



Tecnológico de Estudios Superiores de Ecatepec

**División de Ingeniería en Sistemas
Computacionales**

Academia en Ciencias de la Ingeniería

Materia: Lenguajes y Autómatas II

Grupo 5602

Alumno

Campero Granados Luis Daniel

Profesor

Hernández Rojas Rosa María

Proyecto segundo parcial

Índice

1. Introducción	3
2. Objetivo General	3
3. Objetivos Específicos	4
4. Competencias para desarrollar	4
5. Desarrollo	5 - 20
5.1. Código	5 - 11
5.2. Capturas	11 - 20
6. Conclusión	21
7. Referencias	21
8. Link de video	21

1. Introducción

Un AP es un autómata de pila, el cual es un modelo matemático de un sistema que recibe una cadena constituida por símbolos de un alfabeto y determina si esa cadena pertenece al lenguaje que el autómata reconoce. Nosotros nos basaremos para desarrollar un programa que valide nuestro autómata creado, este autómata tendrá que validar nuestro RFC y con ellos mostrar el recorrido que hace el autómata.

Los autómatas de pilas nos sirven para validar lenguajes que los autómatas finitos no pueden, estos funcionan mediante el manejo de LIFO (Last In First Out) y esto hace que los datos pueden ser insertados o extraídos de la pila.

Ingresaremos en el programa nuestro RFC, este mostrará el recorrido que hace nuestro autómata, desde cómo van ingresando y saliendo los datos, hasta el que estado se encuentra y como va pasando hasta llegar al final.

2. Objetivo general

Validar nuestro RFC mediante el uso de un autómata de pila que utiliza el método LIFO, con el desarrollo de un programa en el lenguaje de programación JAVA.

3. Objetivos específicos

- Elaborar un autómata capaz de aceptar como entrada nuestro RFC y lo valide.
- Utilizar los métodos aprendidos, para el desarrollo del programa.
- Desarrollar un programa capaz de mostrar el cómo funciona el autómata.

4. Competencias para desarrollar

- Conocer herramientas que faciliten la creación del autómata y programa.
- Capacidad de análisis y síntesis
- Solución de problemas
- Adaptabilidad en entornos de trabajo
- Planificación y organización

5. Desarrollo

5.1. Código

```
import java.awt.Image;

import java.util.*;

import javax.swing.*;

import javax.swing.table.*;

public class Pila extends javax.swing.JFrame
implements Runnable {

    ImageIcon Est0 = new
    ImageIcon(getClass().getResource("/images/Auto
    mata.jpg"));

    ImageIcon Est1 = new
    ImageIcon(getClass().getResource("/images/Estad
    o1.jpg"));

    ImageIcon Est2 = new
    ImageIcon(getClass().getResource("/images/Estad
    o2.jpg"));

    ImageIcon Est3 = new
    ImageIcon(getClass().getResource("/images/Estad
    o3.jpg"));

    ImageIcon Est4 = new
    ImageIcon(getClass().getResource("/images/Estad
    o4.jpg"));

    private String RFC, estado, cadena, pila, letra;

    private Thread h1;

    private int con;

    private ArrayList<Object[] > datos;

    public Pila() {

        initComponents();

        estado = "Estado Q0";

        h1 = new Thread(this);

        con = 0;

        datos = new ArrayList<Object[]>();

        tabla();

        Imagen.setIcon(Est0);

    }

    @SuppressWarnings("unchecked")

    // <editor-fold defaultstate="collapsed"
    desc="Generated Code">

    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();

        jScrollPane1 = new
        javax.swing.JScrollPane();

        Tabla = new javax.swing.JTable();

        Imagen = new javax.swing.JLabel();

        jLabel1 = new javax.swing.JLabel();

        TFRFC = new javax.swing.JTextField();

        jConfirmar = new javax.swing.JButton();

        Limpiar = new javax.swing.JButton();

        jLabel3 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowCo
        nstants.EXIT_ON_CLOSE);

        jPanel1.setBackground(new
        java.awt.Color(204, 204, 255));

        Tabla.setModel(new
        javax.swing.table.DefaultTableModel(

            new Object [][] {

                {null, null, null, null},

                {null, null, null, null},

                {null, null, null, null},

            }

        )

    }

}
```

```

        {null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
));
jScrollPane1.setViewportViewView(Tabla);

Imagen.setIcCon(new
javax.swing.ImageIcon(getClass().getResource("/i
mages/Automata.jpg"))); // NOI18N

Imagen.setBorder(javax.swing.BorderFactory.creat
eLineBorder(new java.awt.Color(0, 0, 0), 2));

jLabel1.setForeground(new java.awt.Color(0,
0, 0));

jLabel1.setText("Ingrese su RFC:");

TFRFC.setBackground(new
java.awt.Color(204, 204, 204));

TFRFC.setForeground(new java.awt.Color(0,
0, 0));

jConfirmar.setBackground(new
java.awt.Color(255, 255, 255));

jConfirmar.setForeground(new
java.awt.Color(0, 0, 0));

jConfirmar.setText("Realizar");

jConfirmar.addActionListener(new
java.awt.event.ActionListener() {

    public void
    actionPerformed(java.awt.event.ActionEvent evt) {

        jConfirmarActionPerformed(evt);

    }

});

```

```

        Limpiar.setBackground(new
java.awt.Color(255, 255, 255));

        Limpiar.setForeground(new java.awt.Color(0,
0, 0));

        Limpiar.setText("Limpiar");

        Limpiar.addActionListener(new
java.awt.event.ActionListener() {

            public void
            actionPerformed(java.awt.event.ActionEvent evt) {

                LimpiarActionPerformed(evt);

            }

        });

        jLabel3.setFont(new java.awt.Font("Dialog", 1,
24)); // NOI18N

        jLabel3.setForeground(new java.awt.Color(0,
0, 0));

        jLabel3.setHorizontalAlignment(javax.swing.Swing
Constants.CENTER);

        jLabel3.setText("Proyecto segundo parcial");

        jLabel3.setHorizontalTextPosition(javax.swing.Swi
ngConstants.CENTER);

        javax.swing.GroupLayout jPanel1Layout =
new javax.swing.GroupLayout(jPanel1);

        jPanel1.setLayout(jPanel1Layout);

        jPanel1Layout.setHorizontalGroup(

            jPanel1Layout.createParallelGroup(javax.swing.Grou
pLayout.Alignment.LEADING)

            .addGroup(jPanel1Layout.createSequentialGroup()

                .addGap(10, 10, 10)

                .addContainerGap(10)

                .addGroup(jPanel1Layout.createParallelGroup(jav
ax.swing.GroupLayout.Alignment.LEADING)

                    .addGroup(jPanel1Layout.createSequentialGroup()

                        .addGroup(jPanel1Layout.createSequentialGroup()

                            .addContainerGap(10)

                            .addGroup(jPanel1Layout.createParallelGroup(jav
ax.swing.GroupLayout.Alignment.LEADING)

```

```

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup())

        .addGap(0, 0, Short.MAX_VALUE)

        .addComponent(Limpiar,
javax.swing.GroupLayout.PREFERRED_SIZE,
560,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(jPanel1Layout.createSequentialGroup(
)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

        .addComponent(jScrollPane1,
javax.swing.GroupLayout.DEFAULT_SIZE, 560,
Short.MAX_VALUE)

.addGroup(jPanel1Layout.createSequentialGroup(
)

        .addComponent(jLabel1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(TFRFC))

        .addComponent(jConfirmar,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))

        .addComponent(Imagen,
javax.swing.GroupLayout.PREFERRED_SIZE,
560,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(0, 0,
Short.MAX_VALUE)))

        .addContainerGap())

        .addComponent(jLabel3,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

);

jPanel1Layout.setVerticalGroup(

```

```

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup(
)

        .addComponent(jLabel3,
javax.swing.GroupLayout.PREFERRED_SIZE, 29,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(12, 12, 12)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(jLabel1)

        .addComponent(TFRFC,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addComponent(jConfirmar)

        .addGap(12, 12, 12)

        .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE,
140,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addComponent(Imagen,
javax.swing.GroupLayout.DEFAULT_SIZE, 217,
Short.MAX_VALUE)

        .addGap(12, 12, 12)

        .addComponent(Limpiar)

        .addContainerGap())

);

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

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

```

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

```
        .addComponent(jPanel1,
            javax.swing.GroupLayout.PREFERRED_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
    );
```

```
    layout.setVerticalGroup(
```

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

```
        .addComponent(jPanel1,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            Short.MAX_VALUE)
```

```
    );
```

```
    pack();
```

```
// </editor-fold>
```

```
private void
jConfirmarActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    this.RFC = TFRFC.getText();
```

```
    if(this.RFC.length() < 10){
```

```
JOptionPane.showMessageDialog(rootPane,
    "Agregue un RFC de 10 letras");
```

```
    return;
```

```
}
```

```
    if(this.RFC.length() > 10){
```

```
JOptionPane.showMessageDialog(rootPane,
    "Agregue un RFC de 10 letras");
```

```
    return;
```

```
}
```

```
h1.start();
```

```
}
```

```
private void
LimpiarActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    Limpiar();
```

```
}
```

```
private void tablar(){
```

```
    DefaultTableModel modelo = new
    DefaultTableModel();
```

```
    String[] columnas = {"Estado", "Por leer",
    "Pila"};
```

```
    for(String columna : columnas){
```

```
        modelo.addColumn(columna);
```

```
    }
```

```
    Tabla.setModel(modelo);
```

```
}
```

```
public static void main(String args[]) {
```

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

```
        public void run() {
```

```
            new Pila().setVisible(true);
```

```
        }
```

```
    });
```

```
}
```

```
// Variables declaration - do not modify
```

```
private javax.swing.JLabel Imagen;
```

```
private javax.swing.JButton Limpiar;
```

```
private javax.swing.JTextField TFRFC;
```

```
private javax.swing.JTable Tabla;
```

```
private javax.swing.JButton jConfirmar;
```

```
private javax.swing.JLabel jLabel1;
```

```
private javax.swing.JLabel jLabel3;
```

```
private javax.swing.JPanel jPanel1;
```

```
private javax.swing.JScrollPane jScrollPane1;
```



```

// End of variables declaration

@Override
public void run(){
    while(con <= this.RFC.length()){
        if(con == 0){
            cadena = this.RFC;
            pila = "";
            cargaTabla(estado, cadena, pila);
            Imagen.setIcon(Est1);
        }
        if(con > 0){
            letra = this.RFC.substring(con - 1, con);
            cadena = this.RFC.substring(con,
this.RFC.length());
            switch (estado){
                case "Estado Q0":
                    Imagen.setIcon(Est1);
                    switch (letra){
                        case "C":
                            pila = "C" + pila;
                            estado = "Estado Q0";
                            cargaTabla(estado, cadena,
pila);
                            break;
                        case "A":
                            pila = "A" + pila;
                            estado = "Estado Q0";
                            cargaTabla(estado, cadena,
pila);
                            break;
                        case "G":
                            pila = "G" + pila;
                            estado="Estado Q0";
                            cargaTabla(estado, cadena,
pila);
                            break;
                    }
                case "L":
                    pila = "L" + pila;
                    estado = "Estado Q0";
                    cargaTabla(estado, cadena,
pila);
                    break;
                case "9":
                    Imagen.setIcon(Est2);
                    pila = pila;
                    estado = "Estado Q1";
                    cargaTabla(estado, cadena,
pila);
                    break;
                default:
                    JOptionPane.showMessageDialog(rootPane, "El
RFC no pasa por el Automata");
                    con = this.RFC.length() + 1;
            }
        }
    }
}

```

```

        if(pila.substring(0,
1).equals("L")){
            Imagen.setIcon(Est3);
            pila = "GAC";
            estado = "Estado Q2";
            cargaTabla(estado,
cadena, pila);
        }
        else{
            JOptionPane.showMessageDialog(rootPane, "El
RFC no pasa por el Automata");
            con = this.RFC.length() +
1;
        }
        break;

        default:

            JOptionPane.showMessageDialog(rootPane, "El
RFC no pasa por el Automata");
            con = this.RFC.length() + 1;
        }
        break;

        case "Estado Q2":
            Imagen.setIcon(Est3);
            switch (letra){
                case "1":

                    if(pila.substring(0,1).equals("G")){
                        pila = "AC";
                        estado = "Estado Q3";
                        cargaTabla(estado,
cadena, pila);
                    }else{

```

```

JOptionPane.showMessageDialog(rootPane, "El
RFC no pasa por el Automata");
            con
            =
this.RFC.length()+1;
        }
        break;

        default:

            JOptionPane.showMessageDialog(rootPane, "El
RFC no pasa por el AutomataO");
            con = this.RFC.length() + 1;
        }
        break;

        case "Estado Q3":
            Imagen.setIcon(Est4);
            switch (letra){
                case "1":

                    if(pila.substring(0,1).equals("A")){
                        pila = "C";
                        estado = "Estado Q3";
                        cargaTabla(estado,
cadena, pila);
                        Imagen.setIcon(Est4);
                    }else{

                        JOptionPane.showMessageDialog(rootPane, "El
RFC no pasa por el Automata");
                        con = this.RFC.length() +
1;
                    }
                    break;

```

```

        case "0":
            if(pila.substring(0,
1).equals("C")){
                pila = "e";
                estado = "Estado Q3";
                cadena = "e";
                cargaTabla(estado,
cadena, pila);
            }
            else{

JOptionPane.showMessageDialog(rootPane, "El
RFC no pasa por el Automata");

                con
                =
this.RFC.length()+1;
            }
            break;

            default:

JOptionPane.showMessageDialog(rootPane, "El
RFC no pasa por el Automata");

                con = this.RFC.length()+1;
            }
            break;
        }
        con++;
        try{
            Thread.sleep(1000);
        }catch(InterruptedException e){
        }
    }
}

```

```

        private void cargaTabla(String estado, String
cadena, String pila){

            DefaultTableModel modelo = new
DefaultTableModel();

            String[] columnas = {"Estado", "Por Leer",
"Pila"};

            for(String columna : columnas){
                modelo.addColumn(columna);
            }

            Tabla.setModel(modelo);

            Object[] informacion = new Object[]{estado,
cadena, pila};

            datos.add(informacion);

            for (Object []datosL : datos) {
                modelo.addRow(datosL);
            }

            Tabla.setModel(modelo);
        }

        public void Limpiar(){
            TFRFC.setText("");

            javax.swing.table.DefaultTableModel dt;

            dt
            =
(javax.swing.table.DefaultTableModel)Tabla.getMo
del();

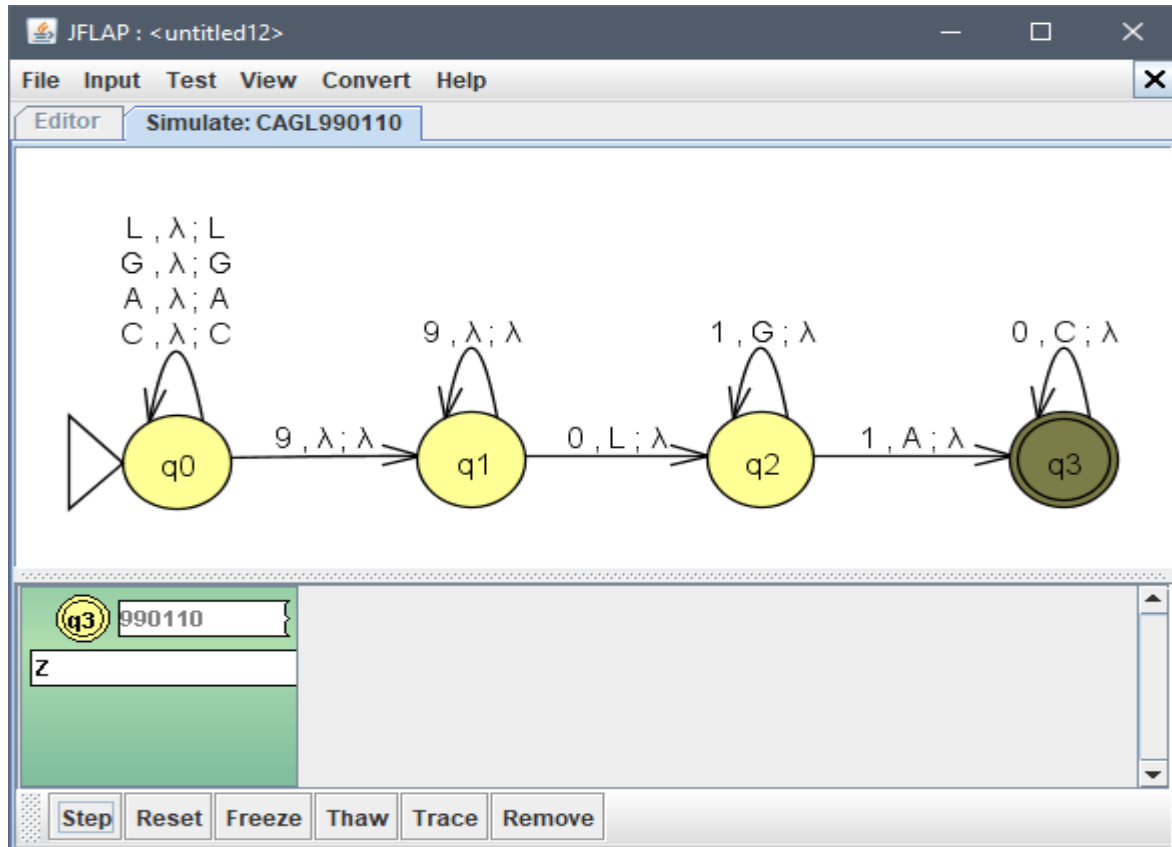
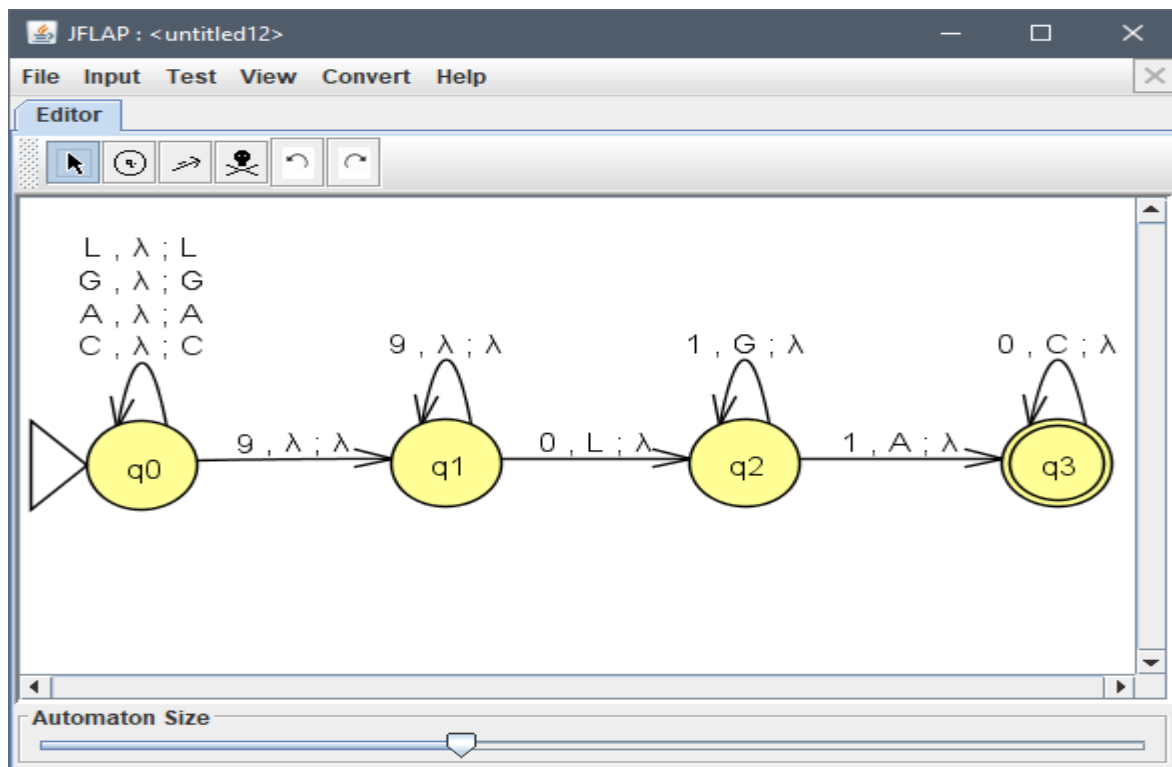
            int rowCount = dt.getRowCount();

            System.out.println(rowCount);

            while(rowCount >= 0){
                dt.removeRow(0);
            }
        }
    }
}

```

5.2. Capturas



Accepting configuration found! X

i

q0

CAGL990110

↓

Z

q0

CAGL990110

↓

CZ

q0

CAGL990110

↓

ACZ

q0

CAGL990110

↓

GACZ

q0

CAGL990110

↓

LGACZ

q1

CAGL990110

Keep looking

I'm done

Accepting configuration found! X

i

↓

q1CAGL990110

LGACZ

↓

q1CAGL990110

LGACZ

↓

q2CAGL990110

GACZ

↓

q2CAGL990110

ACZ

↓

q3CAGL990110

CZ

↓

q3CAGL990110

Keep looking

I'm done

14


Accepting configuration found!



q1	CAGL990110
LGACZ	
↓	
q1	CAGL990110
LGACZ	
↓	
q2	CAGL990110
GACZ	
↓	
q2	CAGL990110
ACZ	
↓	
q3	CAGL990110
CZ	
↓	
q3	CAGL990110
Z	

Keep looking

I'm done

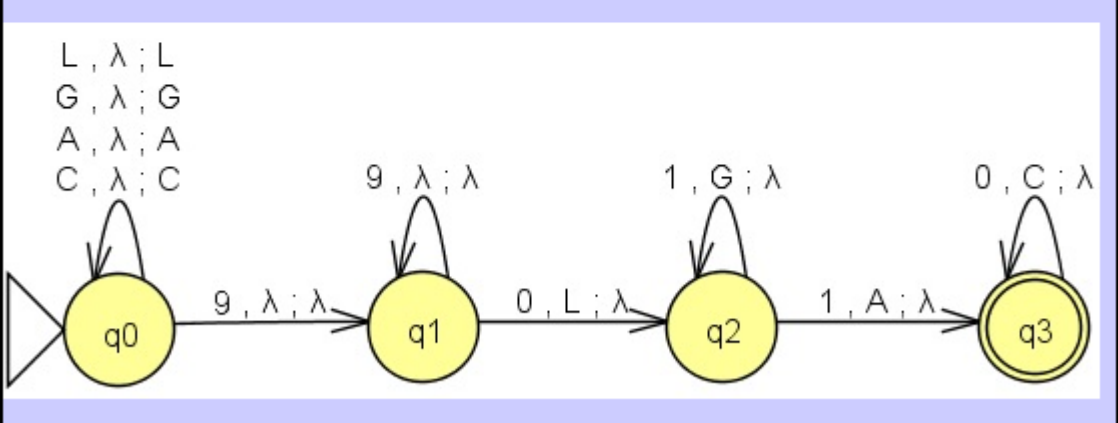

— □ ×

Proyecto segundo parcial

Ingrese su RFC:


Estado	Por leer	Pila

$L, \lambda; L$
 $G, \lambda; G$
 $A, \lambda; A$
 $C, \lambda; C$



```

graph LR
    start(( )) --> q0((q0))
    q0 -- "9, λ; λ" --> q1((q1))
    q1 -- "0, L; λ" --> q2((q2))
    q2 -- "1, A; λ" --> q3(((q3)))
    q0 -- "L, λ; L" --> q0
    q1 -- "9, λ; λ" --> q1
    q2 -- "1, G; λ" --> q2
    q3 -- "0, C; λ" --> q3
  
```

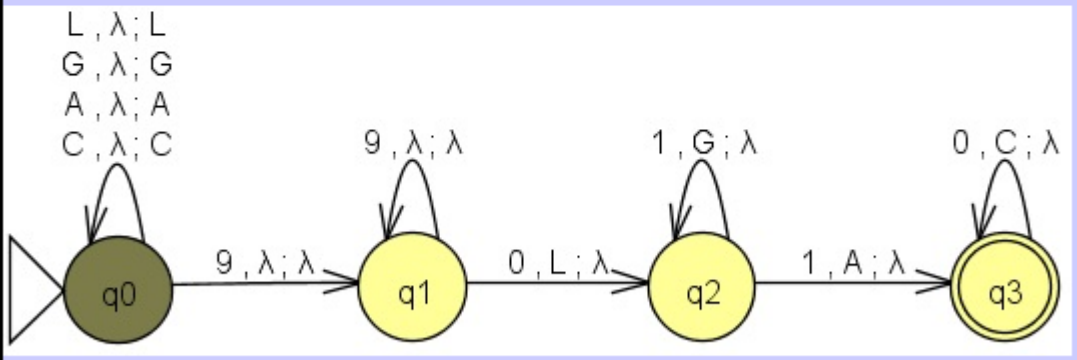


— □ ×

Proyecto segundo parcial

Ingrese su RFC:


Estado	Por Leer	Pila
Estado Q0	CAGL990110	
Estado Q0	AGL990110	C

$L, \lambda; L$
 $G, \lambda; G$
 $A, \lambda; A$
 $C, \lambda; C$



```

graph LR
    q0((q0)) -- "9, λ; λ" --> q1((q1))
    q1 -- "0, L; λ" --> q2((q2))
    q2 -- "1, A; λ" --> q3(((q3)))
    q0 -- "L, λ; L" --> q0
    q1 -- "9, λ; λ" --> q1
    q2 -- "1, G; λ" --> q2
    q3 -- "0, C; λ" --> q3
  
```


— □ ×

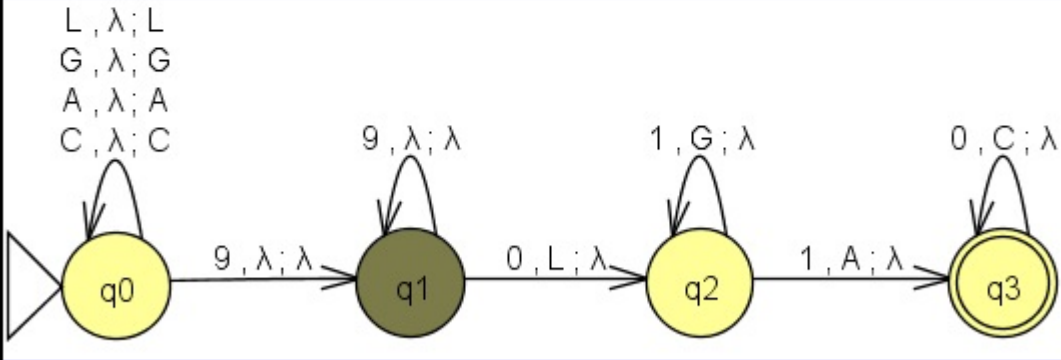
Proyecto segundo parcial

Ingrese su RFC:

Realizar

Estado	Por Leer	Pila
Estado Q0	CAGL990110	
Estado Q0	AGL990110	C
Estado Q0	GL990110	AC
Estado Q0	L990110	GAC
Estado Q0	990110	LGAC
Estado Q1	90110	LGAC
Estado Q1	0110	LGAC


$L, \lambda; L$
 $G, \lambda; G$
 $A, \lambda; A$
 $C, \lambda; C$



```

graph LR
    start(( )) --> q0((q0))
    q0 -- "9, λ; λ" --> q1((q1))
    q1 -- "0, L; λ" --> q2((q2))
    q2 -- "1, A; λ" --> q3(((q3)))
    q0 -- "L, λ; L" --> q0
    q1 -- "G, λ; G" --> q1
    q2 -- "A, λ; A" --> q2
    q3 -- "C, λ; C" --> q3
  
```

Limpiar


— □ ×

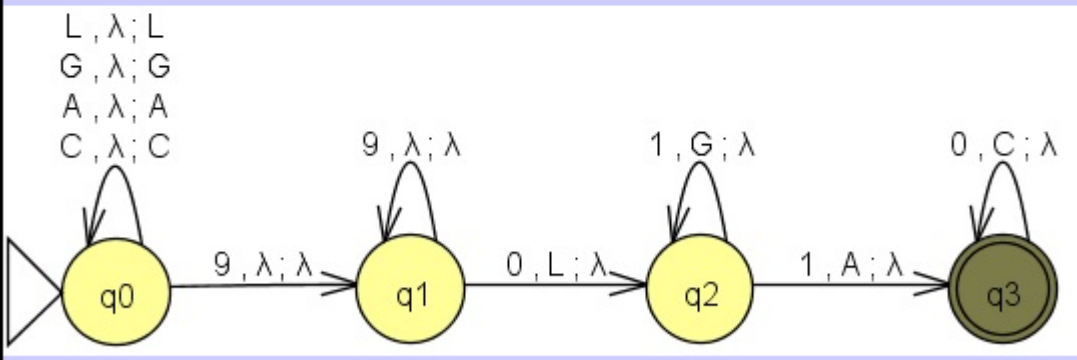
Proyecto segundo parcial

Ingrese su RFC:

Realizar

Estado	Por Leer	Pila
Estado Q0	CAGL990110	
Estado Q0	AGL990110	C
Estado Q0	GL990110	AC
Estado Q0	L990110	GAC
Estado Q0	990110	LGAC
Estado Q1	90110	LGAC
Estado Q1	0110	LGAC
Estado Q2	110	LGAC


$L, \lambda; L$
 $G, \lambda; G$
 $A, \lambda; A$
 $C, \lambda; C$



```

graph LR
    start(( )) --> q0((q0))
    q0 -- "9, λ; λ" --> q1((q1))
    q1 -- "0, L; λ" --> q2((q2))
    q2 -- "1, A; λ" --> q3(((q3)))
    q0 -- "L, λ; L" --> q0
    q1 -- "9, λ; λ" --> q1
    q2 -- "1, G; λ" --> q2
    q3 -- "0, C; λ" --> q3
  
```

Limpiar

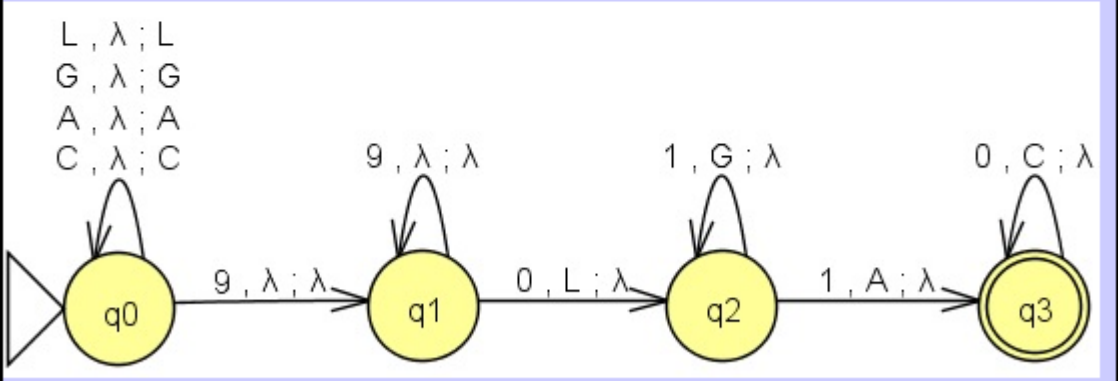

— □ ×

Proyecto segundo parcial

Ingrese su RFC:

Estado	Por leer	Pila

$L, \lambda; L$
 $G, \lambda; G$
 $A, \lambda; A$
 $C, \lambda; C$



```

graph LR
    start(( )) --> q0((q0))
    q0 -- "9, λ; λ" --> q1((q1))
    q1 -- "0, L; λ" --> q2((q2))
    q2 -- "1, A; λ" --> q3(((q3)))
    q0 -- "L, λ; L" --> q0
    q1 -- "9, λ; λ" --> q1
    q2 -- "1, G; λ" --> q2
    q3 -- "0, C; λ" --> q3
  
```

6. Conclusión

El programa se me facilito un poco, aunque el autómata solo funcionara con un RFC especifico, si este RFC llega a cambiar entonces el programa no reconoce el RFC para que pueda hacer el recorrido en el autómata, entonces lo que se tendría que cambiar desde el inicio es el autómata, se tendrá que desarrollar uno que valide dos o más RFC diferentes.

El proyecto se concluyó con lo solicitado que fue el desarrollo de un autómata de pila en un lenguaje de programación, en mi caso ocupe JAVA como lenguaje ya que es el que mas he usado y más fácil se me da.

7. Referencias

<https://www.it-swarm-es.com/es/java/java-como-agregar-una-imagen-jlabel/971051297/#:~:text=Debe%20proporcionar%20a%20JLabel%20una,Image%20image%3DGeneratelnImage.>

<http://foro.chuidiang.org/java-j2se/t3568/>

<http://aprendiendoconjava.blogspot.com/2011/05/como-agregar-una-imagen-un-jlabel.html>

https://www.javamexico.org/foros/java_standard_edition/limpiar_jtable

8. Link del video

<https://www.youtube.com/watch?v=xkiZqCuAheU>