TEORÍA COMPUTACIONAL
Ejemplos de Programas para Expresiones Regulares

//Programa que valida un salón de ESCOM

```
package expresionregular;

import java.util.Scanner;
import java.util.regex.Matcher; // Un objeto de esta clase representa la expresión regular.
import java.util.regex.Pattern; // Esta clase compara el String y la expresión regular.

public class ExpresionRegular {
   public static void main(String[] args) {
     Scanner sc = new Scanner(System.in);
     String salon;
     System.out.print("Introduce un salón de ESCOM: ");
     salon = sc.nextLine();
     Pattern pat = Pattern.compile("^[1-2][0-2]((0[1-7]|9)|(1[0-4]))$");
     Matcher mat = pat.matcher(salon);
     if(mat.find()){    //El método find() indica si el String contienen el patrón.
       System.out.println("Salon Valido");
     }else{
       System.out.println("Salon No Valido");
    }
  }
}
```

//Programa que valida una dirección IP

```
package direccionip;
import java.util.regex.*;
import javax.swing.JOptionPane;

public class DireccionIP {
      public static void main(String[] args) {

       String cadena=JOptionPane.showInputDialog("Escriba una direccion ip");
       System.out.println("Cadena a evaluar:"+cadena);

       String c="0.0.0.0";
       Pattern direccionIp= Pattern.compile("^(([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5]).){3}([1-9]?[0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])$");
       Matcher encaja= direccionIp.matcher(cadena);
       if(encaja.find())
          System.out.println("Es una dirección ip correcta");
       else{
          System.out.println("No es una direccion ip ");
       }
  }
```

//Programa que valida un RFC (Registro Federal de Contribuyentes)

```
package main;

import java.io.*;
import java.util.Vector;
import java.util.regex.*;

class RFC{
   public static void main(String args[]){
        Vector cadenas=new Vector();
        String cadena;
        int i;
        System.out.println("\nEscribe las cadenas a validar y 0 para finalizar la captura\n");
        while(true){
           cadena=leerCadena();
           if(!cadena.equals("0"))
             cadenas.add(cadena);
           else
              break;
        }
        for(i=0;i<cadenas.size();i++){
           System.out.println(cadenas.elementAt(i)+"    "+(validaRFC(cadenas.elementAt(i).toString())==true?"Correcto :)":"Incorrecto :/"));
        }
   }
   static String leerCadena(){
        String cad;
        try{
          InputStreamReader isr= new InputStreamReader(System.in);
```

```
            BufferedReader in =new BufferedReader(isr);
            cad=in.readLine();
          }catch(IOException e){
            System.out.println("Error: "+e.getMessage());
            cad=null;
          }
          return cad;
    }
    static Boolean validaRFC(String cad){
        Pattern pat=Pattern.compile("^[A-Z]{4}[0-9]{2}(0[1-9]{1}|1[1-2]{1})(0[1-9]{1}|[1-2][0-9]{1}|3[0-1]{1})[A-Z0-9]{3}$");
        Matcher mat = pat.matcher(cad);
        return mat.find();
    }

}
```