NÚMEROS
Revista de Didáctica de las Matemáticas
http://www.sinewton.org/numeros
ISSN: 1887-1984
Volumen 71, agosto de 2009, páginas 87–100

Matemáticas para detectar y corregir errores

Mikel Lezaun Iturralde (Universidad del País Vasco)

Fecha de recepción: 8 de junio de 2009
Artículo solicitado al autor por la revista

Resumen

Hoy en día la información almacenada en un ordenador, en una cámara fotográfica o en un CD, y la contenida en las ondas emitidas por televisión o por un teléfono móvil está traducida a números, está codificada, es digital. En el proceso de codificación o en la transmisión de un mensaje pueden ocurrir errores que habrá que detectarlos, y mejor corregirlos. ¿Cómo hacerlo? Con matemáticas, los números son matemáticas.
En este artículo se estudia el papel de la letra en la detección de errores del Número de Identificación Fiscal (NIF), de los dígitos de control de una cuenta bancaria, y un código de barras. También, se introducen los códigos Hamming para detectar y corregir errores sencillos en un mensaje escrito en números 0 y 1.

Palabras clave Dígitos de control, distancia Hamming, códigos correctores

Abstract

Today, the information stored on a computer, a photographic camera or a CD, and the contained in the waves emitted by television or mobile phone is translated to numbers, it is encoded, it is digital. In the encoding process or in the transmission of a message errors may occur that must be detected, and better corrected. How? With mathematics, the numbers are mathematics.
This article studies the role of letter in error detection of the *Número de Identificación Fiscal* (NIF), of check digits of a bank account, and a bar code. It also introduces Hamming codes to detect and correct the simple errors of a message written in numbers 0 and 1.

Keywords Check digits, Hamming distance, error-correcting codes

1. Introducción

En los últimos años, el término digital ha entrado en el lenguaje cotidiano. Empezaron los relojes digitales, luego se extendieron las cámaras fotográficas digitales, las cámaras de video digitales, la televisión digital, y ahora incluso los periódicos digitales. Cuando damos el número completo de la cuenta bancaria, nos piden incluir los dígitos de control. También observamos que el código de barras de un producto contiene toda la información del mismo, precio incluido, que debajo del código hay números, y que cuando falla el lector se escriben los números. Las personas tenemos un número que nos identifica.

Digital se suele contraponer a analógico, y lo digital se asocia a lo numérico, a los números. Hoy en día es un hecho que cualquiera que sea el soporte en que esté almacenada y el medio por el que se transmita, la información, ya sea un texto, una imagen o unos sonidos, está traducida a

números, es digital. En el proceso de traducción a números (codificación) o durante la emisión, transmisión y recepción del mensaje numérico, se pueden cometer errores, y los errores hay que detectarlos, y mejor corregirlos.

Con los números se pueden hacer cuentas, con los números siempre hay matemáticas, y las matemáticas están para utilizarlas. ¿Cómo utilizar las matemáticas para detectar los errores? La idea básica es muy sencilla. A partir de los números del mensaje se calculan otros que se añaden al mismo. La forma de obtener esos números es conocida, se sabe cuales son y cómo han sido calculados. Al recibir todos los números, tanto los del mensaje y como los redundantes, el lector vuelve a calcular los números redundantes que debería haber, y si coinciden con los recibidos, todo ha podido ir bien, y si no, se ha producido un error.

Una vez detectado un error, la primera opción es decir: ¡error! y solicitar que se vuelva a emitir el mensaje. Pero mejor sería corregirlo. ¿Cómo hacerlo? La idea básica también es sencilla. Si al añadir los números redundantes al mensaje numérico, todos los resultados posibles son muy diferentes unos de otros, y si al rehacer el lector los cálculos obtiene un resultado imposible pero cercano a uno de los posibles, el mensaje erróneo se sustituye por el factible más cercano.

Esas son las ideas básicas y a las matemáticas les toca modelizarlas, definir qué operaciones hay que hacer, diseñar los algoritmos de resolución y resolverlos. La informática, la computación, se encargará de hacer todo el proceso operativo.

Este artículo está dividido en cuatro secciones. En la primera se estudia el papel detector de errores de la letra del DNI, cómo se obtiene esa letra y el porqué del método elegido para calcularla. La segunda sección esta dedicada a los dígitos de control de errores de las cuentas bancarias, y la tercera al código de barras. La cuarta sección trata de los códigos correctores de mensajes binarios, y presenta un código Hamming para detectar y corregir los errores más sencillos de un mensaje escrito en números 0 y 1.

2. La letra del NIF

El Documento Nacional de Identidad (DNI) español nació el 2 de marzo de 1944. Este documento está numerado, cada DNI tiene un número (positivo) de ocho cifras: $a = a_7a_6a_5a_4a_3a_2a_1a_0$ donde cada a_i es uno de los diez dígitos. Los casos en que aparecen menos dígitos se deben a que hay ceros por la izquierda que no se han escrito. El 9 de marzo de 1990, al número del DNI se le añadió una letra de control, que denominaremos letra NIF. El número del DNI junto con la letra NIF constituyen el Número de Identificación Fiscal (NIF).

La letra del NIF es una letra de control, que se obtiene a partir del resto de la división entera del número del DNI entre 23, y que no tiene más misión que detectar equivocaciones al escribirlo. Ese resto será un número entero comprendido entre el 0 y el 22. En base a la siguiente tabla, a cada resto se le asigna la letra correspondiente, que será la letra NIF.

resto	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
letra	T	R	W	A	G	M	Y	F	P	D	X	B	N	J	Z	S	Q	V	H	L	C	K	E

Tabla 1. Correspondencia entre el resto 23 del DNI y la letra NIF

Dado que hay 23 restos, sólo son necesarias 23 letras. Las descartadas son: I, O, U y Ñ. Se puede pensar que la elección de la división por 23 está motivada porque se ha considerado que sólo hay 23 letras válidas: la I, O y U se eliminan porque se pueden confundir con 1, 0 y V, y la Ñ por ser específicamente española. Las matemáticas nos van a mostrar que la elección de 23 es fundamental, que es debida a algo más profundo.

Como se ha indicado antes, la letra NIF es de control, sirve para detectar errores. La equivocación más simple es cambiar un dígito. Por tanto, para que la letra NIF cumpla su cometido, lo mínimo que se puede pedir es que dos números DNI que difieran en un solo dígito no puedan tener la misma letra.

2.1. Dos números de DNI que difieren en un solo dígito no pueden tener la misma letra NIF

Sean dos números de DNI $a = a_7a_6a_5a_4a_3a_2a_1a_0$ y $b = b_7b_6b_5b_4b_3b_2b_1b_0$ con a mayor que b. Estos números en base decimal se escriben de la forma

$$a = a_7 10^7 + a_6 10^6 + a_5 10^5 + a_4 10^4 + a_3 10^3 + a_2 10^2 + a_1 10 + a_0 , \qquad (1)$$

$$b = b_7 10^7 + b_6 10^6 + b_5 10^5 + b_4 10^4 + b_3 10^3 + b_2 10^2 + b_1 10 + b_0 . \qquad (2)$$

Si sólo difieren en un dígito, por ejemplo el correspondiente a la potencia k-ésima de 10, restando las expresiones (1) y (2) se obtiene que su diferencia es

$$a - b = (a_k - b_k)10^k \quad \text{con} \quad a_k - b_k \text{ un número entero del 1 al 9.}$$

Si los dos tuvieran la misma letra, los dos tendrían el mismo resto r al dividirlos por 23, es decir se tendrían las descomposiciones

$$a = 23p + r \quad \text{y} \quad b = 23q + r \quad \text{con } p, q \text{ y } r \text{ números enteros no negativos y } r \le 22 .$$

Restando estas expresiones de a y b se obtendría

$$a - b = 23(p - q) .$$

En consecuencia

$$(a_k - b_k)10^k = 23(p - q) .$$

Esto implicaría que 23 es un divisor de $(a_k - b_k)10^k$, con $a_k - b_k$ un número entero del 1 al 9, lo cual es imposible.

¿Ocurriría lo mismo si se hubiera tomado 24 letras, si se hubiera utilizado el resto de la división por 24? Veámoslo.

Supongamos dos números cualesquiera a y b, con a mayor que b, que se diferencian en 3000, por tanto en un solo dígito. Dividiendo cada uno de ellos por 24 se tiene

$a = 24p + r_1$ y $b = 24q + r_2$ con p, q, r_1 y r_2 números enteros no negativos y $r_1, r_2 \leq 23$.

La resta de esos dos números es

$$a - b = 24(p - q) + (r_1 - r_2) = 3000 = 24 \times 125 .$$

En consecuencia

$$24(p - q - 125) = r_2 - r_1 .$$

Como $r_1 - r_2$ está comprendido entre 0 y 23, esta igualdad sólo es posible si $p - q - 125 = 0$, de lo cual se sigue que $r_1 - r_2 = 0$, y los dos restos son iguales.

Así pues, cualesquiera *a* y *b* que se diferencian en 3000 tienen el mismo resto al dividirlos por 24. Por ejemplo, los números 72851023 y 72854023 tienen resto 7. Esto también se cumple si *a* y *b* se diferencian en 30000, 300000, 3000000 o 30000000. En definitiva, si se utilizaran los restos de la división por 24, habría muchísimas parejas de números DNI que se diferencian en un solo dígito y que comparten la misma letra. Es fácil demostrar que esto mismo habría ocurrido si se hubiera dividido por 20 o 25, y que en lo que respecta a lo anterior, el 21, 22, 26 y 27 se comportan como el 23.

Hemos demostrado que con la división por 23, la letra NIF detecta las equivocaciones en un solo dígito. Ahora bien ¿se puede saber cual ha sido la equivocación y corregirla? La respuesta es no, y para ello es fácil encontrar ejemplos. Uno es el siguiente.

Supongamos que queremos copiar el NIF 72851028T. Nos equivocamos sólo en el último dígito y escribimos 72851025T. La letra no se corresponde con esos números y se detecta que hay un error. Pero a la vista de esto ¿cual es el que queríamos escribir? ¿El 72851028T o el 72851005T? A priori no hay forma de saberlo, los dos son correctos y difieren del 72851025 en un sólo dígito. Así que cuando se detecta un error, no queda más remedio que volver a escribir el DNI prestando más cuidado.

Acabamos de ver un ejemplo de dos números de DNI que se diferencian en dos dígitos y que tienen igual letra NIF. Ahora bien, la equivocación más fácil de cometer que sólo involucra a dos dígitos es intercambiar uno por el otro. En estos casos, ¿la letra puede ser la misma? O lo que es lo mismo ¿la letra NIF detecta ese error?

2.2. Dos números de DNI con sólo dos dígitos intercambiados no tienen la misma letra NIF

Sean dos números *a* y *b* ($a > b$) de DNI que sólo se diferencian en que tienen intercambiados dos dígitos. Como ejemplo tomamos $a = a_7a_6a_5a_4a_3a_2a_1a_0$ y $b = a_7a_2a_5a_4a_3a_6a_1a_0$ que tienen intercambiados los dígitos que ocupan los lugares 2 y 6 comenzando por la izquierda. Estos números se pueden escribir

$$a = a_7 10^7 + a_6 10^6 + a_5 10^5 + a_4 10^4 + a_3 10^3 + a_2 10^2 + a_1 10 + a_0 ,$$

$$b = a_7 10^7 + a_2 10^6 + a_5 10^5 + a_4 10^4 + a_3 10^3 + a_6 10^2 + a_1 10 + a_0 .$$

En general, la diferencia de dos números *a* y *b* que tienen las cifras a_k y a_j intercambiadas es

$$a-b=(a_k-a_j)10^k+(a_j-a_k)10^j=(a_k-a_j)10^j(10^{k-j}-1) \quad \text{donde} \quad 0\le j<k\le 7\,.$$

Como se ha hecho otras veces, si tuvieran el mismo resto al dividirlos por 23 se verificaría que

$$a-b=23(p-q) \quad \text{con } p \text{ y } q \text{ enteros no negativos.}$$

En consecuencia se tendría la igualdad

$$(a_k-a_j)10^j(10^{k-j}-1)=23(p-q)\,.$$

Ahora bien, esto es imposible ya que 23 no es divisor de un número de la forma

$$(a_k-a_j)10^j(10^{k-j}-1) \quad \text{con} \quad a_k \text{ y } a_j \text{ enteros variando entre 0 y 9} \quad \text{y} \quad 0\le j<k\le 7\,.$$

De la misma forma se puede demostrar que esto no ocurre si se hace la división por 21, 22, 26 o 27. Por ejemplo, los números 72851028 y 22851078 tienen el mismo resto 16 al dividirlos por 26.

En definitiva, se ha elegido la división por 23 porque es la que detecta los errores más sencillos en la escritura de los números del DNI: la modificación de un dígito y al intercambio de dos. Esto no sería así si se hubiera elegido la división por 20, 21, 22, 24, 25, 26 o 27.

3. Dígitos de control de una cuenta bancaria

El Código Cuenta Cliente (CCC) que asignan las entidades a las cuentas bancarias está formado por un conjunto de 20 dígitos

$$\underbrace{a_8a_7a_6a_5}_{\textit{entidad}}\ \underbrace{a_4a_3a_2a_1}_{\textit{sucursal}}\ \underbrace{c_1c_2}_{\textit{control}}\ \underbrace{n_{10}n_9n_8n_7n_6n_5n_4n_3n_2n_1}_{\textit{número de cuenta}}$$

que responden a los siguientes datos:

CÓDIGO DE LA ENTIDAD: 4 dígitos CÓDIGO DE LA SUCURSAL: 4 dígitos.	Los códigos de la Entidad y de la Oficina Bancaria se utilizan con sus cuatro dígitos. Por tanto, si el número no tiene estos dígitos se completa con ceros a la izquierda.
DÍGITOS DE CONTROL: 2 dígitos	El primer dígito de control, el c_1, sirve para validar los códigos de la Entidad Bancaria y de la Sucursal. El segundo, el c_2, sirve para validar el número de la cuenta.
NUMERO DE LA CUENTA: 10 dígitos	El número de la cuenta bancaria tiene 10 dígitos. Se utiliza siempre con los diez dígitos de manera que si tiene menos dígitos se completa con ceros a la izquierda.

Tabla 2. Descripción de los 20 dígitos de una cuenta bancaria

Como su nombre indica, los dos dígitos de control sirven para detectar errores y se obtienen a partir de los restantes. Para los dos se utilizan los restos de la división entera por once. Comencemos con el dígito c_1. Se empieza calculando

$$\overline{a} = 4a_8 + 8a_7 + 5a_6 + 10a_5 + 9a_4 + 7a_3 + 6a_2 + 3a_1 . \qquad (3)$$

Luego se divide $\overline{a}$ por 11 y se retiene el resto *r*. A 11 se le quita este resto y el resultado es el primer dígito de control, con la salvedad de que si se obtiene 10 se toma 1, y si se obtiene 11 el dígito de control es 0.

Para el dígito c_2 se procede de forma similar. Se comienza calculando

$$\overline{n} = n_{10} + 2n_9 + 4n_8 + 8n_7 + 5n_6 + 10n_5 + 9n_4 + 7n_3 + 6n_2 + 3n_1 . \qquad (4)$$

Este número se divide por 11, se retiene el resto *s*, a 11 se le sustrae este resto y el resultado es el dígito de control, salvo cuando el resultado es 10 o 11 que se toma 1 o 0 respectivamente.

Observemos que tanto el resto 1 como el 10 producen el mismo dígito de control, el 1. Si se hubiera dividido por 10, todos los restos darían origen a un dígito de control distinto. Entonces ¿por qué no se ha dividido por 10? La respuesta se verá más adelante.

3.1. Cambio de un dígito en el CCC

Si en un código de cuenta cliente $a_8a_7a_6a_5\ a_4a_3a_2a_1\ c_1c_2\ n_{10}n_9n_8n_7n_6n_5n_4n_3n_2n_1$ se cambia un dígito de los ocho primeros, el resto *r* de dividir $\overline{a}$ por 11 varía. Lo mismo ocurre con el resto de $\overline{n}$ si se cambia uno de los diez últimos.

En efecto, si cambiamos uno de los ocho primeros dígitos, por ejemplo a_7 por b_7 con $a_7 > b_7$, restando los valores $\overline{a}$ y $\overline{b}$ correspondientes se obtiene $\overline{a} - \overline{b} = 8(a_7 - b_7)$. Si tanto $\overline{a}$ como $\overline{b}$ tuvieran el mismo resto *r* al dividirlos por 11, se tendría

$$\overline{a} = 11p + r \quad \text{y} \quad \overline{b} = 11q + r \quad \text{con } p, q \text{ y } r \text{ enteros no negativos y } r \le 10 .$$

Restando estas expresiones se obtendría $\overline{a} - \overline{b} = 11(p - q)$. En consecuencia se verificaría $8(a_7 - b_7) = 11(p - q)$, lo cual es imposible debido a que $a_7 - b_7$ es un entero entre 0 y 9.

Esto no habría ocurrido si se hubiera dividido $\overline{a}$ y $\overline{b}$ por 10. En efecto, cambiando un dígito de los que tienen coeficiente par en la expresión de (3) o (4) por otro cinco unidades mayor o menor, se obtiene el mismo resto. Por ejemplo, los números $a = 20950001$ y $b = 20950051$ son tales que $\overline{a}$ y $\overline{b}$ tienen el mismo resto 4 al dividirlos por 10. Esta es la razón por la que no se ha elegido dividir por 10.

Sigamos con la división por 11. Con un cambio de un solo dígito se puede obtener los restos 1 y 10, y por tanto el mismo dígito de control. Por ejemplo, las ocho primeras cifras 20950001 y 20950005 dan $c_1 = 1$, y las diez últimas 4032327895 y 4032328895 también producen 1 como segundo dígito de control. Para solventar esta contingencia, los bancos tienen que tener cuidado en no

asignar en dos sucursales distintas un mismo número de cuenta, ni en asignar como número de cuenta uno que al dividirlo por 11 de como resto el 10. De esta forma, el cambio de un dígito de un CCC operativo es incompatible con la conservación de los dígitos de control.

3.2. Intercambio de dos dígitos de una CCC

Para fijar las ideas nos vamos a centrar sólo en los 10 últimos dígitos del Código Cuenta Cliente. Dos CCC que se diferencien en sólo dos dígitos pueden tener el mismo dígito de control. Por ejemplo los números 4037327895 y 4035427895 tienen el mismo control $c_2 = 3$. Pero un intercambio de dos dígitos ¿puede dar origen a un mismo dígito de control? Veamos la respuesta.

Sean *n* y *m* los últimos diez dígitos de dos números CCC tales que uno se obtiene del otro sólo intercambiando dos dígitos. Entonces los números $\overline{n}$ y $\overline{m}$ calculados con la fórmula (4) tienen diferente resto al dividirlos por 11.

En efecto, supongamos que $n_{10}n_9n_8n_7n_6n_5n_4n_3n_2n_1$ y $m_{10}m_9m_8m_7m_6m_5m_4m_3m_2m_1$ se obtienen uno del otro intercambiando dos dígitos, por ejemplo el n_8 y el n_3. Calculando $\overline{n}$ y $\overline{m}$ por la fórmula (4) y restando los valores obtenidos se tiene la relación

$$\overline{n} - \overline{m} = (4-7)(n_8 - n_3).$$

Por otro lado, si $\overline{n}$ y $\overline{m}$ tuvieran el mismo resto *s* al dividirlos por 11 se tendría

$$\overline{n} = 11p + s \quad \text{y} \quad \overline{m} = 11q + s \quad \text{con } p, q \text{ y } s \text{ enteros no negativos y } s \le 10.$$

Restando estos dos números se obtendría

$$\overline{n} - \overline{m} = 11(p-q).$$

En consecuencia se verificaría

$$(4-7)(n_8 - n_3) = 11(p-q) \quad \text{con} \quad n_8 - n_3 \text{ número entero entre 0 y 9,}$$

lo cual es imposible.

Lo anterior no quita que intercambiando dos dígitos se pueda obtener el mismo dígito de control 1. Por ejemplo, el número 4032377886 genera el resto 10 y el 4032737886 el resto 1, por lo que los dos producen el mismo $c_2 = 1$. Ahora bien, una buena gestión de los bancos para no asignar a cuentas números que al dividirlos por 11 den cómo resto el 10 solventa esta anomalía, y los dígitos de control cumplen su función: detectan la modificación de un dígito y el intercambio de dos.

4. Código de barras

Un código de barras es una traducción a un conjunto de líneas paralelas de distinto grosor y espaciado de una determinada información, en general un conjunto de números que se suelen escribir debajo. La primera patente de un código de barras se registró en 1952 en Estados Unidos. Comenzó a

comercializarse en 1996 y su gran éxito le llegó a partir de 1980. Aquí nos centraremos en el mensaje numérico, no en su traducción a barras.

El EAN-13 (European Article Number) es un sistema de códigos de barras adoptado por multitud de empresas. Está constituido por 13 dígitos divididos en cuatro grupos

$$\underbrace{a_{12}a_{11}}_{país}\ \underbrace{a_{10}a_9a_8a_7a_6a_5a_4a_3a_2a_1}_{código\ de\ la\ empresa\ y\ del\ producto}\ \underbrace{c}_{control}$$

1. País: los primeros dígitos identifican a través de qué organización nacional se ha adscrito una empresa al Sistema EAN. En España se encarga de ello Aecoc y su código es el 84.
2. Código de la empresa: es un número compuesto por entre 5 y 8 dígitos, dependiendo de las necesidades de la empresa, el cual identifica al propietario de la marca.
3. Código del producto: completa los 12 primeros dígitos.
4. Dígito de control: consta de un solo dígito y sirve para verificar que el código leído es correcto. El proceso de su cálculo es muy sencillo, basta con seguir tres pasos:

   - Se numeran los doce primeros dígitos comenzando de izquierda a derecha. Los dígitos que ocupan una posición impar se suman, y los que ocupan una posición par se multiplican por 3 y se suman.
   - Se suman los dos números obtenidos.
   - Se busca la decena inmediatamente superior al resultado de la suma anterior y se le resta esa suma. El resultado obtenido es el dígito de control del lugar 13.

Por ejemplo, sea el siguiente código de barras.

Numerando los doce primeros números de izquierda a derecha, la suma de los de lugar par es $8+5+3+1+5+3=25$, y la suma de los de lugares impares multiplicados por tres es $3(4+4+2+6+4+2)=66$. Como $25+66=91$, la decena inmediatamente superior es 100, por lo que el dígito de control es $c=100-91=9$.

En este caso se plantean las mismas cuestiones: dos códigos de barras que sólo se diferencien en un dígito de los 12 primeros ¿pueden tener el mismo dígito de control? Intercambiado dos de los doce primeros dígitos ¿se obtiene el mismo dígito de control? Las respuestas se dejan como ejercicio.

5. Digitalizar para corregir

Actualmente la información, esté contenida en un ordenador, en un disco compacto, en una cámara fotográfica, en las ondas emitidas por una sonda espacial, por televisión o por un teléfono móvil, se encuentra codificada (traducida) a una sucesión de señales binarias, de "bits", que se pueden asimilar matemáticamente a "0" y "1". Hoy en día estamos acostumbrados a que la mayoría de los aparatos sean calificados como digitales.

Una forma de traducir un mensaje escrito a números es utilizar un código ASCII, acrónimo inglés de American Standard Code for Information Interchange, que fue creado en 1963 por iniciativa de un comité de la American Standards Association (ASA). Este código representa los caracteres tipográficos basados en el alfabeto latino mediante conjuntos de ocho bits, denominados octetos. La tabla 3 da la equivalencia numérica de las letras mayúsculas en ASCII.

Binario	Decimal	Representación		Binario	Decimal	Representación
01000001	65	**A**		01001110	78	**N**
01000010	66	**B**		01001111	79	**O**
01000011	67	**C**		01010000	80	**P**
01000100	68	**D**		01010001	81	**Q**
01000101	69	**E**		01010010	82	**R**
01000110	70	**F**		01010011	83	**S**
01000111	71	**G**		01010100	84	**T**
01001000	72	**H**		01010101	85	**U**
01001001	73	**I**		01010110	86	**V**
01001010	74	**J**		01010111	87	**W**
01001011	75	**K**		01011000	88	**X**
01001100	76	**L**		01011001	89	**Y**
01001101	77	**M**		01011010	90	**Z**

Tabla 3. Representación numérica de las letras mayúsculas en código ASCII

Un hecho muy importante es que una mota de polvo en un CD, una tormenta con sus perturbaciones electromagnéticas, o un rayo cósmico que colisiona con un componente electrónico pueden cambiar el sentido de uno o varios bits, a veces de miles de bits. Ahora bien, un 1 que se convierte en 0 es como si un "si" se transformara en un "no", o un "ding" en un "dong", y el sentido del mensaje puede quedar completamente modificado. ¿Cómo solventar estos incidentes, extremadamente frecuentes, que ocurren en el transcurso de la emisión, transmisión o recepción de mensajes (textos, sonidos, imágenes, etc.)?

Una de las grandes ventajas de la digitalización es que permite someter a los mensajes a un tratamiento aritmético, a hacer cuentas con los números. La ideal central de la corrección de errores, conocida desde comienzos de la informática a mediados del siglo XX, consiste en dividir el mensaje inicial en "palabras" de longitud fija, y añadir a cada una de ellas un cierto número de bits redundantes calculados a partir de los bits de la palabra. Se trata pues de alargar las "palabras" para que se puedan reconocer, y en consecuencia corregir, aunque se hayan modificado algunas de sus "letras". Realizar este proceso se denomina codificar.

La capacidad de corregir errores reside en que si todas las palabras codificadas posibles son muy diferentes unas de otras, cuando se recibe una palabra errónea, una palabra imposible, se puede sustituir por la posible más próxima. Obviamente, para ello se deberá poder determinar cual es la palabra válida más cercana a la recibida. Una vez corregidas, de cada palabra codificada se eliminan los bits redundantes, dicho de otra forma, se descodifica para recuperar el mensaje original. Habrá pues que comenzar por definir el concepto de "distancia" entre dos palabras de igual longitud.

Se define la distancia de Hamming entre dos palabras como el número de posiciones que tienen bit diferente. Por ejemplo, la distancia entre 000 y 111 es tres, 000 y 011 tienen distancia dos, y 111 y 101 distancia uno. Las palabras 10100011 y 10110100 tienen distancia cuatro. Para poder corregir

errores de un solo bit en las palabras codificadas, la distancia entre todas ellas tiene que ser como mínimo tres. Así, para cada palabra con un solo bit erróneo habrá una única válida a distancia uno, que será la verdadera palabra a la que corresponde. Si se desea corregir errores de dos bits por palabra, la distancia mínima entre ellas deberá ser como mínimo cinco.

Una forma muy elemental aunque académica de codificar las palabras para poder corregir errores es considerar que cada bit es una palabra, y añadir a cada palabra, a cada bit, otros dos iguales. Así, las únicas palabras código posibles serán 000 y 111. La distancia entre ellas es tres. Si se admite que en el transcurso de una transmisión sólo puede ocurrir un error de un solo bit por palabra, al recibir 011, 101 o 110 se sabe que se había emitido 111, y al recibir 100, 010 o 001 que se había emitido 000. Una vez hechas las correcciones, de las ternas 000 y 111 se eliminan las redundancias y se obtiene el mensaje original correcto.

5.1. Código Hamming (7,4)

El código Hamming (7,4), introducido por matemático estadounidense Richard Hamming en 1950, comienza dividiendo el mensaje en palabras de cuatro bits. Hay 2^4 palabras diferentes: 0000, 0001, 0010, 0100, 1000, 1001, 1010, 1100, 0101, 0110, 0011, 1110, 1101, 1011, 0111, 1111. Luego, a cada palabra $a_1a_2a_3a_4$ del mensaje se le añade tres bits p_1, p_2 y p_3 de redundancia, calculados a partir de a_1, a_2, a_3 y a_4.

Los a_1, a_2, a_3 y a_4 son 0 o 1 y se quiere que los p_1, p_2 y p_3 calculados a partir de ellos también tomen los valores 0 o 1. Es decir, se quiere hacer operaciones con 0 y 1 y que el resultado sea 0 o 1. Para ello se recurre a la teoría de los cuerpos finitos, que nació con el matemático francés Évariste Galois (1811-1832) al estudiar la resolución de las ecuaciones algebraicas. Un cuerpo finito es un conjunto finito de elementos que, como los números reales o complejos, se pueden sumar, restar, multiplicar y dividir, sin que el resultado salga de esos números. En el cuerpo finito de dos elementos $\{0, 1\}$ las operaciones suma y producto son las de las tablas 4 y 5.

+	**0**	**1**
0	0	0
1	1	0

Tabla 4. Suma en un cuerpo de dos elementos

×	**0**	**1**
0	0	0
1	0	1

Tabla 5. Producto en un cuerpo de dos elementos

Las tablas 6 y 7 son las de la adición y multiplicación que hacen del conjunto de cuatro elementos $\{0, 1, x, y\}$ un cuerpo.

+	**0**	**1**	***x***	***y***
0	0	1	x	y
1	1	0	x	y
x	x	y	0	1
y	Y	x	1	0

Tabla 6. Suma en un cuerpo de cuatro elementos

×	**0**	**1**	***x***	***y***
0	0	0	0	0
1	0	1	x	y
x	0	x	y	1
y	0	y	1	x

Tabla 7. Producto en un cuerpo de cuatro elementos

En el código Hamming (7,4), los bits p_1, p_2 y p_3 redundantes se definen por

$$p_1 = a_1 + a_2 + a_4$$
$$p_2 = a_1 + a_3 + a_4$$
$$p_3 = a_2 + a_3 + a_4$$

donde la suma es en el cuerpo finito de dos elementos.

Otra forma equivalente de definir los p_1, p_2 y p_3, que es la que utilizaremos, es

$$p_1 = paridad\ de\ \ a_1 + a_2 + a_4 \tag{5}$$
$$p_2 = paridad\ de\ \ a_1 + a_3 + a_4 \tag{6}$$
$$p_3 = paridad\ de\ \ a_2 + a_3 + a_4 \tag{7}$$

donde la paridad de un número *p* se define como 1 si el número es impar, y 0 si es par.

Así, cada palabra $a_1a_2a_3a_4$ genera la palabra código $p_1p_2a_1p_3a_2a_3a_4$, con los siete dígitos escritos en el orden que se indica. Por tanto, una vez codificadas, de los siete bits de cada nueva palabra sólo cuatro son del verdadero mensaje. De aquí el utilizar la especificación (7,4) en el nombre.

Es fácil comprobar que la distancia mínima entre todas las palabras codificadas es tres, por lo que este código es capaz de corregir un error de un solo bit en cada palabra codificada de siete bits.

Para describir con detalle el funcionamiento del código Hamming (7,4), vamos a ir explicando los procesos a través de un ejemplo sencillo. Supongamos que el mensaje a transmitir es TEIDE, así en mayúsculas. Este mensaje en código ASCII binario es la siguiente secuencia de 40 bits:

$$\underbrace{01010100}_{\text{T}}\underbrace{01000101}_{\text{E}}\underbrace{01001001}_{\text{I}}\underbrace{01000100}_{\text{D}}\underbrace{01000101}_{\text{E}} \tag{8}$$

Las palabras del mensaje (8) son: 0101, 0100, 0100, 0101, 0100, 1001, 0100, 0100, 0100 y 0101. Añadiendo a cada una de ellas los bits de redundancia obtenidos mediante las fórmulas (5)-(7) se obtienen las palabras código: 0100101, 1001100, 1001100, 0100101, 1001100, 0011001, 1001100, 1001100, 1001100, 0100101. La secuencia de estas palabras código constituye el mensaje a emitir. En este caso es el mensaje de 70 bits

0100101100110010011000100101100110000110011001100100110010011000100101

Supongamos que se recibe el mensaje con algunos errores. Por ejemplo que se recibe la secuencia

0100101100100010011000100101*1*011000011001100110010011001001101*1*0100101

en la que se han marcado los errores.

El receptor de este mensaje numérico debe detectar y corregir automáticamente los errores cometidos. Para ello se vuelve a dividir el mensaje en palabras de longitud siete. Se tiene así

$$\underbrace{0100101}_{n_1}\underbrace{1001\bar{0}001}_{n_2}\underbrace{0011000}_{n_3}\underbrace{1001011}_{n_4}\underbrace{\bar{1}0110000}_{n_5}\underbrace{1100111}_{n_6}\underbrace{0011001}_{n_7}\underbrace{0011001}_{n_8}\underbrace{0011001}_{n_9}\underbrace{\bar{1}0100101}_{n_{10}}$$

donde se siguen señalado los tres errores.

Para cada una de estas palabras $q_1q_2b_1q_3b_2b_3b_4$, el receptor calcula los propios dígitos de redundancia s_1, s_2 y s_3 con las mismas fórmulas que antes

$$\begin{aligned} s_1 &= \textit{paridad de } \ b_1+b_2+b_4 \\ s_2 &= \textit{paridad de } \ b_1+b_3+b_4 \\ s_3 &= \textit{paridad de } \ b_2+b_3+b_4 \end{aligned}$$

Estos dígitos los compara con los q_1, q_2 y q_3 que ha recibido y define los tres números r_1, r_2 y r_3 tales que

$$r_j = 0 \quad \text{si} \quad s_j = q_j \qquad \text{y} \qquad r_j = 1 \quad \text{si} \quad r_j \neq q_j, \qquad j = 1, 2, 3\,.$$

Es inmediato calcular la tabla 8 que relaciona los errores en un dígito con los valores de r_1, r_2 y r_3. Pues bien, como se observa en esa tabla, lo ingenioso de este método es que la forma de calcular p_1, p_2 y p_3 y su colocación en los lugares 1, 2 y 4 de las palabras código, hacen que si se expresa el número binario $r = r_3r_2r_1$ en forma decimal, el número que se obtiene es el lugar en que se ha cometido un error, lo cual permite corregirlo.

	r_3	r_2	r_1	**Decimal**
Error en p_1	0	0	1	1
Error en p_2	0	1	0	2
Error en a_1	0	1	1	3
Error en p_3	1	0	0	4
Error en a_2	1	0	1	5
Error en a_3	1	1	0	6
Error en a_4	1	1	1	7

Tabla 8. Detección de los errores en el código Hamming (7,4)

En el ejemplo, los errores se han cometido en la segunda, quinta y novena palabra. Veamos una a una.

- Para la palabra 1001000 los tres dígitos de redundancia deberían ser 000. Comparándolos con los 101 recibidos se tiene $r = 101$, que en base decimal es el 5. Ha habido un error en el quinto dígito.
- Para la palabra 1101100 los tres dígitos de redundancia deberían ser 101. Comparándolos con los 111 recibidos se tiene $r = 010$, que en base decimal es el 2. Ha habido un error en el segundo dígito.
- Para la palabra 1001101 los tres dígitos de redundancia deberían ser 010. Comparándolos con los 101 recibidos se tiene $r = 111$, que en base decimal es el 7. Ha habido un error en el séptimo.
- Para las otras palabras, los dígitos de redundancia que debería haber y los recibidos son los mismos, luego $r = 000$ y no ha habido error.

Corrigiendo los errores se obtienen las palabras código iniciales 0100101, 1001100, 100100, 0100101, 1001100, 0011001, 1001100, 1001100, 1001100, 0100101.

Para terminar, de cada una de estas palabras de siete dígitos se eliminan los dígitos de redundancia de los lugares 1, 2 y 4, y se obtiene el mensaje inicial. Pasando los dígitos a letra se vuelve a obtener TEIDE.

En realidad hay toda una familia de códigos Hamming que se construyen de forma similar. En todos ellos los dígitos de redundancia están en los lugares potencia de dos. Por ejemplo, en el código Hamming (15,11), a cada palabra $a_1a_2a_3a_4a_5a_6a_7a_8a_9a_{10}a_{11}$ de once bits se le añaden cuatro bits p_1, p_2, p_3 y p_4 calculados por las fórmulas

$$p_1 = \textit{paridad de } \ a_1 + a_2 + a_4 + a_5 + a_7 + a_9 + a_{11}$$
$$p_2 = \textit{paridad de } \ a_1 + a_3 + a_4 + a_6 + a_7 + a_{10} + a_{11}$$
$$p_3 = \textit{paridad de } \ a_2 + a_3 + a_4 + a_8 + a_9 + a_{10} + a_{11}$$
$$p_4 = \textit{paridad de } \ a_5 + a_6 + a_7 + a_8 + a_9 + a_{10} + a_{11}$$

Se obtiene así la palabra código $p_1p_2a_1p_3a_2a_3a_4p_4a_5a_6a_7a_8a_9a_{10}a_{11}$ de quince bits, de los cuales once son de información.

En los códigos correctores, un parámetro a tener en cuenta es la tasa de transmisión, que es el cociente entre la longitud de las palabras antes de codificar y las palabras codificadas, o lo que es lo mismo, la proporción del número de bits de mensaje entre el total de bits. La tasa de transmisión es un número menor que uno y mide la velocidad y coste de la transmisión. Cuanto mayor sea ese número mejor. En el código que triplica los bits la tasa de transmisión es $1/3$, en el Hamming (7,4) es $4/7$, y en el Hamming (15,11) es $11/15$.

Los códigos Hamming, lo mismo que el que triplica cada bit, corrigen errores de un solo bit por palabra codificada. Son pues códigos más bien mediocres. Existen muchos otros códigos con mejores prestaciones, los cuales utilizan conceptos sofisticados de teoría de números y, en particular, la aritmética de cuerpos finitos. Se trata, por ejemplo, de los códigos de Reed-Solomon, inventados por los matemáticos Irving S. Reed y Gustave Solomon en el MIT el año 1960, que tienen numerosas utilizaciones. Citaremos entre otras los discos CD y la telemetría de los satélites civiles, donde intervienen códigos de Reed-Solomon basados en los cuerpos finitos de 256 elementos.

Bibliografía

El contenido de este artículo es estándar y se puede encontrar en muchos textos y páginas web. Para localizarlos nada mejor que buscar uno mismo en Google. Una referencia muy recomendable es

Roman, S. (1992). *Coding and Information Theory*. Springer-Verlag, New York.

Mikel Lezaun Iturralde es Catedrático de Matemática Aplicada del Departamento de Matemática Aplicada, Estadística e Investigación Operativa de la Universidad del País Vasco – Euskal Herriko Unibertsitatea. M. Lezaun nació en el Valle de Baztán (Navarra) el 11 de agosto de 1953, es Licenciado en Matemáticas por la Universidad de Zaragoza, Doctor por la Universidad del País Vasco y ha publicado artículos de investigación en revistas de prestigio. Su trabajo *Predicciones del Tiempo y Matemáticas* fue galardonado con el III Premio Sema de Divulgación en Matemática Aplicada. Actualmente dirige el Grupo de Transferencia de Tecnología Matemática, con contratos para empresas como Metro Bilbao, EuskoTren, FEVE, Cespa, Aguas de Barcelona, Sidenor y Eroski.