



**INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO**

TEORÍA COMPUTACIONAL

2CM4

PROFESOR: LUZ MARÍA SÁNCHEZ GARCÍA

PRÁCTICA 3 AUTÓMATA FINITO DETERMINISTA

VÁZQUEZ MORENO MARCOS OSWALDO

2016601777



FECHA DE ENTREGA: 01 DE ABRIL DE 2018

INTRODUCCIÓN

En la siguiente práctica se pretende realizar un programa en lenguaje de programación Java, la cual valide expresiones regulares, en este caso la expresión es un *Hashtag*, en la cual se pretende que no pueda aceptar espacios, siendo rigorista en hacer uso forzosamente del “#” al inicio, seguido de cualquier combinación entre a-z (mayúsculas o minúsculas), 0-9 o “_”.

Quedando de la siguiente manera la Expresión Regular `//#[a-zA-Z0-9]*`.

La implementación como el desarrollo de la idea es sencilla pero eficiente.

Aunado a esto se menciona que la validación no será con la expresión completa sino validación estado a estado, por fortuna mi programa debe de ser de un *hashtag* y no hay tanto rigor en la validación

PLANTEAMIENTO DEL PROBLEMA

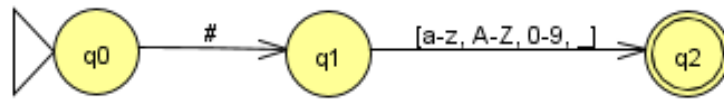
Implementar el algoritmo de codificación de un programa el cual acepte una cadena de caracteres de tipo *Hashtag* haciendo una correcta validación respecto a las siguientes restricciones:

- ✓ Iniciar siempre con el símbolo de almohadilla “#”.
- ✓ Usar cualquier letra del abecedario sin importar si son mayúsculas o minúsculas, además en cualquier combinación.
- ✓ Usar cualquier dígito del 0 al 9 en cualquier combinación.
- ✓ Es permitido más no obligatorio el uso del guion bajo “_”.
- ✓ No es permitido usar cualquier otro tipo de caracteres que no sea guion bajo y almohadilla.
- ✓ No es permitido usar espacios dentro de la cadena.
- ✓ No es permitido usar acentos.
- ✓ La longitud de la cadena no importa.

Esta vez retomando la práctica número 2, ahora se hará la validación estado por estado en su correcta transición

DISEÑO Y FUNCIONAMIENTO DE LA SOLUCIÓN

Siendo el autómata finito:



Una vez con la problemática planteada se necesitaba de poder darle sentido a la solución y es por eso por lo que se decidió crear un programa desarrollado en lenguaje Java utilizando unas librerías y unas palabras reservadas capaz de admitir una expresión regular y llevarla a su validación.

Se llevará a cabo dentro del entorno gráfico de Java en un JFrame en el cual habrá una Etiqueta con las instrucciones, un JButton el cual hará la validación de que el Hashtag insertado en el campo de texto sea correcto, de ser valida la cadena arrojará un mensaje a otra etiqueta diciendo “VALIDA” y de lo contrario diciendo “NO VALIDA”.

IMPLEMENTACIÓN DE LA SOLUCIÓN

```

/* Hecho por Vázquez Moreno Marcos Oswaldo
Boleta 2016601777
El 16 de Marzo de 2018
Materia Teoría Computacional
Práctica número 3 AUTOMATA FINITO DETERMINISTA
Fecha de inicio 01 de Abril
Fecha de terminación 01 de Abril
*/

//Librerías
import java.awt.event.*;
import javax.swing.*;
import java.awt.*;
import java.util.regex.*;
import java.util.regex.Pattern.*;
import java.util.regex.Matcher.*;

//Declaración de la clase pública Lenguajes que se mostrará en un JFrame con la implementación de una acción a algún botón
public class Lenguajes extends JFrame implements ActionListener{
    //Declaración de la variables de tipo botón, Campo de escritura, Etiqueta, Contenedor
    JButton evaluar;
    JTextField campo;
    JLabel instrucciones, resultado;
    Container c;
    //Constructor
    public Lenguajes () {
        //Declaración del contenedor
        c = getContentPane ();
        //creador del título
        setTitle ("HASHTAG");
        //texto dentro de la etiqueta y asignación
        instrucciones = new JLabel ("INGRESE UN HASHTAG SIN ESPACIOS NI CARACTERES, RECONDANDO QUE PUEDE USAR TANTO LETRAS MINUSCULAS COMO MAYUSCULAS");
        //Texto dentro del botón, agregandolo al Frame a la izquierda
        evaluar = new JButton ("Evaluar Hashtag"); c.add("West", evaluar);
    }
  
```

```

evaluar = new JButton ( "evaluar hashtag", c.add("West", evaluar);
//Acción cuando se le de clic para evaluar
evaluar.addActionListener (this);
//Campo para insertar la cadena, en este caso el Hashtag
campo = new JTextField (5); c.add("Center", campo);
//Etiqueta en donde dirá si es valida o no
resultado = new JLabel (""); c.add("South", resultado);
setVisible(true); setSize(850, 150); setResizable(true);
}
//clase para que al hacer clic en el botón se pueda validar la expresión regular
public void actionPerformed ( ActionEvent e ){
    JButton b=(JButton)e.getSource();
    resultado.setIcon( b.getIcon() );
    //palabra reservada de Java en la que puedes insertar expresiones regulares
    //Aquí dentro de los paréntesis colocas tu E.R. siendo forzoso el # seguido de
    //cualquier combinación de la a-z o A-Z o 0-9 o _
    Pattern p = Pattern.compile ("#[a-zA-Z0-9]*");
    Matcher m = p.matcher (campo.getText());
    //validador si es correcta
    if (m.matches ()){
        resultado.setText ("VALIDA");
    }
    else if(m.matches () == false) { //caso contrario para el estado q0 al q1
        resultado.setText ("DEL ESTADO Q0 AL Q1 SOLO SE PUEDE USAR ----> # ");
    }
    if (m.matches ()){
        resultado.setText ("VALIDA");
    }
    else { //caso contrario para la validación del q1 hasta qn ya que no hay límite
        resultado.setText ("DEL ESTADO Q1 AL QN SOLO SE PUEDE USAR ----> A-Z o a-z o 0-9");
    }
}

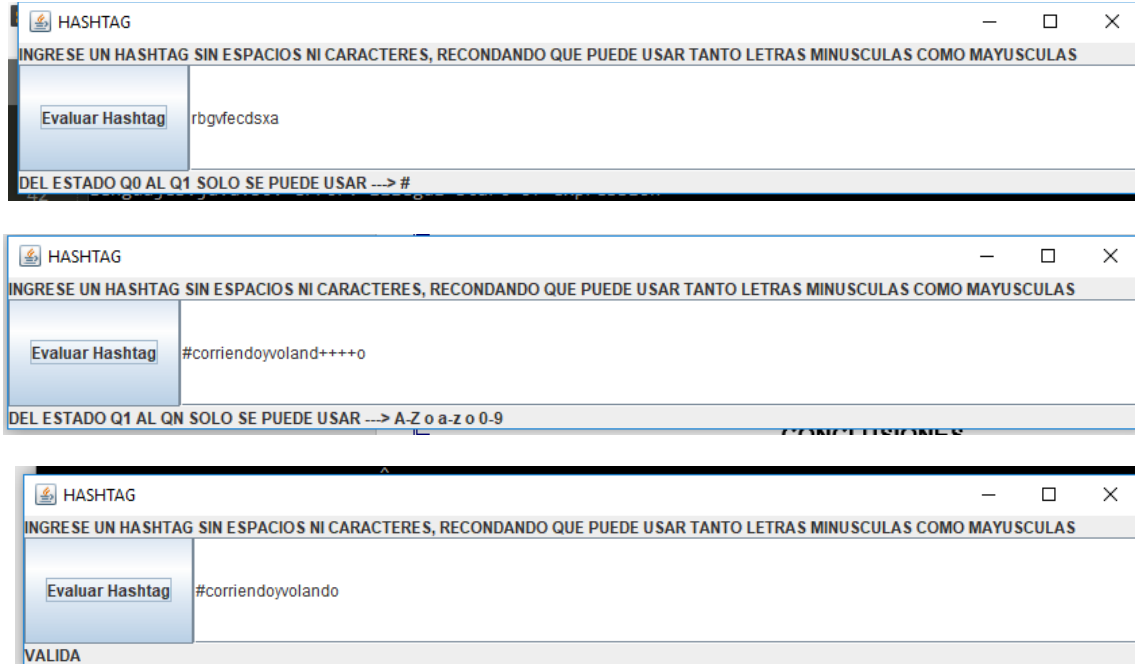
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

//Etiqueta en donde dirá si es valida o no
resultado = new JLabel (""); c.add("South", resultado);
setVisible(true); setSize(850, 150); setResizable(true);
}
//clase para que al hacer clic en el botón se pueda validar la expresión regular
public void actionPerformed ( ActionEvent e ){
    JButton b=(JButton)e.getSource();
    resultado.setIcon( b.getIcon() );
    //palabra reservada de Java en la que puedes insertar expresiones regulares
    //Aquí dentro de los paréntesis colocas tu E.R. siendo forzoso el # seguido de
    //cualquier combinación de la a-z o A-Z o 0-9 o _
    Pattern p = Pattern.compile ("#[a-zA-Z0-9]*");
    Matcher m = p.matcher (campo.getText());
    //validador si es correcta
    if (m.matches ()){
        resultado.setText ("VALIDA");
    }
    else if(m.matches () == false) { //caso contrario para el estado q0 al q1
        resultado.setText ("DEL ESTADO Q0 AL Q1 SOLO SE PUEDE USAR ----> # ");
    }
    if (m.matches ()){
        resultado.setText ("VALIDA");
    }
    else { //caso contrario para la validación del q1 hasta qn ya que no hay límite
        resultado.setText ("DEL ESTADO Q1 AL QN SOLO SE PUEDE USAR ----> A-Z o a-z o 0-9");
    }
}

setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
public static void main (String s[]){
    Lenguajes ex = new Lenguajes ();
}
}

```

FUNCIONAMIENTO



CONCLUSIONES

En conclusión, la práctica número 3 es una extensión y mejora de la práctica número 2 y ya que la número 2 me agradó bastante, la número 3 también, además me ayudó más que nada a practicar de nuevo mi destreza en la programación ahora en un lenguaje nuevo de programación para mí el cual es Java y tengo poco de usarlo

Por otro lado, es una práctica bastante útil ya que la finalidad es comprender cómo funcionan las expresiones regulares y su correcta implementación, se cree que no se necesita el problema más difícil para comprender lo que se ve en la clase, lo cual ha sido bastante bueno y este tipo de códigos e implementación retroalimentan de una muy buena forma lo que aprendemos en el salón de clases, con una excelente profesora.

Me agrada que las prácticas han sido de nivel pero con un gran aprovechamiento educacional, no como los profesores que exigen demasiado en las prácticas y realmente no tiene gran trascendencia.

BIBLIOGRAFÍA

Software de aplicación JFLAP: <http://www.jflap.org/>

Copyright © 1993, 2017, Oracle and/or its affiliates. All rights reserved. Use is subject to license terms.

<https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html>