Algoritmos

Vicente Trigo Aranda
www.vicentetrigo.com

Si la cultura, en general, no está sobrevalorada en nuestro país, la ciencia y la tecnología andan mucho peor. Gente que se escandalizaría al escuchar que "El Quijote" lo escribió Calderón, sustituye el término "algoritmo" por "logaritmo" sin el menor pudor. De hecho, he leído varios artículos científicos en la prensa diaria, por lo general traducciones del inglés, donde *algorithm* se ha transformado en logaritmo, convirtiendo la frase en un galimatías sin sentido.

Pero esta confusión no sólo ocurre al traducir. Por ejemplo, acabo de encontrar en una Web que "*lo importante no es el logaritmo, sino la idea*", algo que no tiene el menor sentido, ya que el autor (escritor con varias publicaciones, para más señas) intentaba expresar que es más importante saber qué hacer que el cómo hacerlo.

Como además, con el auge de la informática, cada día va siendo más habitual y cotidiano el término algoritmo, no viene mal dedicarle una cierta atención; eso sí, en plan divulgativo, huyendo de profundas cuestiones técnicas. En otras palabras, la operación más complicada que aparecerá en este artículo será una división, y vocablos como tiempo de ejecución, problemas NP, optimización, grafos, etc., los dejo para especialistas en algoritmia.

A lo largo de las siguientes páginas le mostraré algunos algoritmos sencillos, interesantes y entretenidos, aunque, claro está, lo primero será comentar la etimología de esta palabra… y su significado actual.

¿DE DÓNDE PROCEDE LA PALABRA ALGORITMO?

En el siglo IX, el matemático y geógrafo Mohammed Ibn Musa Al-Khwarizmi (figura 1) escribió un pequeño libro donde explicaba los nuevos dígitos creados por los hindúes y cómo utilizarlos para realizar sencillas operaciones. Se ha perdido la versión original de esta obra, "*Kitab al-Jam'a wal-Tafreeq bil Hisab al-Hindi*" (también llamada "Aritmética", en forma abreviada), pero se han conservado diversas traducciones al latín, realizadas todas ellas en España, que entonces era el punto de encuentro de las culturas árabe y latina.

La traducción más antigua que se conoce, parece ser[1] que fue escrita alrededor de 1120, bajo el título "*Liber*

[1] Siempre que se hace referencia a nombres, títulos o fechas de aquella época histórica, estos datos deben tomarse con cierta cautela. Al no existir la imprenta, la grafía variaba mucho de un copista a otro y al haberse perdido la documentación que pudiese existir, que no era mucha, las dataciones necesariamente son aproximadas.

algorismi de numero indorum"[2], por el clérigo inglés Robert de Chester o Robert de Ketton, a quien, además, se debe la primera traducción del Corán al latín.

Figura 1. Sello ruso de 1983, en conmemoración de Al-Khwarizmi

Se conserva otra versión más amplia de la Aritmética de Al-Khwarizmi (la anterior está incompleta y, además, contiene errores), que se atribuye al también inglés Adelardo de Bath (1070-1142) y tenía por título "*Alchorismi in artem astronomicam*". Por su parte, el español Juan de Sevilla[3] siguió el mismo camino con su "*Algorismi de practica arismetrice*", aparecido a mediados del siglo XII.

A comienzos del siglo siguiente, fueron surgiendo otras obras que popularizaron los nuevos descubrimientos matemáticos y tuvieron mayor repercusión. Así, merecen recordarse, entre otros, "*Liber Abaci*" de Fibonacci (1202), "*Algorismus vulgaris*" de Johannes de Sacrobosco (1219) y "*Carmen de algorismo*" de Alexandre de Villedieu (1225).

Bueno, después de leer los títulos de los libros anteriores, está claro de donde procede el término "algoritmo", ¿verdad? En efecto, del nombre latinizado del gran matemático árabe Al-Khwarizmi.

Sin embargo, la Real Academia tiene otra opinión. En su diccionario online (http://www.rae.es/) indica la siguiente procedencia para la palabra algoritmo:

> "Quizá del lat. tardío *algobarismus*, y éste abrev. del ár. clás. *Hisābu lgubār*, cálculo mediante cifras arábigas."

¡Qué puedo decirle! Es una muestra más de que la cultura científica en nuestro país brilla por su ausencia… aunque uno esperaba más rigor en la Real Academia[4].

También es verdad que no sólo en los sillones de la anterior institución se idean etimologías originales. Lea, por ejemplo, la que he encontrado en http://www.epsilones.com/paginas/t-etimologias.html:

> "Al-Khowarizmi hizo una exposición tan completa del método de numeración hindú que se acabó conociendo como el sistema de Al-Khowarizmi, que daría lugar después a los términos *guarismo* para cada uno de los signos con los que se representan los números y *algoritmo* para referirse al sistema completo, este último por influjo del griego *arithmós* "número" y el castellano logaritmo."

Ya le comenté que eso de mezclar algoritmo y logaritmo no es algo inhabitual, por desgracia.

¿QUÉ ES UN ALGORITMO?

La primera acepción de algoritmo que ofrece la Real Academia, en su diccionario online, es bastante aceptable.

> "Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema".

Es decir, un algoritmo es la secuencia de pasos que debemos seguir para resolver un problema. ¿Un ejemplo? Seguramente el más cotidiano es cualquier receta de cocina, donde se nos explica qué operaciones tenemos que realizar para obtener un apetitoso plato.

Sin embargo, hay ocasiones en que nuestra actividad culinaria no obtiene una recompensa sabrosa, ya

[2] El manuscrito permanece en la Biblioteca de la Universidad de Cambridge y fue descubierto en el siglo XIX. Comenzaba así: "*Dixit Algoritmi: laudes deo rectori nostro atque defensori dicamus dignas*".

[3] Fue un judío converso y se le conoce por muy variados nombres: Johannes Hispanensis, Johannes (David) Toletanus, Aven Daud, Avendar, Juan de Luna, etc.

[4] Curiosamente, en la misma Web puede encontrarse el discurso "*Ciencia y Lenguaje*" que leyó D. Antonio Colino López el 23 de enero de 1972, en su toma de posesión como académico. Ahí sí se afirma que "*la palabra algoritmo se deriva del nombre del matemático árabe Al-Khuwarizmi del siglo IX*"

que, dejando de lado la poca habilidad que tenemos algunos con las sartenes y cacerolas, hay recetas tan esquemáticas y poco claras que incluso son ilegibles para delantales expertos.

Las características básicas que debe tener todo buen algoritmo, y, por tanto, cualquier receta de cocina, son las siguientes:

- Precisión: No debe haber la menor ambigüedad, ni en las operaciones a realizar ni en su orden, ni omitir algo. Por ejemplo, en una receta dice: "*En otro cuenco, batir dos huevos con los dos azúcares e incorporar la mantequilla fundida*". Si la siguiésemos al pie de la letra, el pastel saldría con múltiples fragmentos de cáscara de huevo… y es que el lenguaje humano es un tanto impreciso.
- Definibilidad: Cada vez que se repita, en las mismas condiciones, el resultado debe ser idéntico. Sí, ya sé que con las recetas de cocina esto no siempre sucede… pero es que las condiciones no son las mismas en todo momento.
- Finitud: Evidentemente, para que el algoritmo tenga alguna utilidad, debe acabar en un número finito de pasos. De hecho, el tiempo que nos lleva completar el algoritmo es casi siempre algo primordial. ¿Por qué, si no, tuvimos que aprendernos las tablas de multiplicar, si podemos averiguar el valor de 9 x 6 sumando 6 veces 9?

No obstante, existen algoritmos que, aún siendo válidos en teoría, no son efectivos en la práctica[5]. Por ejemplo, hallar los divisores primos de un número probando con los inferiores a él resulta práctico cuando se trabaja con números pequeños, pero no cuando éstos constan de centenares de cifras. Un ordenador actual podría necesitar miles de millones de años en encontrar sus divisores primos (en este hecho se basa el cifrado actual de los modernos sistemas informáticos).

ALGORITMO DE EUCLIDES

Uno de los algoritmos más antiguos que se conocen, y que todavía se utiliza hoy en día, se debe a Euclides y permite calcular el máximo común divisor de dos números de una manera cómoda y rápida[6]. Es decir, la finalidad de este algoritmo radica en encontrar el mayor número entero que es divisor de los dos números dados.

Adaptado al lenguaje actual, el algoritmo de Euclides viene a decir que puede obtenerse el máximo común divisor de dos números, a y b, en la siguiente forma:

1. Calcular r, el resto de la división entre a y b
2. Si $r = 0$ el mcd (a, b) es b. En caso contrario, se hace $a = b$ y $b = r$ y se vuelve al paso 1.

Por ejemplo, para hallar el mcd (1230, 450) basta efectuar las siguientes divisiones (figura 2):

```
1230| 450
 330    2

 450| 330
 120    1

 330| 120
  90    2

 120|  90
  30    1

  90|  30 <— mcd
   0    3
```

Figura 2. Divisiones sucesivas para hallar el máximo común divisor

Consecuentemente, se tiene que el mayor divisor común de 1230 y 450 es 30.

Enseguida se aprecia que este algoritmo es mucho más eficiente que el método de descomposición en factores primos que se enseña en la escuela. Por una parte, es mucho más rápido, salvo que trabajemos con números pequeños; por otro lado, no exige memorizar todos los factores primos, sino únicamente tres números. Debido a esto, el algoritmo de Euclides es el que se implementa en los ordenadores para calcular el máximo común divisor.

JUEGO DEL QUITPAL

Existen muchos sencillos juegos que admiten un algoritmo ganador, de modo que, conociendo la ade-

[5] También hay problemas que no son resolubles mediante algoritmos. Reciben el nombre de problemas indecidibles y lo cierto es que son bastante complejos.

[6] Su primera versión conocida tiene unos 2.300 años, pues Euclides lo incluyó en sus Elementos.

cuada estrategia, podemos vencer siempre. En este artículo le voy a comentar dos de ellos, para que pase un rato entretenido… y disfrute ganando.

Empecemos con el más elemental, el Quitpal. Participan dos personas y se juega con 15 fichas. Alternativamente, cada persona coge 1, 2 o 3 fichas, a su elección, perdiendo quien toma la última ficha.

Haga una pausa en la lectura y practique un poco con el Quitpal. Seguro que encuentra rápidamente el algoritmo ganador.

¡En efecto! Quien tiene el primer turno, gana sin más que coger 2 fichas y, después, cada vez que su rival tome k fichas, coger $4 - k$.

Claro está que el juego es tan sencillo que, a pocas partidas que juegue, su rival descubrirá también el algoritmo ganador. Así que la única forma de seguir practicándolo será complicar algo el Quitpal para que no sea tan fácil descubrir el algoritmo ganador.

Por ejemplo, puede dejarle a su rival que decida cuántas fichas hay inicialmente y cuántas se pueden coger, como máximo, en cada turno. ¿Qué deberá hacer, ahora, para ganar siempre?

1. Sólo tiene que decidir quién comienza. Si el máximo número de fichas a coger es m, déjele a su rival un número de fichas que sea múltiplo de $(m + 1)$ más 1; si no es posible, indique a su rival que debe comenzar cogiendo en primer lugar.
2. Durante el juego, cada vez que su contrincante tome k fichas, coja $(m + 1) - k$.

DOMINGO DE PASCUA

La Pascua de Resurrección es, junto con la Navidad, la fiesta más importante del calendario cristiano. Ahora bien, inicialmente la Pascua era la celebración que el pueblo judío hacía de su marcha de Egipto y, al regirse por un calendario lunar, la hicieron coincidir con la primera luna llena de primavera.

En el Concilio de Nicea (año 325) la Iglesia estableció que la Pascua cristiana tendría lugar el domingo posterior a la primera luna llena después del equinoccio de primavera, que se fijó en el 21 de marzo, cuando antes era el 25 de marzo[7]. ¿Queda claro por qué las vacaciones de Semana Santa bailan tanto de un año para otro?

Por ejemplo, en la figura 3 le muestro en qué fecha cae el domingo de Pascua de los próximos años. Como puede observar, las variaciones son notables en algunos casos.

Año	Pascua
2005	27 marzo
2006	16 abril
2007	8 abril
2008	23 marzo
2009	12 abril
2010	4 abril
2011	24 abril
2012	8 abril
2013	31 marzo
2014	20 abril
2015	5 abril
2016	27 marzo
2017	16 abril

Figura 3. Fechas de Pascua de los próximos años

Al ser la Pascua una festividad tan destacada, no es de extrañar que fueran surgiendo diversos algoritmos para permitir su cálculo de una forma sencilla. Así, en 1800 Gauss expuso un algoritmo que tenía como límite 2299; en 1876 Samuel Butcher publicó otro de duración indefinida; etc.

Seguidamente le detallo otro algoritmo, más sencillo que los anteriores, dado a conocer por Thomas H. O'Beirne en su libro "*Puzzles and Paradoxes*" (1965). Eso sí, sólo es eficaz para años comprendidos entre 1900 y 2099… pero considero que ya es suficiente, ¿no cree?

1. Calcular a = año – 1900
2. Sea b el resto de dividir a entre 19
3. Hallar c, el cociente de la división entre $(7b + 1)$ y 19
4. Sea d el resto de dividir $(11b + 4 - c)$ entre 29
5. Hallar e, el cociente de la división entre a y 4
6. Sea f el resto de dividir $(a + e + 31 - d)$ entre 7
7. Calcular $g = 25 - d - f$

[7] Sin embargo, el calendario juliano de aquella época consideraba que el año tenía 365,25 días, cuando, en realidad, su duración es 365,2422 días. Así pues, no es de extrañar que en el siglo XVI hubiese un desfase evidente entre el calendario real y el oficial. Por este motivo, en 1582, el Papa Gregorio XII propuso la reforma del calendario, que dio origen al nuestro actual.

8. Si g es positivo, la fecha es g de Abril; en caso contrario, es $31+g$ de Marzo.

Así, en la figura 4, puede observar los diferentes valores que se van obteniendo al aplicar el algoritmo anterior, cuando se calcula el día de Pascua del año 2006.

Año	2006
a	106
b	11
c	4
d	5
e	26
f	4
g	16
16	Abril

Figura 4. La Pascua del 2006

De todas formas, si no tiene ganas de hacer cálculos, siempre puede acudir a Internet, donde encontrará la fecha de Pascua con suma facilidad. Por ejemplo, la figura 5 corresponde a una Web donde, una vez introducido el número del año, le mostrará en qué día cae su domingo de Pascua (*Easter day*, en inglés).

Figura 5. Su dirección es: http://www.ely.anglican.org/cgi-bin/easter

LA MULTIPLICACIÓN RUSA

El siguiente algoritmo, cuyo nombre hace alusión a su amplia utilización en Rusia (hasta hace un siglo, más o menos), permite calcular el producto de dos números enteros siguiendo un procedimiento muy diferente del que aprendimos en la escuela.

1. Escriba los dos números que vaya a multiplicar, uno al lado del otro, algo separados.
2. Del mayor calcule su doble y del otro la mitad entera (sin decimales). Escriba estos valores debajo de los anteriores.
3. Repita el paso 2, hasta que alcance el 1 al dividir.
4. Sume los números de la columna donde anota los dobles cuyo número correspondiente de la otra columna sea impar. Dicha suma es el producto buscado.

Veamos un ejemplo y comprobará que este algoritmo es muy sencillo de aplicar. Supongamos que nos interesa multiplicar 54 por 23. Siguiendo los tres primeros pasos del algoritmo anterior, se construye la tabla de la figura 6.

54	23
108	11
216	5
432	2
864	1

Figura 6. Para multiplicar 54 por 23

Ahora, sólo tiene que sumar los números de la columna izquierda que tienen a su derecha un número impar; es decir, 54, 108, 432 y 864. El resultado, 1242 (figura 7), es precisamente el valor de 54 x 23.

54	23
108	11
216	5
432	2
864	1
1242	

Figura 7. 54 por 23 es 1242

Si practica este algoritmo con números pequeños apreciará que es muy cómodo y, sobre todo, fácil de aprender, ya que no exige memorizar las tablas de multiplicar. Sin embargo, cuando los números son mayores, resulta un tanto lento y tedioso… para los seres humanos (entenderá esta puntualización al leer el siguiente apartado).

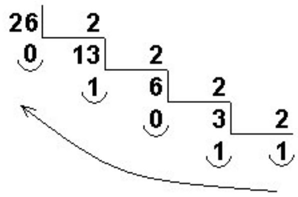
SISTEMA BINARIO

Las personas utilizamos el sistema decimal (base 10) y, ocasionalmente, el sexagesimal (base 60) al calcular con ángulos y tiempo. En cambio, para el ordenador es mucho más eficaz trabajar en sistema binario (base 2), ya que puede identificar un dígito binario, 0 o 1, comprobando si pasa o no corriente por un determinado circuito.

Cuando queremos pasar un número del sistema decimal al binario, podemos utilizar el siguiente algoritmo:

1. Dividir por 2 el número.
2. Dividir por 2 el cociente entero obtenido.
3. Repetir el paso anterior hasta llegar a 1.
4. Escribir, de izquierda a derecha, el último cociente (1) y los restos que se han obtenido al dividir, en orden inverso.

Si no conocía este algoritmo, es posible que su enunciado le parezca poco claro, así que nada mejor que un ejemplo para acabarlo de entender. En la figura 8 se muestra cómo pasar 26 a binario, donde se dividen por 2, sucesivamente, los números 26, 13, 6 y 3.

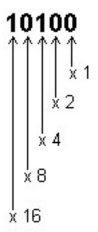


$26_{(10} = 11010_{(2}$

Figura 8. 26 es 11010 en binario

El algoritmo para convertir de binario a decimal también es muy simple. De derecha a izquierda, el primer dígito se multiplica por 1, el segundo por 2, el tercero por 2 al cuadrado, el cuarto por 2 al cubo, etc., sumándose los productos obtenidos.

En la figura 9 puede ver un ejemplo que ilustra este algoritmo de cambio de base.



$10100_{(2} = 4 + 16 = 20$

Figura 9. El número binario 10100 es 20

Ahora que ya sabemos pasar de decimal a binario y viceversa, vamos a analizar un par de detalles. En primer lugar, ¿cómo se multiplica, en binario, un número por 2? ¡Muy bien! Sólo hay que añadir un cero a su derecha y asunto concluido. Es decir, si 20 en binario era 10100, 40 será 101000.

Y, ¿cómo se halla, en binario, el cociente entero de dividir por 2? Pues también es muy sencillo; basta con suprimir su cifra de la derecha. Por ejemplo, como 27 es 11011 en binario, su mitad entera (13) es 1101.

En otras palabras, como el ordenador trabaja en binario le resulta sumamente rápido multiplicar por 2 y dividir por 2. Además, lo de sumar tampoco es complicado para él. ¿Le suena todo esto?

¡Claro que sí! Es el proceso que se sigue en la multiplicación rusa. Por tanto, se trata de un algoritmo que le va de perlas al ordenador para multiplicar dos números enteros. ¡Quién nos iba a decir que sería tan útil un antiguo algoritmo que sólo parecía curioso!

JUEGO DEL NIM

Vamos ahora con el segundo juego prometido, el Nim, cuyo algoritmo ganador no es tan evidente como el del Quitpal.

En el Nim también se utilizan fichas, si bien las reglas varían algo. Se juega con varios montones de fichas, pudiendo haber un número cualquiera de fichas en cada montón. Participan dos personas y, alternativamente, cada una coge las fichas que desee de cualquier montón, pero sólo de uno. Ahora, gana quien coge la última ficha.

Si lo desea, puede hacer un alto en la lectura y practicar un rato con la versión más popular del Nim[8], que se juega con cinco montones de fichas, que tienen, respectivamente, 1, 2, 3, 4 y 5 fichas, que se colocan tal y como se muestra en la figura 10.

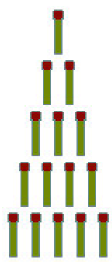


Figura 10. Versión más popular del Nim

[8] El nombre de Nim deriva del alemán *nimm* (coger). En la película “*El año pasado en Marienbad*” de Alain Resnaís (1962), aparece varias veces este juego y, por ello, el Nim también se conoce por Marienbad.

¿Ha sido capaz de encontrar el algoritmo ganador? Le felicito, porque yo tardé bastante (y perdí gran cantidad de partidas). Si la versión simple del Nim no resulta nada sencilla, imagine cuando se generaliza.

El algoritmo ganador exige trabajar en base 2. Escriba en binario los números que hay en cada montón, uno bajo otro, y deje siempre a su contrincante una posición en la que el número de 1 de cada columna sea par.

Por ejemplo, supongamos que hay tres montones, con 3, 5 y 8 fichas. Al escribir estos números en binario, obtenemos:

11 (3)
101 (5)
1000 (8)

Por tanto, si toma 2 fichas del montón de 8, su rival se encontrará ante una posición perdedora, ya que la suma de cada columna es par.

11 (3)
101 (5)
110 (6)

Así pues, puede comprobar que, en la versión popular del Nim (con 1-2-3-4-5 fichas), la jugada inicial ganadora consiste en coger una ficha de los montones 1, 3 o 5.

CÓDIGOS DE CONTROL

Para terminar con el artículo, voy a comentarle unos algoritmos con los que convivimos diariamente y que casi nunca tenemos presentes. Mediante ellos se generan dígitos o letras que tienen por finalidad detectar posibles errores de tecleo o de lectura.

Comencemos con el más habitual: **la letra del NIF**. Como ya sabe, para controlar que el número del DNI se ha introducido correctamente, se añade a su final una letra, conformando así el NIF.

El algoritmo para obtener la letra del NIF consiste en dividir el número del DNI entre 23. Una vez obtenido el resto, la asignación de letra se rige por la tabla siguiente:

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

Así, por ejemplo, si el número del DNI es 12345678, al dividirlo por 23 se obtiene 536768 de cociente y 14 de resto. Con este último número, vamos a la tabla anterior y averiguamos que la letra buscada es Z.

Otro código de control con el que nos topamos en nuestro trabajo es el **ISBN** (*International Standard Book Number*). Los diez dígitos que identifican un libro están repartidos en cuatro bloques: las dos primeras cifras un identificador de grupo (a España le corresponde el 84), las tres siguientes determinan la editorial, las cuatro siguientes aluden al libro dentro de la editorial[9] y, finalmente, el último dígito controla que los datos anteriores se han introducido sin errores.

El algoritmo que permite obtener el código de control, a partir de los nueve primeros dígitos, es el siguiente:

1. Se multiplica la primera cifra de la izquierda por 10, la siguiente por 9 y así sucesivamente, hasta el noveno dígito, que se multiplica por 2.
2. Se suman los productos anteriores.
3. Se divide la suma anterior entre 11.
4. Si el resto es 0, es el código de control; si es 1, se escribe X. En los demás casos, el código de control es 11 menos el resto.

Por ejemplo, como puede comprobar en la figura 11, los nueve primeros dígitos del ISBN de mi último libro son 844151645. Veamos qué resulta al seguir los pasos anteriores:

1. Se hacen las multiplicaciones 8x10, 4x9, 4x8, …, 5x2
2. Al sumar los productos obtenidos, se obtiene 236.
3. Al dividir 236 entre 11, el resto es 5.
4. El código de control es 6 (11 – 5).

ISBN 84-415-1645-6
9 788441 516458

Figura 11. ISBN y código de barras

El **código de barras** EAN (*Europe Article Number*) más habitual es el EAN13, que consta de 13 dígitos, el

[9] Esta agrupación de dígitos es variable y depende de la cantidad de libros que prevé producir la editorial o el grupo.

último de los cuales es un dígito de control. El algoritmo para calcularlo, a partir de los doce dígitos anteriores, es el siguiente:

1. Se suman los dígitos correspondientes a posiciones impares, comenzando por la izquierda.
2. Se suman los dígitos correspondientes a posiciones pares y el resultado se multiplica por 3.
3. Se suman los valores hallados en los dos pasos anteriores y nos quedamos sólo con la cifra *c* de las unidades de la suma.
4. Si *c* es 0, es el dígito de control. En los demás casos, el dígito de control es 10 menos *c*.

Por ejemplo, en la figura 11 los doce primeros dígitos son 978844151645. Al seguir los pasos anteriores, se tiene:

1. Al calcular 9 + 8 + 4 + 1 + 1 + 4, resulta 27
2. Al hallar 7 + 8 + 4 + 5 + 6 + 5, da 35. Al multiplicar 35 por 3, se obtiene 105.
3. La suma de 27 y 105 es 132. Por tanto *c* = 2.
4. El dígito de control es 8 (10 – *c*)… algo que ya sabíamos por la figura 11.

Como puede imaginar, siempre que haya datos que se almacenan o se leen desde un ordenador, los dígitos de control son imprescindibles para detectar errores; así, seguro que los encuentra en su pasaporte, en los cheques, en los billetes de avión, etc.

Lógicamente, no puedo comentarle todos los algoritmos generadores (muchos los desconozco), porque el artículo se alargaría bastante. Así que voy a terminar explicándole cómo averiguar el dígito de control que llevan las tarjetas de crédito.

Las tarjetas de crédito constan de 16 dígitos (al menos las que hay en mi cartera), siendo el primero 4 (Visa) o 5 (Mastercard) y el último el código de control. Para hallar éste, se utiliza el siguiente algoritmo:

1. Comenzando por la izquierda, se multiplican por dos las cifras que ocupan posición impar. Si el resultado de alguno de estos productos es mayor que 9, se sustituye por la suma de sus dos cifras. Después, se suman los números obtenidos anteriormente.
2. Se suman los dígitos que ocupan posición par (excluyendo el dígito de control, claro está)
3. Se suman los valores hallados en los dos pasos anteriores. Sea *c* la cifra de las unidades de esta nueva suma.
4. Si *c* es 0, es el dígito de control. En los demás casos, el dígito de control es 10 menos *c*.

Por ejemplo, supongamos que los 15 primeros dígitos de una tarjeta son 4000 0012 3456 789. Para calcular el dígito de control, según el algoritmo anterior, hacemos lo siguiente:

1. Se suman los números 8 (4x2), 0 (0x2), 0 (0x2), 2 (1x2), 6 (3x2), 1 (5x2=10 y 1+0=1), 5 (7x2=14 y 1+4=5) y 9 (9x2=18 y 1+8=9). El resultado es 31.
2. Se suman los dígitos 0, 0, 0, 2, 4, 6 y 8. El resultado es 20.
3. La suma de 31 y 20 es 51. Por tanto *c* = 1.
4. El dígito de control es 9 (10 – *c*).

Después de haber visto tantos algoritmos para controlar errores, la pregunta que surge es evidente: ¿Realmente detectan los errores que se cometen? Como no hay nada perfecto, la respuesta es… depende.

Los gazapos más comunes al teclear consisten en escribir un dígito en lugar de otro (error simple) o intercambiar el orden de dos dígitos consecutivos (error de transposición) y, por ello, los algoritmos de control se centran en ellos.

De hecho, todos los algoritmos de control analizados anteriormente detectan los errores simples. En cuanto a los de transposición, se detectan siempre en la letra del NIF y en el ISBN; en las tarjetas de crédito la probabilidad de detección es del 99%, pero en el código de barras sólo es del 89%.

En otras palabras, le recomiendo que repase la cuenta del supermercado… No, no se lo tome a risa. En una ocasión que fui de compras a un hiper, adquirí 36 botellas de agua y no revisé el ticket de compra hasta llegar a casa. Entonces, comprobé que me habían cobrado ¡36 almohadas!… Eso sí, no tuve ningún problema para recuperar mi dinero.