierda=");
            Serial.println(IQ);
            Serial.println("+++++++");
            izquierda();
        }
    }

}

}else{
    secuencias=0;
    auxruta=0;
    rutaC ="1";
    consul=0;
    HacerRuta=0;
    for(int i=1; i <100 ; i+=1) {
        ARUTA[HacerRuta]="0";
    }
    //modifica();
}

}else{
    secuencias=0;

```

```

        auxruta=0;
        rutaC ="1";
        consul=0;
        HacerRuta=0;
        //modifica();
    }
}
}
}

void modifica(){
    unsigned long currentMillis = millis();

    if ( currentMillis - previousMillis > watchdog ) {
        previousMillis = currentMillis;
        WiFiClient client;

        if (!client.connect(host, port)) {
            Serial.println("Conexión falló...");
            return;
        }

        String url = "/programas_phpLab2/proceso_eventos/ModificaEnable.php?enable=";
        url += 1;

        // Envío de la solicitud al Servidor
        client.print(String("POST ") + url + " HTTP/1.1\r\n" +
            "Host: " + host + "\r\n" +
            "Connection: close\r\n\r\n");
        unsigned long timeout = millis();
        while (client.available() == 0) {
            if (millis() - timeout > 5000) {
                Serial.println(">>> Superado tiempo de espera!");
                client.stop();
                return;
            }
        }
        // Lee respuesta del servidor
        while(client.available()){
            line = client.readStringUntil('\r');
            Serial.print(line);
        }

        Serial.print("Dato ENVIADO");
    }
}

void consulta(){
    unsigned long currentMillis = millis();

    if ( currentMillis - previousMillis > watchdog ){
        previousMillis = currentMillis;
        WiFiClient client;

        if (!client.connect(host, port)) {
            Serial.println("Conexión falló...");
            return;
        }

        String url = "/programas_phpLab2/proceso_eventos/ConsultarRuta.php";

        // Envío de la solicitud al Servidor
        client.print(String("POST ") + url + " HTTP/1.1\r\n" +
            "Host: " + host + "\r\n" +
            "Connection: close\r\n\r\n");
        unsigned long timeout = millis();
        while (client.available() == 0) {
            if (millis() - timeout > 5000) {
                Serial.println(">>> Superado tiempo de espera!");
                client.stop();
                return;
            }
        }
        // Lee respuesta del servidor
        while(client.available()){
            line = client.readStringUntil('\r');
        }
    }
}

void espera(float seg){
    delay(seg*1000);
}

```



```

void adelante(){
    digitalWrite(OUTPUT1, 0);
    digitalWrite(OUTPUT2, 1);
    digitalWrite(OUTPUT3, 1);
    digitalWrite(OUTPUT4, 0);
}

void derecha(){
    digitalWrite(OUTPUT1, 0);
    digitalWrite(OUTPUT2, 1);
    digitalWrite(OUTPUT3, 0);
    digitalWrite(OUTPUT4, 1);
}

void izquierda(){
    digitalWrite(OUTPUT1, 1);
    digitalWrite(OUTPUT2, 0);
    digitalWrite(OUTPUT3, 1);
    digitalWrite(OUTPUT4, 0);
}

void parar(){
    digitalWrite(OUTPUT1, 0);
    digitalWrite(OUTPUT2, 0);
    digitalWrite(OUTPUT3, 0);
    digitalWrite(OUTPUT4, 0);
}

void adelanteruedaizquierda(){
    digitalWrite(OUTPUT1, 0);
    digitalWrite(OUTPUT2, 1);
    digitalWrite(OUTPUT3, 0);
    digitalWrite(OUTPUT4, 0);
}

void adelanteruedaderecha(){
    digitalWrite(OUTPUT1, 0);
    digitalWrite(OUTPUT2, 0);
    digitalWrite(OUTPUT3, 1);
    digitalWrite(OUTPUT4, 0);
}

void atrasruedaderecha(){
    digitalWrite(OUTPUT1, 0);
    digitalWrite(OUTPUT2, 0);
    digitalWrite(OUTPUT3, 0);
    digitalWrite(OUTPUT4, 1);
}

void atrasruedaizquierda(){
    digitalWrite(OUTPUT1, 1);
    digitalWrite(OUTPUT2, 0);
    digitalWrite(OUTPUT3, 0);
    digitalWrite(OUTPUT4, 0);
}

```

CÓDIGO FINAL DE LA INTERFAZ GENERADORA DE RUTA

Clase: FigurePattern

```

1 /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5 package net.jc_mouse.unlockpattern;
6 import java.awt.BasicStroke;
7 import java.awt.Color;
8 import java.awt.Graphics2D;
9 import java.awt.Point;
10 import java.awt.Rectangle;
11 import java.awt.geom.Ellipse2D;
12 /**
13  * @web https://www.jc-mouse.net/
14  * @author Mouse
15  */
16 public class FigurePattern {
17     //identificador/valor de objeto (NO debe repetirse)
18     private int key;

```

```

19 //coordenadas
20 private int x = 0;
21 private int y = 0;
22 //bandera
23 private boolean selected = false;
24 //area sensible a eventos del raton
25 private Rectangle area = new Rectangle();
26
27 /**Constructor de clase */
28 public FigurePattern(){
29
30 /**
31  * Constructor de clase
32  * @param k clave de celda
33  * @param x coordenada X de objeto
34  * @param y coordenada Y de objeto
35  */
36 public FigurePattern(int k, int x, int y){
37     this.key = k;
38     this.x = x;
39     this.y = y;
40     //area interna del circulo
41     area.setBounds(x+10, y+10 , 60, 60);
42 }
43
44 public void draw(Graphics2D g2){
45     //fondo
46     g2.setColor( new Color(0,0,0) );
47     g2.fill( new Ellipse2D.Double(x+10, y+10 , 60, 60 ) );
48     //borde exterior
49     g2.setStroke(new BasicStroke( 4 ));
50     g2.setColor( (selected)?new Color(0,204,0):new Color(153,153,153) );
51     g2.draw( new Ellipse2D.Double(x+10, y+10 , 60, 60 ) );
52     //circulo interno
53     g2.setColor( new Color(255,255,255) );
54     g2.fill( new Ellipse2D.Double(x+30, y+30 , 20, 20 ) );
55 }
56
57 public int getKey() {
58     return key;
59 }
60
61 public void setKey(int key) {
62     this.key = key;
63 }
64
65 public boolean isSelected() {
66     return selected;
67 }
68
69 public void setSelected(boolean selected) {
70     this.selected = selected;
71 }
72
73 public Rectangle getArea() {
74     return area;
75 }
76
77 public void setArea(Rectangle area) {
78     this.area = area;
79 }
80
81 /**
82  * retorna el punto medio de la figura
83  * @return Point
84  */
85 public Point getCenterPoint(){
86     Point p = new Point( x+40, y+40);
87     return p;
88 }
89
90 /**
91  * Metodo statico que determina si un punto se encuentra dentro un area rectangular
92  * @param r rectangle
93  * @param p point
94  * @return boolean
95  */
96 public static boolean insideArea(Rectangle r,Point p) {
97     return r.contains(p);
98 }
99
100 }//end:FigurePattern

```

Clase: UnlockJFrame

```
1 /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
4  */
5 package net.jc_mouse.unlockpattern;
6 import java.awt.Dimension;
7 import java.awt.EventQueue;
8 import javax.swing.JFrame;
9 import javax.swing.WindowConstants;
10 /**
11  * @web https://www.jc-mouse.net/
12  * @author Mouse
13  */
14 public class UnlockJFrame extends JFrame {
15
16     private UnlockPanel unlockPanel;
17
18     /**Constructor de clase*/
19     public UnlockJFrame(){
20         initComponents();
21         UnlockJFrame.this.setLocationRelativeTo(null);
22     }
23
24     private void initComponents() {
25         setTitle("Unlock Pattern");
26         setResizable(false);
27         setSize(new Dimension(340,360));
28         setPreferredSize(new Dimension(340,360));
29         unlockPanel = new UnlockPanel(this);
30         setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
31         getContentPane().add(unlockPanel);
32         pack();
33     }
34
35     public static void main(String args[]){
36         EventQueue.invokeLater(() -> {
37             new UnlockJFrame().setVisible(true);
38         });
39     }
40
41 }
```

Clase:UnlockPanelJava

```
1 /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5 package net.jc_mouse.unlockpattern;
6
7 import java.awt.BasicStroke;
8 import java.awt.Color;
9 import java.awt.Graphics;
10 import java.awt.Graphics2D;
11 import java.awt.Image;
12 import java.awt.Point;
13 import java.awt.RenderingHints;
14 import java.awt.event.MouseEvent;
15 import java.awt.event.MouseListener;
16 import java.awt.event.MouseMotionListener;
17 import java.awt.geom.Rectangle2D;
18 import java.util.ArrayList;
19 import javax.swing.Imagelcon;
20 import javax.swing.JFrame;
21 import javax.swing.JPanel;
22 import javax.swing.*;
23
24 import java.sql.Connection;
25 import java.sql.DriverManager;
26 import java.sql.ResultSet;
27 import java.sql.SQLException;
28 import java.sql.Statement;
29 import java.util.logging.Level;
30 import java.util.logging.Logger;
31 import java.sql.PreparedStatement;
32 //import net.jc_mouse.demo.AppDemoJFrame;
33 /**
34  * @web https://www.jc-mouse.net/
35  * @author Mouse
36  */
37 public class UnlockPanel extends JPanel implements MouseListener,MouseMotionListener {
```

```

38
39 private JFrame parent;
40 private String myKey="125895";//patron de desbloqueo
41 //array para las 9 figuras que forman la matriz
42 private ArrayList<FigurePattern> figurePatternArr = new ArrayList();
43 //array para las figuras que conforman el patron desbloqueo
44 private ArrayList<FigurePattern> pattern= new ArrayList();
45 //String para guardar el patron generado por el usuario
46 private String password = "";
47 //para mostrar imagen de "acceso denegado"
48 //private final Image //image = new ImageIcon(getClass().getResource("/net/jc_mouse/unlockpattern/access_denied.jpg")).getImage();
49 private boolean showError=false;
50 public char[] Ruta;
51 public char[] RutaNueva;
52 public int Longitud;
53 public int Final1;
54 public int Final2;
55 public int Inicial1;
56 public int Inicial2;
57 public int Hacer;
58 public int variable;
59 public int id;
60 public String taru;
61 public String taruconvertida;
62 public ArrayList<String> EnviarRuta = new ArrayList<String>();
63
64 public String usuario ="root";
65 public String clave="";
66 public String url="jdbc:mysql://localhost:3306/carroautonomo";
67 public Connection con;
68 public Statement stmt;
69 public ResultSet rs;
70
71
72 /**
73  * Constructor de clas
74  * @param parent JFrame que contiene el JPanel
75  */
76 public UnlockPanel(JFrame parent){
77     this.parent = parent;
78     //se agregan 9 circulos al contenedor
79     figurePatternArr.add(new FigurePattern(11,0,0));
80     figurePatternArr.add(new FigurePattern(12,80,0));
81     figurePatternArr.add(new FigurePattern(13,160,0));
82     figurePatternArr.add(new FigurePattern(14,240,0));
83
84     figurePatternArr.add(new FigurePattern(15,0,80));
85     figurePatternArr.add(new FigurePattern(16,80,80));
86     figurePatternArr.add(new FigurePattern(17,160,80));
87     figurePatternArr.add(new FigurePattern(18,240,80));
88
89     figurePatternArr.add(new FigurePattern(19,0,160));
90     figurePatternArr.add(new FigurePattern(20,80,160));
91     figurePatternArr.add(new FigurePattern(21,160,160));
92     figurePatternArr.add(new FigurePattern(22,240,160));
93
94     figurePatternArr.add(new FigurePattern(23,0,240));
95     figurePatternArr.add(new FigurePattern(24,80,240));
96     figurePatternArr.add(new FigurePattern(25,160,240));
97     figurePatternArr.add(new FigurePattern(26,240,240));
98
99     //Eventos del raton
100     UnlockPanel.this.addMouseListener( UnlockPanel.this );
101     UnlockPanel.this.addMouseMotionListener( UnlockPanel.this);
102 }
103
104 @Override
105 public void paintComponent(Graphics g){
106     Graphics2D g2 =(Graphics2D) g;
107     g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,RenderingHints.VALUE_ANTIALIAS_ON);
108     g2.setColor( new Color(0,0,0) );
109     g2.fill(new Rectangle2D.Double(0,0,getWidth(),getHeight()));
110
111     //dibuja las lineas del patron de desbloqueo entre los circulos
112     Point point = null;
113     for(FigurePattern figurePattern: pattern){
114         g2.setStroke( new BasicStroke( 20 ) );
115         g2.setColor( new Color(204,204,204) );
116         if(point!=null)
117             g2.drawLine(point.x, point.y, figurePattern.getCenterPoint().x, figurePattern.getCenterPoint().y);
118         point = figurePattern.getCenterPoint();
119     }
120
121     //dibuja los circulos para los patrones de desbloqueo
122     if(figurePatternArr!=null)
123         figurePatternArr.stream().forEach((b) -> {

```

```

124         b.draw(g2);
125     });
126
127     //muestra error
128     if(showError){
129         //g2.drawImage(image, (getWidth()-220)/2, (getHeight()-93)/2, 220, 93, null);
130     }
131
132 }
133
134 @Override
135 public void mouseClicked(MouseEvent e) {}
136
137 @Override
138 public void mousePressed(MouseEvent e) {}
139
140 @Override
141 public void mouseReleased(MouseEvent e) {
142     checkUnlockPattern();
143 }
144
145 @Override
146 public void mouseEntered(MouseEvent e) {}
147
148 @Override
149 public void mouseExited(MouseEvent e) {}
150
151 @Override
152 public void mouseDragged(MouseEvent e) {
153     //cuando se arrastra el cursor del mouse
154     figurePatternArr.stream().filter((p) -> (
155         FigurePattern.insideArea(p.getArea(), e.getPoint())
156     )).forEach((p) -> {
157         if(this.pattern.isEmpty()){ //Si esta vacio se añade objeto
158             pattern.add(p); //añade a array
159             p.setSelected(true); //cambia color
160             password += p.getKey(); //concatena valor de figura
161         } else { //Si ya existen objetos en el array
162             //se comprueba que objeto nuevo no se repita con el ultimo añadido
163             if(pattern.get(pattern.size()-1).getKey() != p.getKey()){
164                 pattern.add(p); //añade a array
165                 p.setSelected(true); //cambia color
166                 password += p.getKey(); //concatena valor de figura
167             }
168         }
169     });
170     repaint();
171 }
172
173 @Override
174 public void mouseMoved(MouseEvent e) { showError=false; repaint(); }
175
176 /**
177  * metodo para comprobar que patron de desbloqueo es correcto
178  * de ser asi -> abre nuevo formulario
179  * caso contrario -> muestra mensaje de error
180  */
181 public void checkUnlockPattern(){
182     PreparedStatement ps;
183     String sql;
184     Ruta= password.toCharArray();
185     Longitud=Ruta.length;
186     Longitud--;
187     Final1 = Character.getNumericValue(Ruta[Longitud-1]);
188     Final2 = Character.getNumericValue(Ruta[Longitud]);
189     Inicial1 = Character.getNumericValue(Ruta[0]);
190     Inicial2 = Character.getNumericValue(Ruta[1]);
191     if((Inicial1==2 && Inicial2==3){
192         if(Final1==1 && Final2==4)
193         {
194             JOptionPane.showMessageDialog(null,
195                 "Ruta enviada",
196                 "PopUp Dialog",
197                 JOptionPane.INFORMATION_MESSAGE);
198
199             taru=password;
200
201             taruconvertida = conversorDeRuta(taru, 4);
202             //RutaLista();
203
204
205         }
206     }
207     try {
208         Class.forName("com.mysql.cj.jdbc.Driver");
209     } catch (ClassNotFoundException ex) {
210         Logger.getLogger(UnlockPanel.class.getName()).log(Level.SEVERE, null, ex);
211     }
212 }

```

```

210     }
211
212     try {
213         con=DriverManager.getConnection(url,usuario,clave);
214         stmt=con.createStatement();
215
216         stmt.executeUpdate("DELETE FROM carroaut");
217
218
219         sql = "INSERT INTO carroaut VALUES(?,?,?)";
220         ps = con.prepareStatement(sql);
221         id=1;
222         ps.setInt(1,id);
223         ps.setInt(2, 0);
224         ps.setString(3,taruconvertida);
225         ps.executeUpdate();
226         //stmt.executeUpdate("INSERT INTO carroaut VALUES(null,0,i)");
227         System.out.println(taruconvertida);
228
229
230
231     } catch (SQLException ex) {
232         Logger.getLogger(UnlockPanel.class.getName()).log(Level.SEVERE, null, ex);
233     }
234
235     } else{
236         //showError=true;
237         JOptionPane.showMessageDialog(null,
238             "El final de la ruta es incorrecta",
239             "PopUp Dialog",
240             JOptionPane.INFORMATION_MESSAGE);
241     }
242     } else{
243         JOptionPane.showMessageDialog(null,
244             "El inicio de la ruta es incorrecta",
245             "PopUp Dialog",
246             JOptionPane.INFORMATION_MESSAGE);
247     }
248     pattern.clear();//limpia movimientos antiguos
249     password="";//resetea password
250     //reinicia valor para objetos
251     figurePatternArr.stream().forEach((b) -> {
252         b.setSelected(false);
253     });
254     repaint();
255     EnviarRuta.clear();
256 }
257
258
259 public String conversorDeRuta(String ruta, int columnas) {
260     String rutaConvertida = "5";
261     char[] aCaracteres = ruta.toCharArray();
262
263     for (int i = 0; i < ruta.length()-2; i+=2) {
264         int a = aCaracteres[i]*10+aCaracteres[i+1]-(aCaracteres[i+2]*10+aCaracteres[i+3]);
265
266         if(a==1) {
267             rutaConvertida += "7";if(i+1<ruta.length()-3){rutaConvertida+="3";}
268         }else if(a==1) {
269             rutaConvertida += "3";if(i+1<ruta.length()-3){rutaConvertida+="7";}
270         }else if(a==columnas) {
271             rutaConvertida += "5";if(i+1<ruta.length()-3){rutaConvertida+="1";}
272         }else if(a==columnas) {
273             rutaConvertida += "1";if(i+1<ruta.length()-3){rutaConvertida+="5";}
274         }else if(a==columnas-1) {
275             rutaConvertida += "4";if(i+1<ruta.length()-3){rutaConvertida+="8";}
276         }else if(a==columnas+1) {
277             rutaConvertida += "8";if(i+1<ruta.length()-3){rutaConvertida+="4";}
278         }else if(a==columnas+1) {
279             rutaConvertida += "6";if(i+1<ruta.length()-3){rutaConvertida+="2";}
280         }else if(a==columnas-1) {
281             rutaConvertida += "2";if(i+1<ruta.length()-3){rutaConvertida+="6";}
282         }
283     }
284     return rutaConvertida;
285 }
286
287 }//end

```

CONCLUSIONES

- El IDE de Arduino es un software muy esencial para el desarrollo de prototipos de este tipo, ya que posee herramientas y bibliotecas que permiten darle funcionalidad de manera sencilla e intuitiva a los componentes y a la placa base del proyecto.
- Para llevar a cabo el desarrollo de cualquier prototipo con componentes electrónicos se debe revisar muy cuidadosamente la alimentación y las respectivas conexiones para así evitar que los dispositivos se quemen o se dañen.
- Llevar a cabo una buena investigación tanto bibliográfica como visual es de gran utilidad ya que es posible encontrar información útil para el desarrollo del proyecto.

BIBLIOGRAFÍA

[1] [\[Tutorial Electronica\] - Carro controlado con Arduino, Bluetooth y Android](#)

[2] <https://www.jc-mouse.net/proyectos/crea-patron-de-desbloqueo-para-tus-app-java>

[3] http://www.mediafire.com/file/8hgs5i3x2drh17e/CARRO_BLUETOOTH.ino/file