

Teoría de la Información y Codificación
**Práctica 0: Toma de Contacto con la
Asignatura**

José A. Montenegro Montes

23 de octubre de 2011

1. Enunciado

En esta tarea vamos a codificar cinco tipos de algoritmos de detección de errores: NIF, ISBN, ISBN13, Código de producto y Código Control número de cuenta bancaria.

Los algoritmos para crear los códigos de control mencionados son proporcionados en las transparencias de la asignatura y adjuntos en la última sección de las prácticas, menos el Código Control número de cuenta bancaria.

La práctica consta de la creación de una clase abstracta de codificación, y la implementación de los cinco algoritmos solicitados.

Una vez implementados el alumno debe incluir métodos en las clases:

- Creación de un código control para un mensaje dado (*generarCodigoControl*).
- Verificación mensaje es correcto según el código control (*verificar*).
- Corrección del código control erróneo de un mensaje dado (*corregirControl*).
- Corrección del bit erróneo de un mensaje (solo un error posible), dado un código control correcto (*corregirControl*).

En el caso concreto del ISBN-10 sabemos que el código de control es mod 11, que ocurre si el código de control que obtenemos es 10. ¿Cómo lo incluimos en el código?.

2. Conclusiones

El objetivo de esta práctica es tomar contacto con este tipo de algoritmos y observar las ventajas y desventajas que nos presentan estos algoritmos.

Se abre el debate si son códigos correctores o simplemente códigos detectores de errores.

A lo largo del curso veremos una variedad de algoritmos tanto para la detección como para la corrección de errores, cuya calidad puede medirse en los número de bits que detecta / corrige.

3. Código

Clase Codificación

```
1  /*
2   * To change this template, choose Tools / Templates
3   * and open the template in the editor.
4   */
5
6  package practica0;
7
8  public abstract class Codificacion {
9
10     /**
11      * Comprueba si el código es válido.
12      *
13      * @param codigo    Código a comprobar.
14      * @return          Verdadero si es código es válido, falso en otro caso.
15      */
16     public abstract boolean verificar(String codigo);
17
18     /**
19      * Devuelve el código válido a partir del código incorrecto que recibe como parámetro.
20      *
21      * @param codigo    Código incorrecto.
22      * @return          Código corregido.
23      */
24     public String corregirControl(String codigo){
25         if (verificar(codigo)) return codigo;
26         else return generarCodigoControl(codigo);
27     }
28
29
30     /**
31      * Genera el código control a partir de los datos
32      * @param codigo
33      * @return
34      */
35     public abstract String generarCodigoControl(String codigo);
36 }
```

```

37  /* *****
38   * Supuestamente el código de control está bien buscar donde es el error.
39   * @param codigo
40   * @return
41   */
42  public abstract String[] corregirDatos(String codigo);
43  }

```

Clase ISBN10 Java.

```

1  /*
2   * To change this template, choose Tools / Templates
3   * and open the template in the editor.
4   */
5
6  package practica0;
7
8  /**
9   *
10  * @author
11  */
12 public class ISBN extends Codificacion {
13
14     private static final int MODULO = 11;
15
16     /* *****
17      *
18      * @param codigo
19      * @return
20      */
21     public boolean verificar(String codigo) {
22         codigo = codigo.replaceAll("-", "");
23         int resultado = 0;
24         try {
25             for (int i = 0; i < codigo.length(); i++) {
26                 resultado += Integer.parseInt(codigo.substring(i, i+1))*(i+1);
27             }
28         } catch (NumberFormatException e) {
29             return false;
30         }
31         return resultado % MODULO == 0;
32     }
33
34     /* *****

```

```

35  *
36  * @param codigo
37  * @return
38  */
39  @Override
40  public String generarCodigoControl(String codigo) {
41
42      codigo = codigo.replaceAll("-", "");
43
44      int resultado = 0;
45
46      try {
47          for (int i = 0; i < codigo.length(); i++) {
48              resultado += (Integer.parseInt(codigo.substring(i, i+1))*(i+1));
49          }
50      } catch (NumberFormatException e) {
51          return null;
52      }
53      return codigo+(resultado% MODULO);
54  }
55
56
57
58  @Override
59  public String[] corregirDatos(String codigo) {
60      throw new UnsupportedOperationException("Not supported yet.");
61  }
62
63
64  }

```

Clase ISBN13 Java.

```

1  /*
2  * To change this template, choose Tools / Templates
3  * and open the template in the editor.
4  */
5
6  package practica0;
7
8  /**
9   *
10  * @author monte
11  */

```

```

12 public class ISBN13 extends Codificacion{
13
14     private static final int MODULO = 10;
15
16     public boolean verificar(String codigo) {
17         codigo = codigo.replaceAll("-", "");
18         int resultado = 0;
19         try {
20             for (int i = 0; i < codigo.length(); i++) {
21                 if (i%2==0)
22                     resultado += (Integer.parseInt(codigo.substring(i, i+1))*1);
23                 else
24                     resultado += (Integer.parseInt(codigo.substring(i, i+1))*3);
25             }
26         } catch (NumberFormatException e) {
27             return false;
28         }
29         return resultado % MODULO == 0;
30     }
31
32
33     public String generarCodigoControl(String codigo) {
34
35         codigo = codigo.replaceAll("-", "");
36
37         int resultado = 0;
38
39         try {
40             for (int i = 0; i < codigo.length(); i++) {
41                 if (i%2==0)
42                     resultado += (Integer.parseInt(codigo.substring(i, i+1))*1);
43                 else
44                     resultado += (Integer.parseInt(codigo.substring(i, i+1))*3);
45             }
46         } catch (NumberFormatException e) {
47             return null;
48         }
49
50         resultado = resultado % MODULO;
51         resultado = (MODULO -resultado)%MODULO;
52         return codigo+resultado;
53     }
54
55
56     public String[] corregirDatos(String codigo) {
57         throw new UnsupportedOperationException("Not supported yet.");

```

```

58     }
59
60 }

```

Clase NIF Java.

```

1  /*
2  * To change this template, choose Tools / Templates
3  * and open the template in the editor.
4  */
5
6  package practica0;
7
8  /**
9   *
10  * @author
11  */
12  public class NIF extends Codificacion {
13
14
15      private static final String NIF_TABLA = "TRWAGMYFPDXBNJZSQVHLCKE";
16
17
18
19
20      @Override
21      public boolean verificar(String nif) {
22          int dni = Integer.parseInt(nif.substring(0, nif.length()-1));
23          try {
24              return NIF_TABLA.charAt(dni % NIF_TABLA.length()) == nif.charAt(nif.length()-1);
25          } catch (NumberFormatException e) {
26              return false;
27          }
28      }
29
30
31      /**
32       *
33       * @param codigo
34       * @return
35       */
36      public String generarCodigoControl(String codigo) {
37
38          if (codigo!=null){

```

```

39         int dni = Integer.parseInt(codigo, 10);
40         return codigo+ NIF_TABLA.charAt(dni % NIF_TABLA.length());
41     }
42     else return null;
43 }
44
45
46     public String [] corregirDatos(String codigo) {
47         throw new UnsupportedOperationException("Not supported yet.");
48     }
49 }

```

Clase UPC Java.

```

1  /*
2   * To change this template, choose Tools / Templates
3   * and open the template in the editor.
4   */
5
6  package practica0;
7
8  /**
9   *
10   * @author
11   */
12  public class UPC extends Codificacion {
13
14      private static final int MULTIPLIO = 10;
15
16      /**
17       *
18       * @param codigo
19       * @return
20       */
21      public boolean verificar(String codigo) {
22          int resultado = 0;
23          int v;
24          try {
25              for (int i = 0; i < codigo.length(); i++) {
26                  v = Integer.parseInt(codigo.substring(i, i+1));
27                  resultado += i%2 == 0 ? v*3 : v;
28              }
29          } catch (NumberFormatException e) {
30              return false;

```

```

31     }
32     return resultado % MULTIPLO == 0;
33 }
34
35 /*****
36  *
37  * @param codigo
38  * @return
39  */
40
41 public String corregirControl(String codigo) {
42     int i = 0;
43     boolean ok = false;
44     String codigoCorregido = new String();
45
46     while (i < codigo.length() && !ok) {
47         int digito = 0;
48         while (digito < 9 && !ok) {
49             codigoCorregido = codigo.substring(0, i-1) + digito +
50                                     codigo.substring(i+1, codigo.length());
51             if (verificar(codigoCorregido)) {
52                 ok = true;
53             } else {
54                 digito++;
55             }
56         }
57     }
58     return codigoCorregido;
59 }
60
61 /*****
62  *
63  * @param codigo
64  * @return
65  */
66 public String generarCodigoControl(String codigo) {
67     throw new UnsupportedOperationException("Not supported yet.");
68 }
69
70
71 public String[] corregirDatos(String codigo) {
72     throw new UnsupportedOperationException("Not supported yet.");
73 }
74
75 }

```

Clase Main Java.

```
1  /*
2   * To change this template, choose Tools / Templates
3   * and open the template in the editor.
4  */
5
6  package practica0;
7
8  import practica0.Codificacion;
9  import practica0.ISBN;
10 import practica0.NIF;
11 import practica0.UPC;
12
13 /**
14  *
15  * @author Administrador
16 */
17 public class Main {
18
19     /**
20      * @param args the command line arguments
21     */
22     public static void main(String[] args) {
23
24
25         String codigoISBN = "0-444-85193-3"; //"84-206-8186-5";
26         String codigoUPC = "9780444485193"; //"039400014416";
27         String codigoDNI = "55555555";
28         String codigoISBN13 = "978-84-473-5602-"; //;
29
30         *****
31         *
32         */
33         Codificacion cod = new UPC();
34         System.out.println(cod.verificar(codigoUPC) ? "válido" : "No válido");
35
36         *****
37         *
38         */
39         cod = new ISBN();
40         System.out.println(cod.generarCodigoControl(codigoISBN));
41         System.out.println(cod.verificar(codigoISBN) ? "válido" : "No válido");
42
43         *****
```

```

44      *
45      */
46      cod = new NIF();
47      System.out.println(cod.generarCodigoControl(codigoDNI));
48      System.out.println(cod.verificar(codigoDNI+"K") ? "válido" : "No válido");
49
50      /*****
51      *
52      */
53      cod = new ISBN13();
54      System.out.println(cod.generarCodigoControl(codigoISBN13));
55      System.out.println(cod.verificar(codigoISBN13+"7") ? "válido" : "No válido");
56  }
57
58 }

```
