

# Codificador-Decodificador ISBN-EAN13 (GTK2.0,GdkPixbuf,GDK & GCC)

*Jose Alberto Benítez Andrades*



Teoría de la Información y Códigos

2ºIngeniería Informática, DNI:71454586A

[Infjab02@estudiantes.unileon.es](mailto:Infjab02@estudiantes.unileon.es)

## **Índice**

**INTRODUCCIÓN**

**1.DESCRIPCIÓN DEL MÉTODO: CODIFICACIÓN-DECODIFICACIÓN**

**2.ALGORITMOS DE RESOLUCIÓN**

**3.PROGRAMACIÓN**

**4.FUNCIONAMIENTO DEL PROGRAMA**

## **INTRODUCCIÓN**

Los códigos de barras son utilizados de diferentes maneras a lo largo de la cadena comercial. El procesamiento de la información obtenida mediante el código de barras posibilita a la empresa la toma de decisiones además de proveer los siguientes beneficios:

- **El fabricante mejora la comunicación con los transportistas y los comercios minoristas**, mejora el control de inventarios, almacén y distribución, obtiene más y mejor información de la demanda del mercado lo que le permite acortar los ciclos de pedido y entrega, simplificar el proceso de información y reducir los costos de administración.
- **El mayorista agiliza el proceso de pedido, recibo y despacho de las mercancías**, debido al control permanente de inventarios, beneficiándose además con la reducción de los costos de administración.
- **El minorista utiliza los códigos de barras para simplificar y agilizar los cambios de precios, ahorrar gastos de etiquetado, obtener información por producto sin errores tanto en el almacén como en el punto de ventas**. Los datos obtenidos pueden ser utilizados para establecer patrones de ventas, tomar decisiones de *merchandising* y acortar los tiempos de resurtido.
- **El consumidor obtiene un rápido servicio en su paso por el punto de venta, desaparecen los errores de precio y tecleo y obtiene un ticket pormenorizado de los artículos adquiridos**.

Uno de los sistemas de codificación más utilizados en la actualidad, y que tratamos en este trabajo, es el EAN13.

## 1.DESCRIPCIÓN DEL MÉTODO: CODIFICACIÓN-DECODIFICACIÓN

El ISBN posee 13 dígitos, el primero es el dígito que determina la paridad, y los otros 12, se dividen en parte izquierda ( los primeros 6 dígitos ) y parte derecha ( el resto). Para codificar nuestro ISBN a EAN13 debemos seguir una serie de pasos:

- Primero analizamos el bit que determina la paridad de los bits de la parte izquierda. Esto se consigue mirando en la siguiente tabla:

PRIMER DIGITO	CON QUE PARIDAD CODIFICAR					
	SEGUNDO DIGITO	CODIGO FABRICANTE				
		1	2	3	4	5
0 (UPC-A)	impar	impar	impar	impar	impar	impar
1	impar	impar	par	impar	par	par
2	impar	impar	par	par	impar	par
3	impar	impar	par	par	par	impar
4	impar	par	impar	impar	par	par
5	impar	par	par	impar	impar	par
6	impar	par	par	par	impar	impar
7	impar	par	impar	par	impar	par
8	impar	par	impar	par	par	impar
9	impar	par	par	impar	par	impar

Para facilitar la escritura, a los pares se les denota con una "B" y a los impares con una "A", así, dependiendo del primer dígito, podemos tener combinaciones del tipo:"AAAAA","ABABB","ABBAB","ABBBA","BAABB","BBAAB","BBBAA","BABAB","BABBA" o "BBABA".

- Después de analizar la codificación que sigue la parte izquierda, simplemente debemos sacar la palabra completa siguiendo la siguiente tabla:

DÍGITO	CODIFICACION IZQUIERDA		CODIFICACION DERECHA
	PARIDAD IMPAR (A)	PARIDAD PAR (B)	ALL CHARACTERS
0	0001101	0100111	1110010
1	0011001	0110011	1100110
2	0010011	0011011	1101100
3	0111101	0100001	1000010
4	0100011	0011101	1011100
5	0110001	0111001	1001110
6	0101111	0000101	1010000
7	0111011	0010001	1000100
8	0110111	0001001	1001000
9	0001011	0010111	1110100

Al codificar un número ISBN a EAN13, tendremos 12 números \* 7 dígitos binarios cada uno = 84 dígitos, secuencia de ceros y unos que debemos convertir en código de barras. El código de barras se crearía poniendo un 101 al inicio del código, otro 101 al final y un 01010 justo en el centro, es decir quedaría estructurado de la siguiente manera:

*101 parteizquierda 01010 partederecha 101 .*

Un ejemplo de **codificación**, supongamos que tenemos el número **6422421303647**, para codificarlo debemos hacer lo siguiente:

- El 6 en la primera tabla tiene la siguiente secuencia

6	impar	par	par	par	impar	impar
---	-------	-----	-----	-----	-------	-------

- Con lo que tendríamos que la parte izquierda es 422421 y en la segunda tabla deberíamos buscar:

4: impar	0100011
2: par	0011011
2: par	0011011
4: par	0011101
2: impar	0010011
1: impar	0001101

- Una vez codificada la parte izquierda, la derecha es muy simple, sólo debemos buscar los valores en la segunda tabla "Codificación Derecha" y tenemos que para 303647 los códigos binarios serían:

3: 1000010
0: 1110010
3: 1000010
6: 1001000
4: 1011100
7: 1110100

- Con lo que el EAN13 final quedaría de la siguiente forma:

101|0100011|0011011|0011011|0011101|0010011|0001101|01010|  
 Inic. | 4 | 2 | 2 | 4 | 2 | 1 | centro |  
 1000010|1110010|1000010|1001000| 1011100|1110100|101  
 3 | 0 | 3 | 6 | 4 | 7 | FIN

Para **decodificar** hay que tener en cuenta si el EAN13 que nos están dando, está del derecho (se lee de izquierda a derecha) o del revés (se lee de derecha a izquierda). La comprobación de si está del derecho o del revés es muy sencilla. Cuando el usuario introduce el EAN13, comprobamos la primera secuencia de 7 dígitos binarios buscándolo en la tabla de "Codificación Izquierda". Si está en la columna de "Pares" significa que el código está del revés y debemos leerlo de derecha a izquierda, en caso contrario el número estaría bien introducido y se leería de izquierda a derecha.

Por ejemplo, el número introducido anteriormente que nos dio el siguiente código EAN13:

```

101|0100011|0011011|0011011|0011101|0010011|0001101|01010|
INI.| 4 | 2 | 2 | 4 | 2 | 1 |CENTRO|
1000010|1110010|1000010|1001000| 1011100|1110100|101
 3 | 0 | 3 | 6 | 4 | 7 | FIN

```

Sería igual que leer de izquierda a derecha el siguiente número:

```

101|0010111|0011101|0001001|0100001|0100111|0100001|01010|
FIN.| 7 | 5 | 6 | 3 | 0 | 3 |centro|
1011000|1100100|1011100|1101100| 1101100|1100010|101
 1 | 2 | 4 | 2 | 2 | 4 | INI

```

Además para comprobar si el código de barras está bien hecho, deberemos comprobar el dígito de control (que está en la posición 13 del ISBN). El método de verificación es muy sencillo:

- 1) Por ejemplo tenemos el siguiente código de barras:

642242130364X

El peso de cada dígito es 1-3-1-3-1-3-1-3-1-3-1-3-1. Para calcular el dígito de verificación, se sumarían los 12 primeros dígitos multiplicados por su peso de la siguiente forma:

$6*1 + 4*3 + 2*1 + 2*3 + 4*1 + 2*3 + 1*1 + 3*3 + 0*1 + 3*3 + 6*1 + 4*3 = 73$ , como el código debe ser múltiplo de 10, la cifra que debemos sumar a 73, para que sea múltiplo de 10, es 7, con lo cual el código de verificación es el 7.

## 2.ALGORITMOS DE RESOLUCIÓN

Los algoritmos que necesitamos para la codificación-decodificación de un código ISBN-EAN13 serían los siguientes:

Las variables y constantes comunes a los 2 algoritmos son:

- ISBN[13]: Array de enteros que almacena el ISBN.
- EAN13: Cadena de caracteres que almacena el EAN13.

- Codificar:

**Función** codificar(ISBN):EAN13;

{ Función que recibe el código ISBN introducido por el usuario y devuelve el código EAN13 }

{ CompruebaDC es una función que verifica el código de control y comprueba si el ISBN es correcto.}

**Si** CompruebaDC(ISBN) *entonces*

Mirando ISBN[0], comprobar secuencia a seguir (AAAAA,ABBAA....);

**Desde** i = 1 hasta 6

**Si** es impar *entonces* EAN13 = EAN13 + binarioIMPAR(ISBN[i])

**sino** EAN13 = EAN13 + binarioPAR(ISBN[i]);

**Desde** i = 7 hasta 12 EAN13 = EAN13 + binarioDERECHA(ISBN[i]);

**sino** salir.

- Decodificar:

**Función** decodificar(EAN13):ISBN;

{ Función que recibe el código EAN13 introducido por el usuario y devuelve el código ISBN }

CadAux : cadena;

j : entero;

{ Almacenamos el primer dígito para comprobar la paridad, si es impar, leeremos de izquierda a derecha, en caso contrario, de derecha a izquierda. }

CadAux = EAN13[0] hasta EAN13[6];

{ Si cadAux es verdadero, se lee de izquierda a derecha, sino, de derecha izquierda.

Para saber si es impar o par, busca en la tabla de impares, si está, es verdadero, sino, es falso.}



**Si** EsImpar(CadAux) **entonces**

{ Desde que comienza la cadena hasta que termina, almacena en la cadena auxiliar las secuencias de 7 dígitos binarios }

**Desde** i = 0 **hasta** finEan13

**Si** (i mod 7 = 0) **entonces**

{ Cuando el resto de la i entre 7 es 0, significa que ya tenemos una secuencia de 7 dígitos en CadAux que ya podemos comprobar. Si i es menor que 43, nos encontramos en la parte izquierda. }

**Si** (i < 43) **entonces**

{ Si es impar, busca en la tabla de impares, sino en la de pares, y almacena el número decimal en el array de enteros ISBN, posición j. }

**Si** EsImpar(CadAux) **entonces** ISBN[j] =

decImpar(CadAux);

**sino** ISBN[j] = decPar(cadAux);

{ Sino, busca en la tabla de los números de la DERECHA y almacena el número decimal en el array de enteros ISBN, posición j. }

**sino** ISBN[j] = decDcha(CadAux);

j = j + 1;

{ Al finalizar el if, inicializamos la cadena a vacía para poder introducir más números. }

CadAux = "";

CadAux = CadAux + EAN[i];

i = i + 1;

**sino**

**Desde** i = finEan13 **hasta** 0

**Si** (i mod 7 = 0) **entonces**

**Si** (i > 42) **entonces**

**Si** EsImpar(CadAux) **entonces** ISBN[j] =

decImpar(CadAux);

**sino** ISBN[j] = decPar(cadAux);

**sino** ISBN[j] = decDcha(CadAux);

j = j + 1;

CadAux = "";

CadAux = CadAux + EAN[i];

i = i - 1;

CompruebaDC sería así:

**Función** CompruebaDC(ISBN):boolean;

{ Recibe el ISBN, hace la suma de los 13 dígitos multiplicados por sus pesos, si la suma total entre 10 da de resto 0, es correcto, sino, no es correcto. }

**Desde** i = 0 **hasta** 13

**Si** (i mod 2 = 0) **entonces** suma = suma + ISBN[i]

**sino** suma = suma + ISBN[i]\*3;

**Si** (suma mod 10 = 0) **entonces** CompruebaDC = VERDADERO;

**sino** CompruebaDC = FALSO;

### 3.PROGRAMACIÓN

El programa consta de una función principal *main()* y 4 funciones *void* que no devuelven ningún tipo de dato.

**Int** VerificaISBN();

- Función que comprueba si el código ISBN es correcto o no. Si la suma de sus dígitos multiplicada por sus pesos, dividido entre 10, da resto 0, el código es correcto(valor 1), de lo contrario, el código es incorrecto (valor 0);

**void** DibujarArea(GtkWidget \*widget);

- Esta función se encarga de dibujar el rectángulo principal donde dibujaremos más adelante el código de barras.

**void** DibujarCodBarras(GtkWidget \*widget);

- Mediante esta función, después de haber calculado el código EAN13, dibujamos el código de barras de una manera muy sencilla. En la variable global *str2* he guardado el código EAN13 completo sin ningún tipo de separación, así que, mediante un *gdk\_draw\_line* dibujo las primeras barras (101) , las centrales (01010) y las finales (101), y una vez dibujadas las primeras, mediante un bucle *for* dibujo la parte izquierda, de forma que, cada vez que encuentra un '1' en la cadena, pinta líneas de 2 px de ancho, y si no, no pinta nada, de la misma forma se hace con la parte derecha, y así queda el código de barras totalmente pintado.

```
void calculaEAN13(GtkWidget *wid);
```

- Para calcular el EAN13 a partir del ISBN, es muy sencillo, mediante las tablas de paridad que he almacenado en 3 arrays, y teniendo en cuenta la paridad, almaceno en una cadena todo el código EAN13.

```
void calculaISBN(GtkWidget *wid);
```

- Es tan sencillo como el anterior, con la salvedad de que ahora, leemos *ceros y unos* y devolvemos el ISBN (módulo10). Lo primero que hacemos es comprobar el primer dígito para ver si el código de barras introducido está del revés, o en posición normal, y dependiendo de esto, entramos en el primer bucle que lee de izquierda a derecha, o en el segundo bucle, que lee de derecha a izquierda.

- Y así almacenamos el ISBN en un array de números, que más tarde uniremos en una cadena para poder mostrar en pantalla. Además, guardamos el código de barras en su forma de lectura "izquierda-derecha" para poder pintar el código de barras correctamente en pantalla.

```
int main(int argc, char *argv[]);
```

- En la función principal simplemente se crean los widgets que vamos a utilizar, la pantalla principal, botones, y se le asigna a cada botón su función.

## 4.FUNCIONAMIENTO DEL PROGRAMA

El funcionamiento es muy sencillo, contamos con una interfaz gráfica en la que se diferencian 2 partes importantes, en la primera parte está el código de barras, que en principio está en blanco, y en la 2ª parte están las 2 entradas de texto que nos pedirán el *ISBN* y el *EAN13* dependiendo de lo que queramos calcular. Al pulsar *Calcular ISBN* calcula el *ISBN* a partir del *EAN13* y viceversa.



Si por algún motivo, introducimos un número ISBN que no es correcto, mostrará una ventana de error el programa.

