



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

Práctica 3 - Autómata

Unidad de aprendizaje: Teoría computacional

Grupo: 2CM4

Alumno(a):

Nicolás Sayago Abigail

Profesor(a):

Sanchez García Luz María

22 de Marzo de 2018

Índice

1	Introducción	2
2	Planteamiento del problema	3
3	Diseño de la solución	4
4	Implementación de la solución	5
5	Funcionamiento	12
6	Conclusiones	19
	Referencias	19

1. Introducción

En la siguiente práctica se implementara el autómata de una expresión regular en el lenguaje de programación de JAVA. La expresión regular sera de un número de boleta que pertenece a un Sistema Administrativo y Escolar de Educación Básica (SAEEB), que está siendo desarrollado. Dicha expresión regular ya fue explicada en el reporte 2.

A lo largo del documento usaremos la palabra **Autómata**, por lo cual tenemos que definir ¿Qué es un autómata?.

Autómata

La palabra autómata viene del griego *automatos* que significa espontáneo o con movimiento propio.

Autómata programable

En 1938 el matemático norteamericano Claude Elwood Shannon, estableció las bases de la aplicación de la lógica matemática para los circuitos combinatorios, y secuenciales, contruidos a bases de relés y más adelante con otros dispositivos de vacío y estado solido. Los autómatas son sistemas capaces de transmitir información. En amplio sentido, todo sistema que acepta señales de su entorno y, como resultado, cambia de estado y transmite otras señales al medio, puede considerarse un autómata. No tiene sus propios movimientos, sino que estos parecen ser de robot.

Teoría de Autómatas

Para la teoría de la computación existe un área de estudio conocida como la teoría de autómatas, esta estudia las máquinas abstractas y los problemas que éstas son capaces de resolver. Un autómata es un modelo matemático para una máquina de estados, que, dada una entrada de símbolos, salta a través de una serie de estados de acuerdo a una función de transición produciendo salidas. La teoría de autómatas está estrechamente relacionada con la teoría del lenguaje formal ya que los autómatas son clasificados a menudo por la clase de lenguajes formales que son capaces de reconocer.

El autómata recibe los símbolos de entrada, uno detrás de otro, es decir secuencialmente.

Estado de un autómata

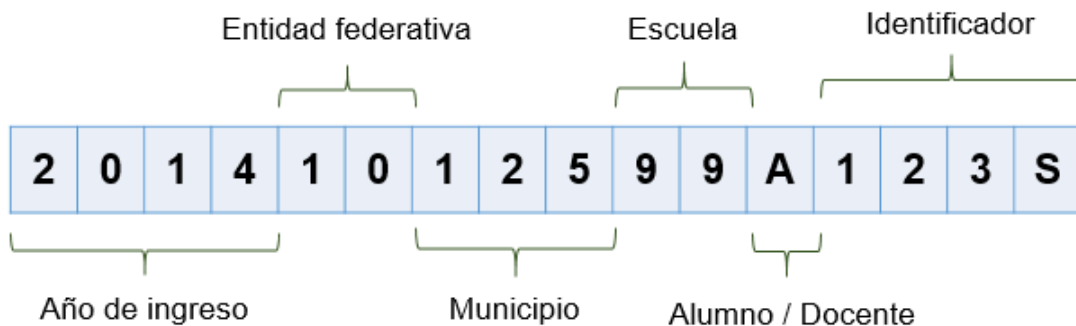
Es toda la información necesaria en eun momento dado, para poder deducir, dado un símbolo de entrada en ese momento, cual será el símbolo de salida.

El autómata tendrá un determinado número de estados(pudiendo ser infinitos), y se encontrará en uno u otro según sea la historia de símbolos que le han llegado.

2. Planteamiento del problema

Recordando que para esta práctica delimitamos el número de boleta para un alumno del Estado de México. Así que diseñamos la solución para implementar el autómata definido por lo siguiente:

- Niños que hayan ingresado en los años 2014-2017.
- Que la escuela pertenezca a la entidad federativa Estado de México.
- Que sea de uno de los 125 municipios de esa entidad federativa.
- Pertenezca a una de las escuelas en ese municipio (0-99)
- Puede ser Alumno o Docente, siendo identificados con A y D respectivamente.
- Número de usuario dado por 3 números y una letra. (Identificador).
- Tiene un total de 16 caracteres.



Algunos ejemplos son:

Cadenas validas son:

- 20141012599A123A
- 20141012599D144A
- 20151000198A126R
- 20171003474A112D

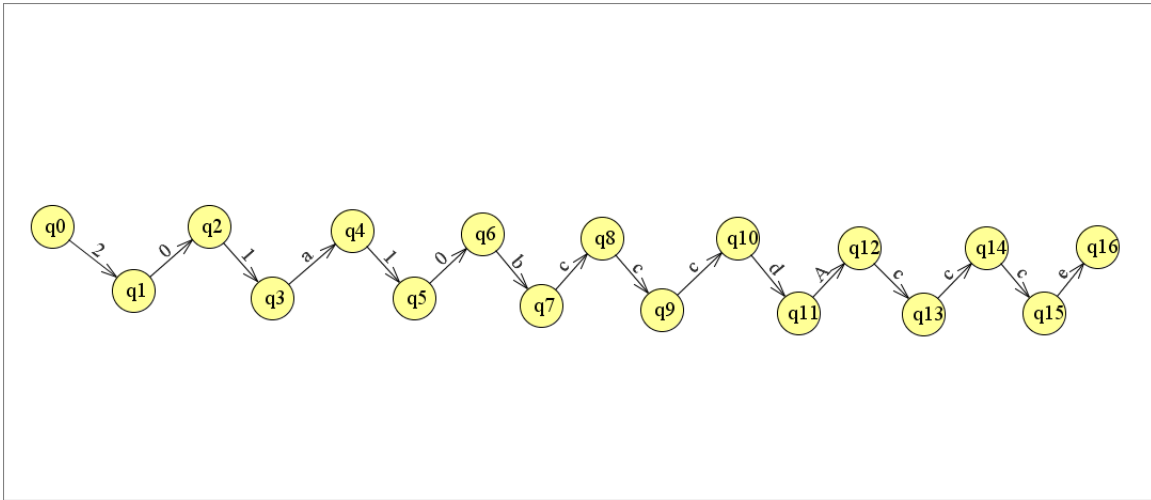
Cadenas no validas:

- 20181312599S123A12
- 20171013515D014S
- 20141012599A131<
- 20131012610D123A

3. Diseño de la solución

A continuación mostraré el diseño que se tiene para implementar la solución, dados los requerimientos expresados antes.

El autómata es:



Donde:

- **a:** Números del 4 al 7
- **b:** Número 0 o 1
- **c:** Números del 0 al 9
- **d:** Números del 1 al 9
- **e:** Letras de la A a la Z

4. Implementación de la solución

De forma general el programa esta estructurado de tal forma que se introduce una cadena, al dar click al boton **OK** el programa empieza a validar cada uno de los estados.

Existen dos archivos, el primero llamado **Main.java** y el segundo **Boleta.java**. A continuación se muestra cada parte de ellos fundamental para que el programa funcione.

IMPLEMENTACIÓN DEL MAIN

```
1 // Recibe los eventos al presionar el boton
2 public void actionPerformed(ActionEvent e)
3 {
4     JButton b = (JButton)e.getSource();
5     // Si se presiona el boton OK
6     if(b == Enviar)
7     {
8         // Se limpia la entrada
9         AreaSalida.setText("");
10        // Se env a la boleta
11        Automata.setBoleta(Cadena.getText());
12        // Se inicia el proceso de evaluar estados
13        Automata.Iniciar();
14        // Se verifica si la cadena es valida
15        if(Automata.Verificar())
16            AreaSalida.setText(AreaSalida.getText() + " Cadena Valida\r\n");
17        else
18            AreaSalida.setText(AreaSalida.getText() + " Cadena NO Valida\r\n");
19    }
20 }
```

IMPLEMENTACIÓN DE LA CLASE BOLETA

→ **Métodos principales y necesarios**

```
1 // Metodo constructor
2 public Boleta(String NumBoleta, JTextArea auxText)
3 {
4     this.NumBoleta = NumBoleta;
5     this.auxText = auxText;
6 }
```

```
1 // Obtiene la boleta
2 public void setBoleta(String NumBoleta)
3 {
4     this.NumBoleta = NumBoleta;
5 }
```

```
1  /* M todo que permite verificar si la cadena cumple
2     con la longitud requerida, en este caso 16. */
3  public boolean Verificar()
4  {
5      if(i==16)
6          return true;
7      else
8          return false;
9  }
```

```
1  /* M todo que inicia la verificacin de estados */
2  public void Iniciar()
3  {
4      i = 0; // Se inicializa la variable contadora
5      q0(); // Empieza el estado 0
6  }
```

→ Estado CERO

```
1  // M todo del ESTADO 0
2  public void q0()
3  {
4      auxText.setText(auxText.getText()+"Q0\r\n");
5      if(i< NumBoleta.length())
6      {
7          // Verifica que sea un n mero 2
8          if('2'==NumBoleta.charAt(i))
9              i++; q1();
10     }
11 }
```

→ Estado UNO

```
1  // M todo del ESTADO 1
2  public void q1()
3  {
4      auxText.setText(auxText.getText()+"Q1\r\n");
5      if(i< NumBoleta.length())
6      {
7          // Verifica que sea un n mero 0
8          if('0'==NumBoleta.charAt(i))
9              i++; q2();
10     }
11 }
```

→ Estado DOS

```
1 // M todo del ESTADO 2
2 public void q2 ()
3 {
4     auxText.setText (auxText.getText () + "Q2\r\n");
5     if (i < NumBoleta.length ())
6     {
7         // Verifica que sea un n mero 1
8         if ('1' == NumBoleta.charAt (i))
9             i++; q3 ();
10    }
11 }
```

→ Estado TRES

```
1 // M todo del ESTADO 3
2 public void q3 ()
3 {
4     auxText.setText (auxText.getText () + "Q3\r\n");
5     if (i < NumBoleta.length ())
6     {
7         int num = Integer.parseInt (" "+NumBoleta.charAt (i));
8         // Cumple el rango de [4-7]
9         if ( (num > 3) && (num < 8) )
10        {
11            i++;
12            q4 ();
13        }
14    }
15 }
```

→ Estado CUATRO

```
1 // M todo del ESTADO 4
2 public void q4 ()
3 {
4     auxText.setText (auxText.getText () + "Q4\r\n");
5     if (i < NumBoleta.length ())
6     {
7         // Verifica que sea un n mero 1
8         if ('1' == NumBoleta.charAt (i))
9             i++; q5 ();
10    }
11 }
```


→ Estado CINCO

```
1 // M todo del ESTADO 5
2 public void q5()
3 {
4     auxText.setText(auxText.getText() + "Q5\r\n");
5     if(i < NumBoleta.length())
6     {
7         // Verifica que sea un n mero 0
8         if('0'==NumBoleta.charAt(i))
9             i++; q6();
10    }
11 }
```

→ Estado SEIS

```
1 // M todo del ESTADO 6
2 public void q6()
3 {
4     auxText.setText(auxText.getText() + "Q6\r\n");
5     if(i < NumBoleta.length())
6     {
7         int num = Integer.parseInt(""+NumBoleta.charAt(i));
8         // Cumple el rango de [0-1]
9         if( (num >= 0) && (num < 2))
10            i++; q7();
11    }
12 }
```

→ Estado SIETE

```
1 // M todo del ESTADO 7
2 public void q7()
3 {
4     auxText.setText(auxText.getText() + "Q7\r\n");
5     if(i < NumBoleta.length())
6     {
7         int num = Integer.parseInt(""+NumBoleta.charAt(i));
8         int numAnt = Integer.parseInt(""+NumBoleta.charAt(i-1));
9         if(numAnt == 0)
10        {
11            // Cumple el rango de [0-9]
12            if( (num >= 0) && (num < 10))
13                i++; q8();
14        }
15        else
16        {
17            // Cumple el rango de [0-2]
18            if( (num >= 0) && (num < 3))
19                i++; q8();
20        }
21    }
22 }
```

→ Estado OCHO

```
1 // M todo del ESTADO 8
2 public void q8()
3 {
4     auxText.setText(auxText.getText() + "Q8\r\n");
5     if(i < NumBoleta.length())
6     {
7         int num = Integer.parseInt(""+NumBoleta.charAt(i));
8         // Cumple el rango de [0-9]
9         if( (num >= 0) && (num < 10))
10             i++; q9();
11     }
12 }
```

→ Estado NUEVE

```
1 // M todo del ESTADO 9
2 public void q9()
3 {
4     auxText.setText(auxText.getText() + "Q9\r\n");
5     if(i < NumBoleta.length())
6     {
7         int num = Integer.parseInt(""+NumBoleta.charAt(i));
8         // Cumple el rango de [0-9]
9         if( (num >= 0) && (num < 10))
10             i++; q10();
11     }
12 }
```

→ Estado DIEZ

```
1 // M todo del ESTADO 10
2 public void q10()
3 {
4     auxText.setText(auxText.getText() + "Q10\r\n");
5     if(i < NumBoleta.length())
6     {
7         int num = Integer.parseInt(""+NumBoleta.charAt(i));
8         // Cumple el rango de [1-9]
9         if( (num >= 1) && (num < 10))
10             i++; q11();
11     }
12 }
```

→ Estado ONCE

```
1 // M todo del ESTADO 11
2 public void q11()
3 {
4     String letra1 = "A", letra2 = "D";
5     char car = NumBoleta.charAt(i);
6     auxText.setText(auxText.getText() + "Q11\r\n");
7     if(i < NumBoleta.length())
8     {
9         if(String.valueOf(car).equals(letra1) || String.valueOf(car).equals(letra2))
10             i++; q12();
11     }
12 }
```

→ Estado DOCE

```
1 // M todo del ESTADO 12
2 public void q12()
3 {
4     auxText.setText(auxText.getText() + "Q12\r\n");
5     if(i < NumBoleta.length())
6     {
7         int num = Integer.parseInt(""+NumBoleta.charAt(i));
8         // Cumple el rango [0-9]
9         if( (num >= 0) && (num < 10))
10             i++; q13();
11     }
12 }
```

→ Estado TRECE

```
1 // M todo del ESTADO 13
2 public void q13()
3 {
4     auxText.setText(auxText.getText() + "Q13\r\n");
5     if(i < NumBoleta.length())
6     {
7         int num = Integer.parseInt(""+NumBoleta.charAt(i));
8         // Cumple el rango [0-9]
9         if( (num >= 0) && (num < 10))
10             i++; q14();
11     }
12 }
```

→ Estado CATORCE

```
1 // M todo del ESTADO 14
2 public void q14()
3 {
4     auxText.setText(auxText.getText() + "Q14\r\n");
5     if(i < NumBoleta.length())
6     {
7         int num = Integer.parseInt(""+NumBoleta.charAt(i));
8         // Cumple el rango [0-9]
9         if( (num >= 0) && (num < 10))
10             i++; q15();
11     }
12 }
```

→ Estado QUINCE

```
1 // M todo del ESTADO 15
2 public void q15()
3 {
4     auxText.setText(auxText.getText() + "Q15\r\n");
5     char[] abecedario;
6     int j;
7     if (i < NumBoleta.length())
8     {
9         abecedario = new char[26];
10        for(j=0; j<26; j++)
11            abecedario[j] = (char) ('A' + j);
12        for(j=0; j<26; j++)
13        {
14            if(NumBoleta.charAt(i) == abecedario[j])
15                j=26; q16(); i++;
16        }
17    }
18 }
```

→ Estado DIEZ Y SEIS

```
1 // M todo del ESTADO 16 - FIN -
2 public void q16()
3 {
4     auxText.setText(auxText.getText() + "Q16\r\n");
5     if(NumBoleta.length()>16)
6         i=17;
7 }
```

5. Funcionamiento

Primero que nada, mostramos la interfaz inicial. Observamos que el usuario tiene la oportunidad de ingresar una cadena. Al dar click en el botón OK, se mostrarán los estados por los que pasa la cadena para mostrar si es una cadena valida o no.

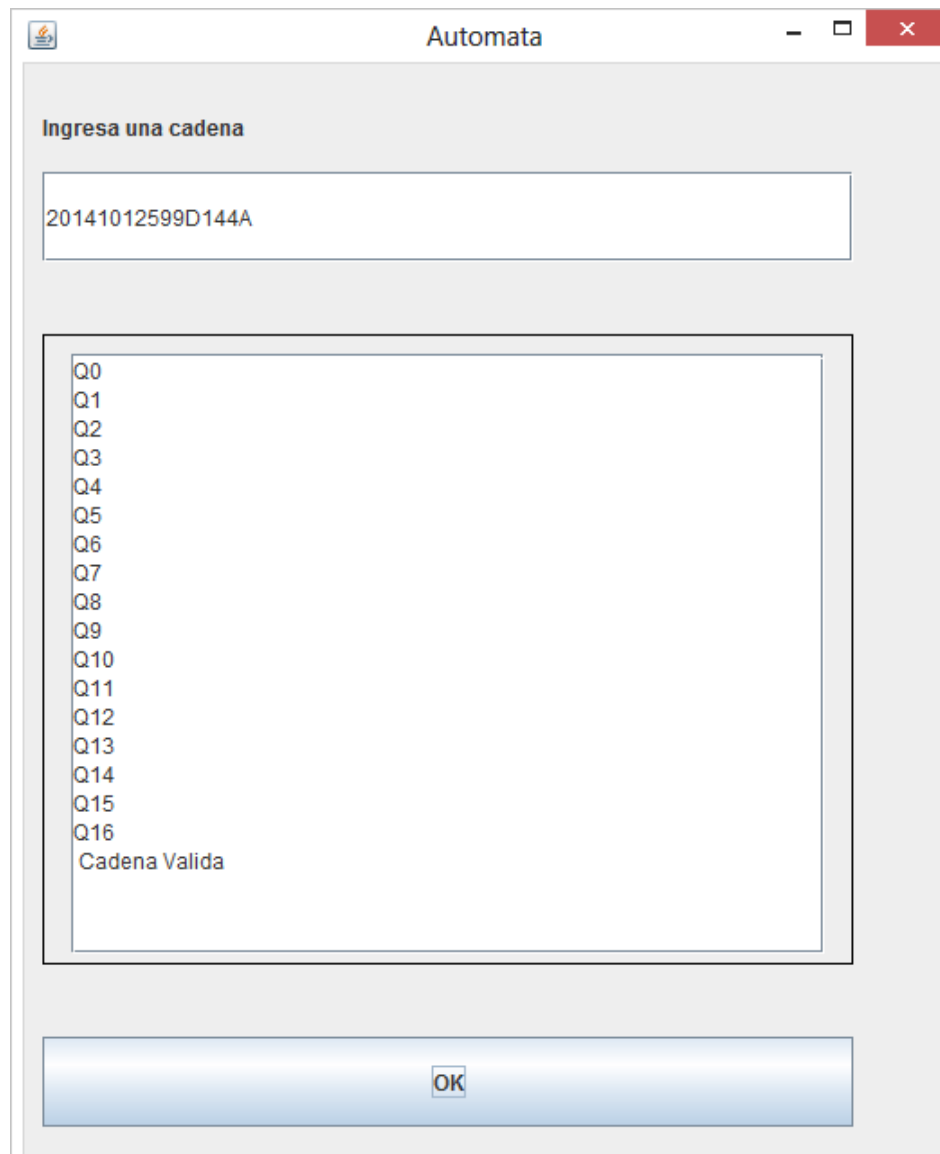
Se muestra **CADENA VALIDA** si la cadena que se ha ingresado cumple con el formato del número de boleta en caso contrario, se mostrará **CADENA NO VALIDA**.

La pantalla inicial es la siguiente:



A continuación muestro ejemplos de **cadenas validas**:

The screenshot shows a Java Swing window titled "Automata". Inside the window, there is a label "Ingresa una cadena" above a text input field containing the string "20141012599A123A". Below the input field is a large rectangular area containing a list of states: Q0, Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9, Q10, Q11, Q12, Q13, Q14, Q15, and Q16. At the bottom of this list area, the text "Cadena Valida" is displayed. At the very bottom of the window is a blue "OK" button.



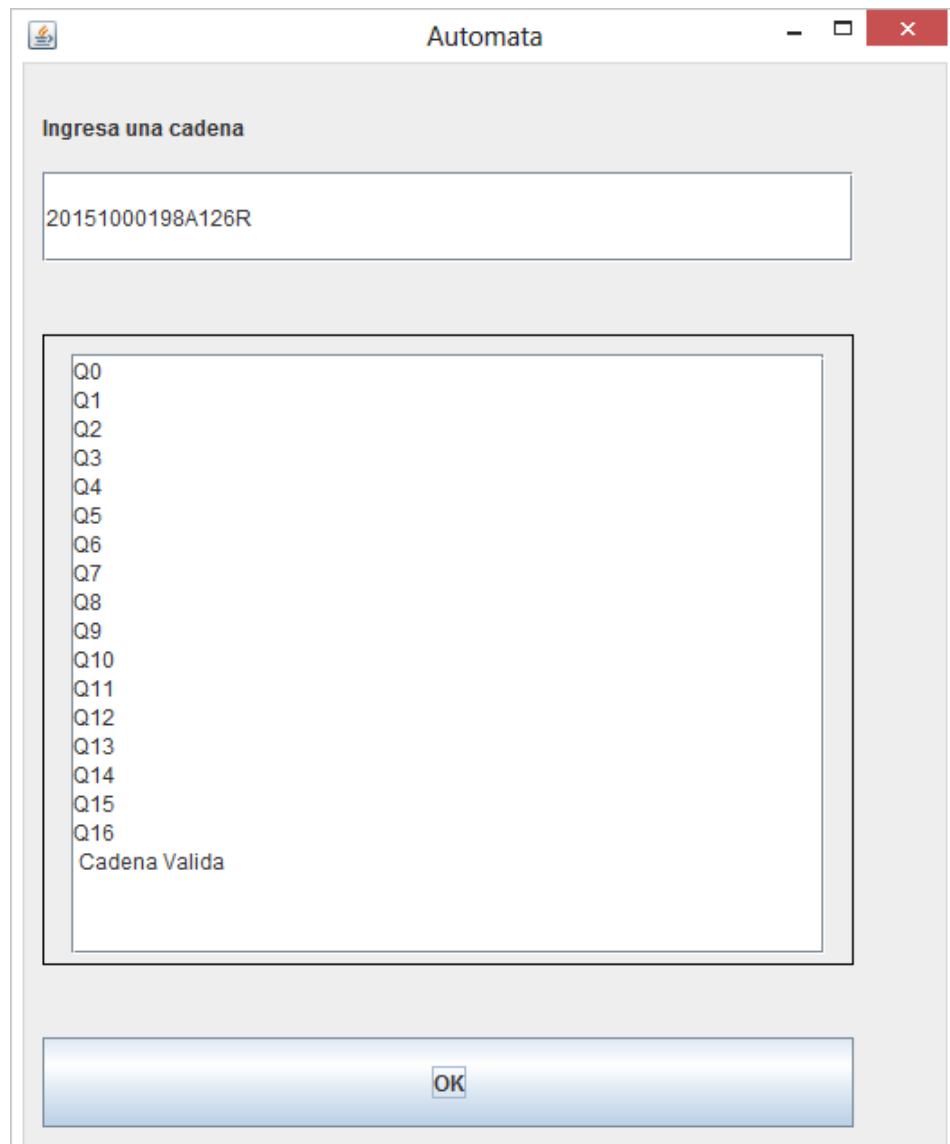
The image shows a Java Swing window titled "Automata". Inside the window, there is a label "Ingresa una cadena" (Enter a string). Below this label is a text input field containing the string "20141012599D144A". Below the input field is a large rectangular area containing a list of states: Q0, Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9, Q10, Q11, Q12, Q13, Q14, Q15, and Q16. Below the list is the text "Cadena Valida". At the bottom of the window is a blue button with the text "OK".

Ingresa una cadena

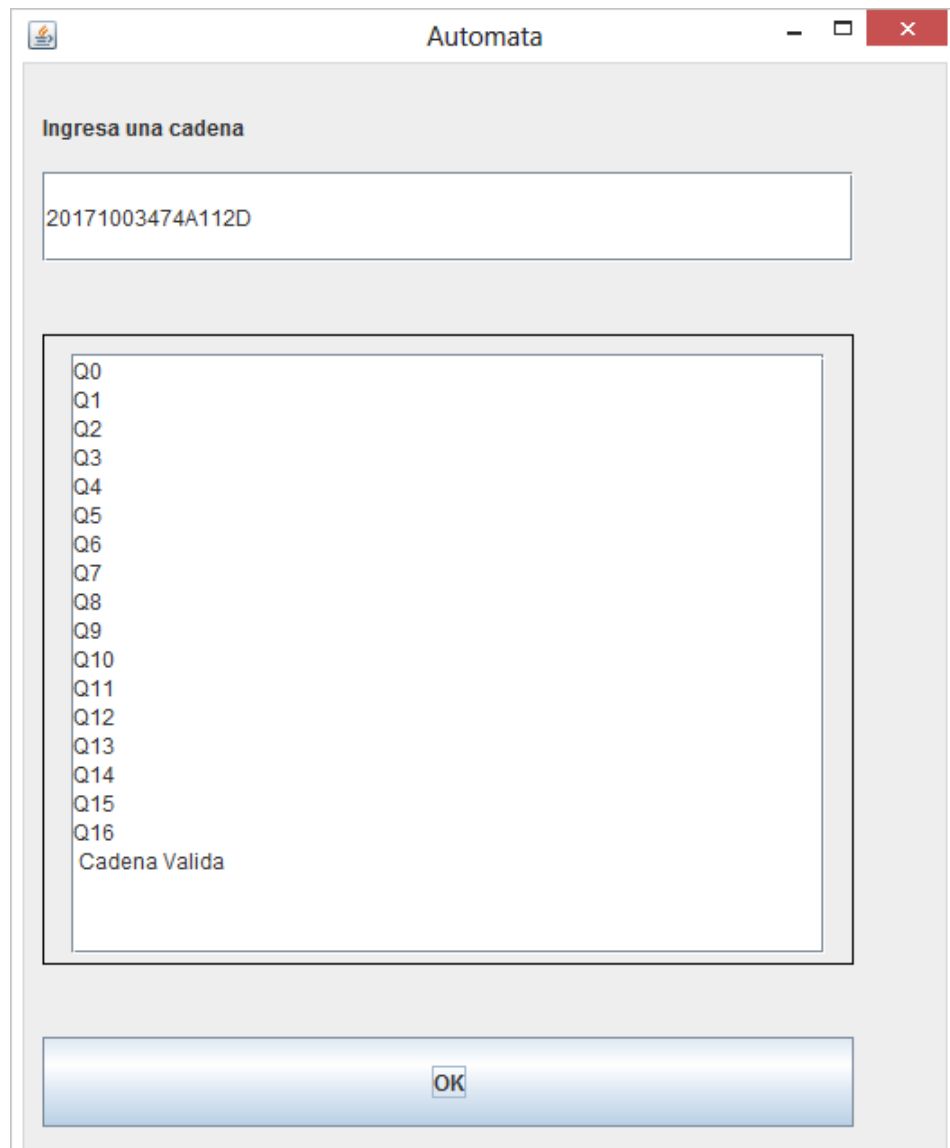
20141012599D144A

Q0
Q1
Q2
Q3
Q4
Q5
Q6
Q7
Q8
Q9
Q10
Q11
Q12
Q13
Q14
Q15
Q16
Cadena Valida

OK

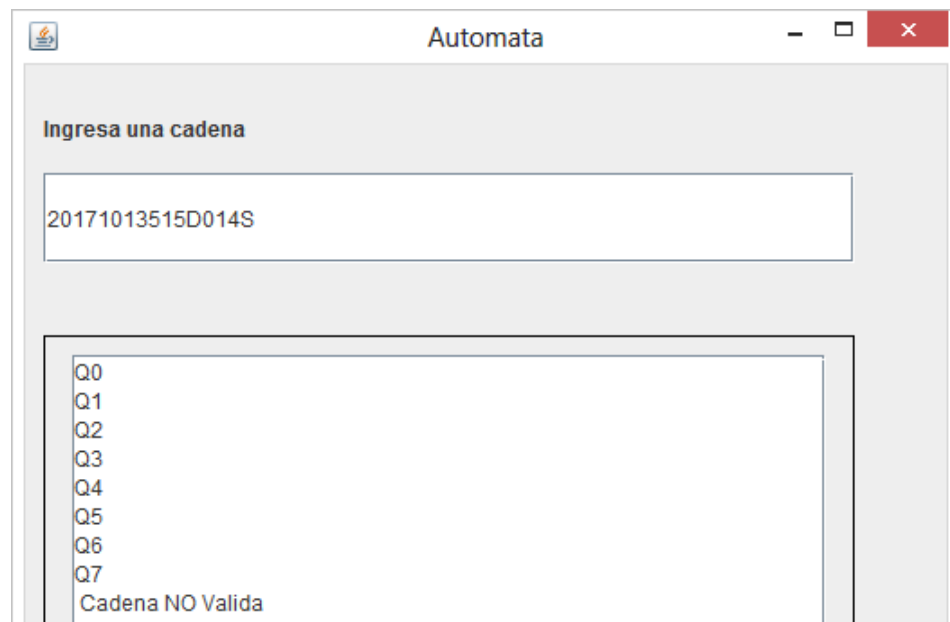


The image shows a Java Swing window titled "Automata". Inside the window, there is a label "Ingresa una cadena" (Enter a string). Below this label is a text input field containing the string "20151000198A126R". Below the input field is a large rectangular area containing a list of states: Q0, Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9, Q10, Q11, Q12, Q13, Q14, Q15, and Q16. Below the list, the text "Cadena Valida" (Valid String) is displayed. At the bottom of the window is a blue button labeled "OK".



The image shows a Java Swing window titled "Automata". Inside the window, there is a label "Ingresa una cadena" (Enter a string). Below this label is a text input field containing the string "20171003474A112D". Below the input field is a large rectangular area containing a list of states: Q0, Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9, Q10, Q11, Q12, Q13, Q14, Q15, and Q16. Below the list, the text "Cadena Valida" (Valid String) is displayed. At the bottom of the window is a blue button with the text "OK".

Ahora ejemplos de **cadenas no validas**:



Automata

Ingresa una cadena

20141012599A131<

Q0
Q1
Q2
Q3
Q4
Q5
Q6
Q7
Q8
Q9
Q10
Q11
Q12
Q13
Q14
Q15
Cadena NO Valida

Automata

Ingresa una cadena

20131012610D123A

Q0
Q1
Q2
Q3
Cadena NO Valida

6. Conclusiones

Al terminar esta práctica pude llevar a cabo la implementación del autómata que esta basado en el número de boleta de un alumno del estado de México que cursa la escuela secundaria, dicho número de boleta esta propuesto desde la práctica anterior. Específicamente, esta práctica me ha ayudado a observar de manera más directa como se comporta un autómata, y que es mejor programarlo por estados.

Inicialmente mi idea era usar muchos *if* para que se validará la cadena, pero cuando iba a la mitad me di cuenta que estaba trabajando de más y que no era la forma más eficiente de llevarlo a cabo, entonces hice la implementación que fue mostrada anteriormente.

Referencias

- [1] E. A. Martinez, *Alfabetos, símbolos y cadenas*. [Online]. Available: <http://eafranco.com/docencia/teoriacomputacional/files/03/Clase15.pdf>