Método Main.

```java
import java.util.Scanner;
// @author LuisR
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(source: System.in);
        System.out.print(s: "Ingrese la cadena a evaluar (Solo 0 y 1): ");
        String w = sc.nextLine();
        funcionTransicion afd = new funcionTransicion();
        int n = 0;
        String q = afd.getEstadoInic();
        String[] conjRes = null;
        while (n < w.length()) {
            if (n != 0) {
                conjRes = afd.d(funTr: conjRes, s: w.substring(beginIndex: n, n + 1));
            } else {
                conjRes = afd.funcionInic(q, s: w.substring(beginIndex: n, n + 1));
            }
            n++;
        }
        System.out.println(x: afd.evaluar(res:conjRes));
    }
}
```

Clase AFN.

```java
import java.util.ArrayList;
import java.util.HashMap;
// @author LuisR
public class AFD {
    private final String q;
    private final String[] conjEst = {"q0", "q1", "q2", "q3", "q4"};
    private final String[] conjFin = {conjEst[2], conjEst[4]};
    private final String[] q0_0 = {conjEst[0], conjEst[3]};
    private final String[] q0_1 = {conjEst[0], conjEst[1]};
    private final String[] q1_0 = null;
    private final String[] q1_1 = {conjEst[2]};
    private final String[] q2_0 = {conjEst[2]};
    private final String[] q2_1 = {conjEst[2]};
    private final String[] q3_0 = {conjEst[4]};
    private final String[] q3_1 = null;
    private final String[] q4_0 = {conjEst[4]};
    private final String[] q4_1 = {conjEst[4]};
    private final HashMap<String, String[]> funTran = new HashMap<>();
```

```java
    public AFD() {
        q = conjEst[0];
        funTran.put(key:"q0-0", value: q0_0);
        funTran.put(key:"q0-1", value: q0_1);
        funTran.put(key:"q1-0", value: q1_0);
        funTran.put(key:"q1-1", value: q1_1);
        funTran.put(key:"q2-0", value: q2_0);
        funTran.put(key:"q2-1", value: q2_1);
        funTran.put(key:"q3-0", value: q3_0);
        funTran.put(key:"q3-1", value: q3_1);
        funTran.put(key:"q4-0", value: q4_0);
        funTran.put(key:"q4-1", value: q4_1);
    }

    public String getEstadoInic() {
        return q;
    }

    public String[] funcionInic(String q, String s) {
        return funTran.get(q + "-" + s);
    }

    public String evaluar(String[] res) {
        boolean band = false;
        if (res != null) {
            for (int i = 0; i < conjFin.length; i++) {
                for (int j = 0; j < res.length; j++) {
                    if (conjFin[i].equals(res[j])) {
                        band = true;
                        break;
                    }
                }
                if (band) {
                    break;
                }
            }
        }
        if (band) {
            return "La cadena es aceptada";
        } else {
            return "La cadena no es aceptada";
        }
    }

    public String[] d(String[] funTr, String s) {
        String[] conjunto;
        ArrayList<String[]> temp = new ArrayList<>();
        for (int i = 0; i < funTr.length; i++) {
```

```
68                  temp.add(e: funcionInic(funTr[i], s));
69              }
70              if (temp.size() < 2) {
71                  conjunto = temp.get(index: 0);
72              } else {
73                  conjunto = unionEstados(temp);
74              }
75              return conjunto;
76          }
77
78          private String[] unionEstados(ArrayList<String[]> temp) {
79              ArrayList<String> estRes = new ArrayList<>();
80              for (int i = 0; i < temp.size(); i++) {
81                  String[] estados = temp.get(index: i);
82                  if (estados != null) {
83                      for (int j = 0; j < estados.length; j++) {
84                          estRes = ingresarElemento(estRes, estados[j]);
85                      }
86                  }
87              }
88              return convArrayObjToString(array: estRes.toArray());
89          }
90
91          private ArrayList<String> ingresarElemento(ArrayList<String> estRes, String q) {
92              boolean existe = estRes.contains(o: q);
93              if (!existe) {
94                  estRes.add(e: q);
95              }
96              return estRes;
97          }
98
99          private String[] convArrayObjToString(Object[] array) {
100             String[] arrayCadenas = new String[array.length];
101             for (int i = 0; i < array.length; i++) {
102                 arrayCadenas[i] = array[i].toString();
103             }
104             return arrayCadenas;
105         }
106 }
```