

2.

Numeración posicional

dígito. (Del lat. *digitus*, dedo).

1. m. *Mat.* número dígito.

número dígito.

1. m. *Mat.* El que puede expresarse con un solo guarismo.

*Diccionario de la Lengua Española
Vigésima segunda edición
Real Academia Española*

Los elementos usados en los computadores tienen dos estados, por lo cual en los procesadores el sistema binario es natural e incluso se impone. Esto introduce la necesidad de estudiar el sistema binario y los métodos para hacer cambios de base.

Sin embargo, en computación, no sólo se utiliza la base 2; por comodidad se suele usar la base 16 y, en raras ocasiones, la base 8. Estas bases son útiles, ya que los números binarios pueden ser muy largos, y se vuelven difíciles de leer; la base 16 es más sintética, y el cambio de base 16 a base 2 —o viceversa— es fácil de hacer.

En este capítulo, se presentan los sistemas de numeración posicional de manera general, no solo el binario, ya que así se adquiere una mejor comprensión de su funcionamiento.

El capítulo empieza presentando el concepto de base de numeración, luego de lo cual se desarrolla una formalización matemática de las mismas así como algoritmos para cambio de base.

2.1 INTRODUCCIÓN

¿Qué es un número? Las personas que han intentado responder a esta pregunta han descubierto una cosa: los demás no están de acuerdo con su respuesta. Debido a esto, no nos vamos a concentrar en la naturaleza de los números sino en su representación. Veremos qué hay detrás de los dígitos que usamos todos los días.

Estamos acostumbrados a llamar "números" a secuencias de dígitos, como por ejemplo "51". Esto no es preciso; un número es una entidad abstracta. Lo que sí

podemos decir es que la secuencia de dígitos "51" representa un número, así como también lo hacen los caracteres "LI" en la numeración romana.¹

Analicemos el problema de representar los números. En primera aproximación se puede pensar en un sistema similar al lenguaje: una palabra por objeto. Este método, que intentó "Funes el memorioso" en un cuento de Jorge Luis Borges, es decepcionante cuando se cae en cuenta de que hay infinitos números.

Se necesita un método más articulado. Se dio un paso adelante con los griegos. Ellos desarrollaron un sistema (copiado por los romanos) más flexible aunque todavía eran necesarios infinitos símbolos para poder representar todos los números.

Aparentemente los hindús inventaron el sistema posicional que utilizamos hoy día. Su característica fundamental consiste en que con un conjunto finito de símbolos se puede representar un conjunto infinito de números.

La idea básica es sencilla; se trata de contar agrupando. Veamos un caso análogo: se pueden contar frutas por unidades, pero si hay muchas es mejor contar por bultos de frutas; si aun estos son excesivos, se podría contar por "cargas". Así, una cierta cantidad de frutas puede expresarse como: cinco cargas, tres bultos y doce unidades.

El primer problema que encontramos es definir cuántas frutas componen un bulto y cuántos bultos una carga. Una solución es la siguiente: un agrupamiento superior se compone de un número fijo de agrupamientos inferiores. En nuestro caso podemos considerar la carga compuesta por 20 bultos y el bulto de 20 frutas. En el ejemplo anterior teníamos 5 cargas, es decir 100 bultos ($5 \cdot 20$); como además teníamos 3 bultos, en total tenemos 103, cada uno con 20 frutas, lo cual da 2060 frutas; esto, más las 12 sueltas, da 2072 frutas.

En el ejemplo anterior, podemos notar que para cada nivel de agrupamiento sólo necesitamos 20 números básicos: del 0 al 19. En efecto, si hay más de diecinueve se puede trasladar parte al agrupamiento superior; por ejemplo, si hay 25 frutas podemos hablar, más bien, de un bulto y 5 frutas. Estos "números básicos" es lo que llamamos *dígitos*; en nuestro ejemplo tenemos 20 dígitos

El sistema que hemos mostrado es limitado, ya que el número máximo que podemos representar es "diecinueve cargas, diecinueve bultos y diecinueve frutas". Es claro que necesitamos agrupamientos superiores a la carga, pero si nos pusiéramos a inventar más, tendríamos el mismo problema: siempre habría un límite máximo. Por otro lado, tampoco podemos tener infinitas palabras para representar estos agrupamientos superiores. ¿Cómo resolver este problema? Puesto que no podemos tener infinitas palabras para denotar los agrupamientos pues es mejor no tener ninguna y buscar otro método: vamos a representar los agrupamientos por su posición. Así nuestro ejemplo inicial se escribiría:

5, 3, 12

¹ Y como la palabra "perro" representa un perro, sin que por ellos tenga pulgas.

No tenemos nombres para los agrupamientos, pero sabemos que, en un agrupamiento determinado, cada elemento vale por veinte de los que están inmediatamente a la derecha, y esto es suficiente.

Ahora no hay límite para los números que podemos representar; si es necesario podemos agregar más números a la izquierda que representen agrupamientos superiores. Así, en el número:

2, 5, 3, 12

no sabemos el nombre del agrupamiento que corresponde al "2", pero sabemos que vale veinte de los anteriores (veinte cargas), cada uno de los cuales vale veinte de los anteriores, etc. Hemos cambiado la representación de nombres a posiciones.

Por supuesto, si alguno de los agrupamientos no tiene componentes, por ejemplo: "14 cargas, ningún bulto y 7 frutas", indicamos su ausencia por medio del número cero:

14, 0, 7

Debe notar que no hay tal cosa como un sistema "natural" para contar. Si contamos en la base diez es, probablemente, porque tenemos diez dedos; pero hay otros sistemas: los mayas tenían un sistema base veinte; a veces se cuenta por "docenas" y "gruesas", lo cual constituye un sistema base doce rudimentario; cuando se lee la hora, se está usando un sistema base sesenta (sexagesimal) casi sin darse cuenta, herencia que nos dejó Mesopotamia. En nuestro ejemplo desarrollamos un sistema base veinte (vigesimal) sin mayores problemas.

Cuando se usan bases inferiores a diez, los dígitos que conocemos son suficientes; sin embargo, si usamos una base superior a diez necesitamos más dígitos. La convención en esos casos es utilizar las letras del alfabeto. Así, el dígito diez se representa por la letra "A", el dígito once por la "B" etc. Nuestro ejemplo inicial (5 cargas, 3 bultos, 12 doce frutas) se escribiría:

(53C)₂₀

Donde el subíndice "20" indica que la base de numeración es veinte. También usaremos otra notación: se adjunta una letra al final del número para indicar la base: B para binario, Q para octal, D para decimal y H para hexadecimal. Por ejemplo, diez, en cada una de estas bases, se escribe: 1010B, 12Q, 10D, AH.

2.2 FORMALIZACIÓN

En esta sección se definen y modelan matemáticamente los sistemas de numeración posicional. Estudiaremos métodos para describir en qué consiste la representación de un número, y métodos para cambiar de una representación a otra.

Los números enteros

Definamos la representación de un número en una base b como una secuencia de enteros, cada uno de los cuales tiene un valor entre 0 y $b-1$. Los dígitos de este sistema son los enteros entre 0 y $b-1$, los dos inclusive.

Sea $b > 1$. Definimos:

$$\text{Dígitos}_b = \{ x \mid 0 \leq x < b \}$$

$$\text{Representación}_b = \{ (d_n \dots d_0) \mid d_i \in \text{Dígitos}_b \}$$

Por ejemplo, en el sistema base 2 (binario) tenemos:

$$\text{Dígitos}_2 = \{ 0, 1 \}$$

Y las representaciones de los números serían del estilo (11011).

Para el sistema base 16 (hexadecimal) tenemos:

$$\text{Dígitos}_{16} = \{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F \}$$

Con representaciones del estilo (1B).

Por comodidad, se adopta la convención de escribir las representaciones con un subíndice que indica la base. Los ejemplos serían: $(11011)_2$ y $(1B)_{16}$.

Necesitamos ahora definir a qué número corresponde cada una de estas secuencias; necesitamos un método para interpretarlas: si estamos trabajando en la base b , sabemos que cada elemento de un agrupamiento vale por b elementos del agrupamiento inmediatamente inferior; de esto se deduce que si multiplicamos el dígito en cuestión por b , podemos saber a cuántos elementos del agrupamiento inmediatamente inferior corresponde. Al resultado le podemos aplicar el mismo proceso hasta llegar a las unidades.

Matemáticamente podemos expresar lo anterior por medio de una función, que llamaremos Entero_b , que toma una representación y retorna su valor:

$$\text{Entero}_b(d_n \dots d_0) = d_n * b^n + \dots + d_0 * b^0 = \sum_{i=0}^n d_i * b^i$$

Dicho en otras palabras: los números en los sistemas de numeración posicional son representados por polinomios en b donde los coeficientes son los d_i .

Por ejemplo, vamos a interpretar $(11011)_2$:

$$b = 2$$

$$\text{Entero}_2(11011) = 1 * 2^4 + 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0$$

En total, esto da: $\text{Entero}_2(11011) = 16 + 8 + 0 + 2 + 1 = 27$. Tenemos entonces que $(11011)_2$ representa al número 27.

Ahora tomemos el $(1B)_{16}$:

$$b = 16$$

Recordemos, además, que "B" es sólo un símbolo que representa el dígito 11.

$$\text{Entero}_{16}(1B) = 1 * 16^1 + 11 * 16^0$$

En total esto da: $\text{Entero}_{16}(1B) = 1 * 16 + 11 = 27$. Lo cual quiere decir que $(1B)_{16}$ también representa el número 27.

Cambio de base

La definición de la sección anterior es, además, un algoritmo para pasar de cualquier base a la base 10: si se tiene un número en una base diferente de 10, basta con convertir cada dígito a la base 10 y después calcular el polinomio.

Los ejemplos de la sección anterior ilustran este método: convertimos $(11011)_2$ y $(1B)_{16}$ a decimal.

Queda el problema de cómo pasar de la base 10 a otras bases. Es decir, tenemos un número N y queremos encontrar $(d_n \dots d_0)_b$.

$$\text{Por definición:} \quad N = \text{Entero}_b(d_n \dots d_0) = \sum_{i=0}^n d_i * b^i$$

Si dividimos el número N por b , obtenemos:

$$\frac{N}{b} = \sum_{i=0}^n \frac{d_i * b^i}{b} = \left(\sum_{i=1}^n d_i * \frac{b^i}{b} \right) + \frac{d_0}{b} = \sum_{i=1}^n d_i * b^{i-1} + \frac{d_0}{b}$$

Note que, en la última ecuación, la sumatoria es un número entero (¿por qué?) mientras que la fracción (d_0/b) necesariamente es menor que 1 (¿por qué?); lo cual quiere decir que la sumatoria es el cociente de la división N/b , y que d_0/b es la parte fraccionaria, o sea que d_0 es el residuo de la división.

En conclusión: si tenemos un número N y lo dividimos por b , el residuo de la división corresponde al dígito menos significativo del número cuando se representa en la base b .

Si ahora tomamos el cociente obtenido, y lo dividimos por b , vamos a obtener:

$$\frac{1}{b} * \sum_{i=1}^n d_i * b^{i-1} = \sum_{i=1}^n \frac{d_i * b^{i-1}}{b} = \left(\sum_{i=2}^n d_i * \frac{b^{i-1}}{b} \right) + \frac{d_1}{b} = \sum_{i=2}^n d_i * b^{i-2} + \frac{d_1}{b}$$

El residuo es d_1 ; es decir, el siguiente dígito de la representación de N en base b . Podemos continuar aplicando este método para obtener los otros dígitos, como usted puede verificar.

A partir de lo anterior podemos desarrollar un método para convertir de la base 10 a otras bases: al dividir un número por el valor de la base a la cual se quiere pasar (b), el residuo nos indica cuál es el primer dígito del número en la base b ; se continúa realizando el mismo proceso con el cociente de la división, obteniendo el segundo dígito, el tercero y así sucesivamente. Veamos un ejemplo, pasar el número 27 a la base 4:

$$\begin{aligned} [27 / 4] &= 6, \text{ con un residuo de } 3 \\ [6 / 4] &= 1, \text{ con un residuo de } 2 \end{aligned}$$

Se continúa haciendo las divisiones hasta que el resultado de la división sea menor que la base; en el ejemplo anterior, ocurre en la segunda división (el cociente es 1). El número buscado está compuesto por el cociente de la última división y los residuos de las otras divisiones; en nuestro ejemplo, esto es 123. Entonces:

$$\text{Entero}_4(123) = 27.$$

Hay que tener cuidado cuando se trabaja con bases superiores a 10, recuerde que en este caso los dígitos pueden ser mayores que 10. A manera de ejemplo, pasemos 1627 a la base 16:

$$\begin{array}{r} 1627 \quad | \quad 16 \\ 11 \quad 101 \quad | \quad 16 \\ \quad \quad 5 \quad 6 \end{array}$$

El resultado es $(6, 5, 11)$, es decir $(65B)_{16}$.

Matemáticamente podemos sintetizar lo anterior de la siguiente manera: si un polinomio $P_n(x)$ de grado n se divide por x , se obtiene un polinomio de grado $n-1$, y el residuo de la división es el coeficiente d_0 que multiplica a x^0 .

Las fracciones

El sistema anterior se puede extender a las fracciones. Al trabajar en una base b , cada elemento de un cierto agrupamiento vale por b elementos del agrupamiento inmediatamente inferior. Por tanto, el agrupamiento inferior a la unidad debe ser $1/b$ (es decir, b^{-1}), el siguiente será $1/b^2$ (es decir, b^{-2}) y así sucesivamente.

En consecuencia, la interpretación de una fracción en base b viene dada por:

$$\text{Fracción}_b(d_{-1} \dots d_{-k}) = d_{-1} * b^{-1} + \dots + d_{-k} * b^{-k} = \sum_{i=-1}^{-k} d_i * b^i$$

Para indicar que una secuencia es de dígitos es una fracción, le ponemos un punto al principio, por ejemplo: $(.121)_3$.

Para pasar una fracción de una base cualquiera a la base 10, lo mismo que en el caso anterior, basta con calcular la sumatoria. Veamos esto con la fracción $(.121)_3$:

$$\text{Fracción}_3(.121) = 1 * 3^{-1} + 2 * 3^{-2} + 1 * 3^{-3} = 0.\overline{3} + 0.\overline{2} + 0.\overline{037} = 0.\overline{592}$$

Donde la línea sobre el número indica que la fracción se repite al infinito.²

En el caso contrario, pasar de la base 10 a otra base, el método es parecido al de los enteros, pero hay que multiplicar en lugar de dividir. Veámoslo en detalle: tenemos una fracción f en la base 10 y queremos encontrar los d_i en la base b , tales que:

$$f = \sum_{i=-1}^{-k} d_i * b^i$$

Multiplicando por b , obtenemos:

$$f * b = \sum_{i=-1}^{-k} d_i * b^i * b = \left(\sum_{i=-2}^{-k} d_i * b^{i+1} \right) + d_{-1}$$

En la última ecuación, observamos que la sumatoria es la parte fraccionaria, y d_{-1} es la parte entera. Esto es suficiente: basta con hacer multiplicaciones sucesivas por la base a la cual queremos pasar; cada vez tomamos la parte entera, que será un

² Note que una fracción finita en una base puede dar una fracción infinita en otra base.

nuevo dígito de la expansión fraccionaria en la base b , y continuamos el proceso con la parte fraccionaria hasta que esta sea 0. Por ejemplo, pasemos 0.890625 a la base 4:

$$\begin{aligned} 0.890625 * 4 &= 3.5625, & \text{parte entera } 3 \\ 0.5625 * 4 &= 2.25, & \text{parte entera } 2 \\ 0.25 * 4 &= 1.0, & \text{parte entera } 1 \end{aligned}$$

En total tenemos $(.321)_4$, o, dicho de otra forma:

$$\text{Fracción}_4(.321) = 0.890625$$

Puede ocurrir que la parte fraccionaria del resultado de la multiplicación nunca llegue a ser 0; en tal caso se trata de una expansión infinita. En este caso es necesario fijar una precisión y calcular dígitos hasta llegar a la precisión deseada.

También se puede parar en cuanto se identifique el periodo. Cuando se obtenga una fracción que ya haya aparecido, se habrá identificado el periodo; efectivamente, este estará constituido por todos los dígitos que se hayan calculado desde la primera vez que se multiplicó la fracción repetida hasta su segunda aparición. Como ejemplo pasemos $0.1\bar{3}$ a la base 3:

$$\begin{aligned} 0.1\bar{3} * 3 &= 0.3\bar{9} = 0.4 \text{ (¿Por qué?)}, & \text{parte entera } 0 \\ 0.4 * 3 &= 1.2, & \text{parte entera } 1 \\ 0.2 * 3 &= 0.6, & \text{parte entera } 0 \\ 0.6 * 3 &= 1.8, & \text{parte entera } 1 \\ 0.8 * 3 &= 2.4, & \text{parte entera } 2 \end{aligned}$$

Llegamos a la fracción 0.4, que ya se había presentado, y concluimos que el periodo es 1012; luego el número en base 3 se escribe: $(0.01012)_3$. El comienzo de la fracción puede no pertenecer al periodo —en el ejemplo es el primer cero—.

2.3 CAMBIO DE BASE: MÉTODOS RÁPIDOS

En las secciones anteriores hemos visto métodos para pasar de una base a otra. Estos métodos son seguros, pero pueden ser incómodos si la conversión se hace manualmente. En este caso hay formas más prácticas de realizar dicha conversión.

El primer sistema que vamos a estudiar es útil, sobre todo, como algoritmo de cambio de base para implementar en computador. El método está basado en la factorización de la sumatoria que define el valor de una representación, es decir:

$$\begin{aligned} \text{Entero}_b(d_n \dots d_0) &= d_n * b^n + d_{n-1} * b^{n-1} + \dots + d_1 * b + d_0 \\ &= (\dots(d_n * b + d_{n-1}) * b + \dots + d_1) * b + d_0 \end{aligned}$$

Como puede notar, sólo se hacen n multiplicaciones; en el otro caso hay que hacer $n*(n+1)/2$ multiplicaciones. Por ejemplo, para saber el valor de $(B1A)_{16}$:

$$\text{Entero}_{16}(B1A) = (11*16 + 1)*16 + 10 = 2842$$

Veamos otro sistema para pasar números pequeños de binario a decimal. Se escriben, de derecha a izquierda, las potencias de dos. Después se escribe el número debajo y se suman las potencias que quedan encima de los unos:

16	8	4	2	1	
1	0	1	1	0	luego, $\text{Entero}_2(10110) = 22$
0	1	0	1	1	luego, $\text{Entero}_2(1011) = 11$

El método que presentamos a continuación se utiliza únicamente cuando las bases implicadas son dos y una potencia de dos. Supongamos que tenemos que pasar de la base 2 a la base 2^k . Procedemos así: se parte el número en la base 2, de derecha a izquierda, en grupos de k dígitos. En seguida, cada uno de estos grupos se interpreta como un dígito en la base 2^k . Por ejemplo, para pasar el número $(1100101101011)_2$ a la base 8:

$8 = 2^3$, por lo tanto, tenemos que dividir en grupos de 3:
 1 100 101 101 011 tomamos cada grupo como un dígito base 8:
 1 4 5 5 3 en conclusión, el número base 8 es:
 $(14553)_8$

El mismo número base 16 sería:

$16 = 2^4$, hay que dividir en grupos de 4:
 1 1001 0110 1011 tomamos cada grupo como un dígito base 16
 1 9 6 11 el 11, como dígito, se escribe B, luego:
 $(196B)_{16}$

Este método también se puede utilizar para pasar de la base 2^k a binario: cada dígito se escribe como k dígitos binarios (si el dígito se puede escribir usando menos de k dígitos binarios, se completa con ceros a la izquierda); luego se escribe todo junto. Ejemplos: leer, de abajo hacia arriba, los dos ejemplos anteriores.

Es importante dominar el último método, ya que, aunque los computadores son binarios, se suele trabajar en base 16 ó 8; así se evita escribir largas secuencias de 1 y 0 (los números hexadecimales son más compactos). En lugar de hablar de $(01011100)_2$, hablamos de $(5CH)_{16}$. Debe tener muy en cuenta que el sistema hexadecimal es sólo una forma compacta de escribir los números; los valores en el computador siempre son binarios.

2.4 OPERACIONES ARITMÉTICAS EN OTRAS BASES

Estamos habituados a efectuar operaciones aritméticas solo en base diez. En esta sección vamos a generalizar los algoritmos aritméticos a otras bases.

Precisemos: no nos interesa definir la operación matemática de la suma, sino definir un algoritmo para, a partir de dos números representados usando numeración posicional, calcular cómo debe ser el resultado de su suma en el mismo sistema. Vamos a operar sobre las representaciones de números. Formalmente:

$$\begin{aligned} x, y, z &\in \text{Representación}_b \\ \text{Algoritmo}_b(x, y) &= z \Rightarrow \\ \text{Entero}_b(x) + \text{Entero}_b(y) &= \text{Entero}_b(z) \end{aligned}$$

El algoritmo de suma en la base b (Algoritmo_b) opera sobre representaciones.

La suma

Vamos a empezar por la suma. Tenemos dos números X y Y tales que:

$$\begin{aligned} \text{Entero}_b(x_n \dots x_0) &= X \\ \text{Entero}_b(y_n \dots y_0) &= Y \end{aligned}$$

X y Y tienen la misma cantidad de dígitos.³ Ya que sus representaciones son polinomios, en principio, debemos hacer la suma de dos polinomios:

$$X + Y = \sum_{i=0}^n x_i * b^i + \sum_{i=0}^n y_i * b^i = \sum_{i=0}^n (x_i + y_i) * b^i$$

Sin embargo, de lo anterior se deduciría que los términos $(x_i + y_i)$ son los dígitos que componen el resultado, pero esto no necesariamente es así, puesto que $x_i + y_i$ puede ser mayor o igual a b , y, en consecuencia, no sería un dígito válido.

Esta dificultad se puede salvar de la siguiente manera: si $(x_i + y_i) \geq b$, necesariamente debe ser de la forma: $(x_i + y_i) = k * b + r$, donde k es el cociente de la división $(x_i + y_i) / b$ y r es el residuo de la misma.

Esto nos da las bases para el algoritmo de suma:

- Se suman los dos polinomios (es decir, las parejas de dígitos de los dos números que se encuentran en la misma posición).
- Se recorren de derecha a izquierda (de menos significativo a más significativo) los $(x_i + y_i)$:
 - Si $(x_i + y_i) < b$, se deja así (es un dígito válido).
 - Si $(x_i + y_i) \geq b$, se convierte en $k * b + r$. En el resultado, el dígito en la posición i será r , y k será el acarreo al siguiente dígito; es decir, el dígito $i+1$ se convierte en $(x_{i+1} + y_{i+1} + k)$.

³ Si este no fuera el caso, podríamos agregar ceros a la izquierda al más pequeño hasta darles la misma longitud.

La última acción se justifica porque el i -ésimo dígito del número corresponde con el término $(x_i + y_i) * b^i$ del polinomio; si $(x_i + y_i) \geq b$, efectuamos el reemplazo:

$$(x_i + y_i) * b^i = (k * b + r) * b^i = k * b^{i+1} + r * b^i$$

Como r queda multiplicado por b^i , esto quiere decir que es el i -ésimo dígito, y como k queda multiplicado por b^{i+1} , esto quiere decir que hace parte del dígito $i+1$.

Hemos puesto un número encima del otro y sumado los dígitos por columnas. El módulo b de cada suma es el dígito de la columna, y el cociente de la división se suma a la columna siguiente. Expresado por medio de ecuaciones, esto es:

$$\begin{aligned} c_0 &= 0 \\ d_i &= (x_i + y_i + c_i) \text{ MOD } b, & 0 \leq i \leq n+1 \\ c_{i+1} &= \lfloor (x_i + y_i + c_i) / b \rfloor, & 0 \leq i \leq n \end{aligned}$$

Donde c_i es el acarreo al siguiente, y los d_i son los dígitos que representan la suma.

Veamos un ejemplo en hexadecimal:

$$\begin{array}{r} 8B3 \\ + 95A \\ \hline 120D \end{array}$$

- En la primera columna, tenemos los dígitos A y 3, A vale 10, luego tenemos $10+3 = 13$; $13 < 16$, luego queda igual (pero se escribe D en hexadecimal).
- En la segunda columna, tenemos los dígitos 5 y B, el dígito B vale 11, luego tenemos $11+5 = 16$; no es menor que 16, luego dividimos por 16, y obtenemos residuo 0 y cociente 1; o sea que el resultado es cero con un acarreo de 1.
- Por último, en la tercera columna, tenemos los dígitos 8 y 9, y el acarreo de 1, luego efectuamos la suma $8+9+1 = 18$; como no es menor que 16, efectuamos la división por 16 y obtenemos residuo 2 y cociente 1.

La multiplicación

Empecemos por el caso de la multiplicación de un número por un dígito. Sean:

$$\begin{aligned} \text{Entero}_b(x_n \dots x_0) &= X \\ \text{Entero}_b(y_0) &= Y \end{aligned}$$

En términos de polinomios, la operación es:

$$Y * X = y_0 * \sum_{i=0}^n x_i * b^i = \sum_{i=0}^n x_i * y_0 * b^i$$

La situación es similar a la de la suma: si $x_i * y_0 < b$, es un dígito válido y se deja así; si es mayor o igual, se convierte en: $(x_i * y_0) = k * b + r$, donde k es el cociente de la división $(x_i * y_0) / b$ y r es el residuo de la misma.

En consecuencia, las ecuaciones que definen los dígitos del resultado (los d_i) son:

$$\begin{aligned} c_0 &= 0 \\ d_i &= (x_i * y_0 + c_i) \text{ MOD } b, & 0 \leq i \leq n+1 \end{aligned}$$

$$c_{i+1} = [(x_i * y_0 + c_i) / b], \quad 0 \leq i \leq n$$

Es decir, multiplicamos cada dígito del número X por el dígito y_0 . El módulo b de esta multiplicación es uno de los dígitos del resultado; el cociente es lo que se lleva al siguiente. Un ejemplo en base 5 es:

$$\begin{array}{r} 2304 \\ * 3 \\ \hline 12422 \end{array}$$

- Tenemos que $3*4$ da 12, y este no es un dígito válido en base 5, luego dividimos por 5 y obtenemos un residuo de 2 y un cociente de 2; así que el resultado es 2 y llevamos 2.
- En seguida, $3*0 + 2 = 2$; es un dígito válido en base 5, luego lo dejamos igual.
- El siguiente es $3*3$, que da 9, como no es un dígito válido, dividimos por 5, y obtenemos un cociente de 1 con un residuo de 4.
- Por último, $3*2 + 1$ da 7; dividimos por 5 y obtenemos 2 y llevamos 1.

Para multiplicar dos números, se procede como en decimal: se multiplica uno de los números por cada dígito del otro, corriendo una posición a la izquierda después de cada multiplicación, y sumando los resultados al final. Por ejemplo, en base 4:

$$\begin{array}{r} 3202 \\ * 231 \\ \hline 3202 \\ 22212 \\ 13010 \\ \hline 2132322 \end{array}$$

El caso binario es más simple, ya que los dígitos son 1 ó 0:

$$\begin{array}{r} 1101 \\ * 101 \\ \hline 1101 \\ 0000 \\ 1101 \\ \hline 1000001 \end{array}$$

En términos de polinomios, está ocurriendo lo siguiente:

$$X * Y = \sum_{i=0}^n x_i * b^i * \sum_{j=0}^n y_j * b^j = \sum_{j=0}^n \left(\sum_{i=0}^n x_i * b^i * \right) y_j * b^j$$

La última ecuación nos dice que el número X se debe multiplicar por cada uno de los dígitos de Y , y los productos parciales se suman, pero como están multiplicados por b^j , esto explica el “corrimiento a la izquierda” de los productos parciales.

La división

El método de la división es como en el caso decimal. Se separan dígitos del dividendo hasta que el divisor sea menor que los dígitos separados. Se mira cuántas veces cabe el divisor en el número separado. Se multiplica el divisor por esta

cantidad y el resultado de la multiplicación se le resta al número separados. Se agrega el dígito siguiente del dividendo al resultado de la resta y se continúa aplicando el método. A continuación puede ver un ejemplo base 3:

$$\begin{array}{r}
 10221 \overline{) 12} \\
 \underline{-101} 210 \\
 12 \\
 \underline{-12} \\
 001 \\
 \underline{-000} \\
 001
 \end{array}$$

De nuevo, la división binaria es más sencilla puesto que un número en otro cabe o no cabe, 1 ó 0, no hay que analizar más casos:

$$\begin{array}{r}
 101010 \overline{) 11} \\
 \underline{-11} 1110 \\
 100 \\
 \underline{-11} \\
 11 \\
 \underline{-11} \\
 00 \\
 \underline{-0} \\
 0
 \end{array}$$

Como se puede ver, los algoritmos en otras bases, en el fondo, son iguales a los algoritmos en base 10. Solo que las divisiones y módulos se hacen por la base en cuestión, la cual, evidentemente, cambia.

2.5 OPERACIONES LÓGICAS

Estrictamente hablando, no se trata aquí de operaciones lógicas sino de una extensión de estas operaciones para trabajar sobre números binarios.

La idea es la siguiente: vamos a interpretar los números 1 y 0 como los valores de verdad "verdadero" y "falso", respectivamente; de esta manera podemos extender las operaciones lógicas (Y, O, O-excluyente, NO) a los números binarios. Por ejemplo, veamos cómo queda la tabla de verdad de la operación Y (&):

x	y	x & y
1	1	1
1	0	0
0	1	0
0	0	0

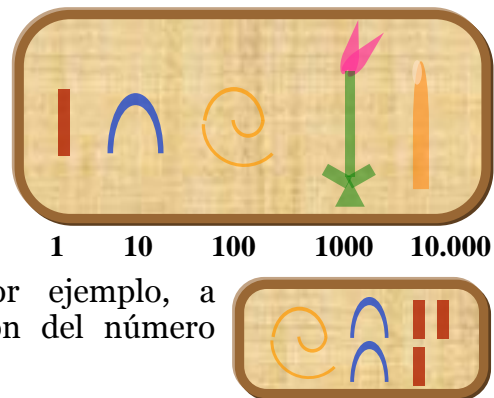
Además, podemos hacer una generalización de estas operaciones a dos números binarios cualesquiera; para lo cual realizamos la operación dígito a dígito. Por ejemplo:

$$\begin{array}{r}
 10111011 \\
 \& 11011100 \\
 \hline
 10011000
 \end{array}$$

Lo anteriormente es aplicable a cualquier operador lógico, pero, en la práctica, sólo se suele utilizar la negación y los tres operadores O, Y y O-excluyente. Más adelante volveremos a encontrar estas operaciones.

EJERCICIOS

- 1- El sistema de numeración de los egipcios no era posicional, pero sí funcionaba por agrupación. Se disponía de símbolos para representar las potencias de 10: la vara, el talón, la cuerda, la flor de loto, el dedo, etc. (ver la figura). Cada símbolo se replicaba entre 0 y 9 veces para representar los diferentes valores; por ejemplo, a continuación encuentra la representación del número 123.



a- Represente los números 23.561 y 16.643 en numeración egipcia.

b- Desarrolle un algoritmo para sumar en notación egipcia; pruébelo con los dos números anteriores.

- 2- En el sistema imperial de unidades, una milla tiene 1760 yardas, una yarda tiene 3 pies y un pie 12 pulgadas. Completar la siguiente tabla pasando de unidades del sistema imperial a pulgadas o viceversa:

mi, yd, ft, in	inches
2, 134, 2, 11	
3, 0, 1, 7	
	131555
	190099

- 3- Completar la siguiente tabla pasando de horas, minutos y segundos a segundos o viceversa:

hh:mm:ss	Segundos
(3:54'12")	
(4:0'37")	
	8621
	10825

- 4- Complete la siguiente tabla, convirtiendo cada número a las otras bases. Use los métodos que desee para cambiar de base.

Base	7	4	16
	(236) ₇		
		(321) ₄	
			(3B) ₁₆

- 5- Es posible resolver los siguientes ejercicios mentalmente y en poco tiempo.

- a-** Diga cuántos unos tiene la representación binaria de: $2^{(a + b)}$, donde $a = 43 * b$, $a > 300$ y $b \leq 10$.
- b-** Al pasar el número 1898 a la base 33 ¿El resultado es divisible por 10? ¿Es primo?
- c-** ¿El número $(7A9JG23K3)_{30}$ es primo?
- d-** Se tiene un número racional, en alguna base, con fracción infinita. Al cambiarlo de base ¿La fracción se puede volver finita? Si en la nueva base la fracción es infinita, ¿Puede no ser periódica?
- 6-** Manejo de fracciones.
- a-** Escriba la fracción $(.542)_{10}$ en base 2.
- b-** Escriba la fracción $(.\bar{3})_{10}$ en base 3.
- c-** Escriba la fracción $(.0101)_2$ en base 10.
- d-** Repita la parte c- pero usando el método de escribir el número binario debajo de las potencias de dos (en el texto lo vimos únicamente para enteros, extiéndalo al caso de las fracciones).
- e-** Tiene la fracción $(.0\bar{3})_4$. Escríbala en base cuatro pero usando sólo una posición decimal y sin perder precisión.
- 7-** Para convertir una fracción de una base a otra, se puede fijar una precisión y detener el proceso de conversión cuando se llegue a ella. Para determinar la precisión, se parte de la siguiente idea: si se tiene una fracción con i dígitos fraccionarios en la base A, la precisión es $(0.1)_A^i$. Si se quiere pasar a la base B, conservando la precisión, hay que buscar un j tal que: $(0.1)_A^i = (0.1)_B^j$. Despeje j en términos de i , A y B (el valor de j define cuántos dígitos hay que calcular en la base B).
- 8-** Un maya está negociando con un español. El maya usa la base 20, y no le es fácil captar los números en decimal; el español usa la base 10 y se le dificulta entender los números en base 20.
- a-** El español da los precios en decimal, pero el sistema decimal no es "natural" para el maya; ¿qué algoritmo usa para saber el valor del número?
- b-** El maya, amablemente, da los precios en decimal. Dado que solo conoce los precios en la base 20 ¿cómo hace para obtener el número decimal correspondiente?
- c-** Si el maya diera los precios en base 20 ¿Qué debería hacer el español para entender el número? Compare la respuesta con la de la parte a-.
- 9-** Justifique matemáticamente el "método rápido" para pasar de base 2 a base 2^k . ¿Se puede generalizar a b y b^k ?
- 10-** Hacer las siguientes operaciones con horas, minutos y segundos:
- a-** $(4:35'15'') + (3:51'35'') + (2:44'20'')$.

- b-** $(3:51'35'') * 2$.
- c-** $(3:51'35'') / 2$.
- 11-** Haga las siguientes operaciones en unidades del sistema imperial:
- a-** (4 millas, 550 yardas, 1 pies, 5 pulgadas) + (3 millas, 1209 yardas, 2 pies, 7 pulgadas).
- b-** (3 millas, 1209 yardas, 2 pies, 7 pulgadas) * 2.
- c-** (3 millas, 1208 yardas, 2 pies, 7 pulgadas) / 3.
- 12-** Multiplicación por la base.
- a-** Calcule en binario: $(101100)_2 * (1000)_2$.
- b-** En general, se tiene el número $(d_n \dots d_1)_b$ en base b y se multiplica por b^k . Describa (en términos de los d_j) cómo debe ser el resultado.
- 13-** División por la base.
- a-** Calcule en binario (división entera): $(101011)_2 / (1000)_2$.
- b-** Calcule en binario: $(101011)_2$ módulo $(1000)_2$.
- c-** En general, se tiene el número $(d_n \dots d_1)_b$ en base b y se divide por b^k , $k < n$. Describa (en términos de los d_j) cómo deben ser el cociente y el residuo de esta división.
- 14-** Se hizo la siguiente operación: $(11)_B * (11)_B = (121)_B$. ¿Es posible determinar cuál es la base B?
- 15-** Realice las siguientes sumas directamente en la base en cuestión.
- a-** $(101100)_2 + (1101101)_2$.
- b-** $(1E9)_{16} + (382)_{16}$.
- c-** $(1101101)_2 + (337)_8$ (dar el resultado en binario)
- 16-** Se tiene el número $(3B6)_{16}$.
- a-** Calcule el polinomio correspondiente $(3*16^2 + 11*16 + 6)$ en binario. Es decir, en el polinomio, reemplace cada número por su valor en binario y efectúe los cálculos en binario. ¿Cuál debe ser el resultado?
- b-** Repita el ejercicio de la parte a- pero esta vez haciendo los cálculos en base 3. ¿Cuál debe ser el resultado?
- c-** Tome el número base 3 de la parte b-, y calcule su polinomio en binario. ¿Cuál debe ser el resultado?
- 17-** La operación O-negado (NOR) se define por:
- $$x \text{ O-negado } y = \overline{NO(x \text{ O } y)} = \overline{x \vee y}$$
- Defina las operaciones Y, O, NO usando únicamente la operación O-negado.

- 18-** Vamos a notar el O-excluyente con el operador \oplus . Siendo x una variable de un bit, se definen las operaciones f , g y h como:

$$\begin{aligned}f(x) &= x \oplus 1 \\g(x) &= x \oplus 0 \\h(x) &= x \oplus x\end{aligned}$$

¿A qué funciones "estándar" equivalen f , g y h ?

- 19-** Siendo x , y y z variables de un bit:

a- ¿Es $(x \oplus y) \oplus z = x \oplus (y \oplus z)$?

b- ¿A qué es igual $(x \oplus y) \oplus y$?

- 20-** Siendo x y y variables de un bit, se definen las operaciones f y g como:

$$\begin{aligned}f(x, y) &= (x + y) \text{ módulo } 2 \\g(x, y) &= (x - y) \text{ módulo } 2\end{aligned}$$

¿A qué funciones "estándar" equivalen f y g ?

- 21-** Se tiene un número binario de k dígitos.

a- Se quiere que el i -ésimo dígito se convierta en 1 sin importar qué valor tenía antes y sin modificar los demás dígitos. ¿Cuál operación lógica se podría usar para lograrlo?

b- ¿Qué operación lógica habría que efectuar para que el i -ésimo dígito se convierta en cero?

- 22-** Se tiene un número binario de k dígitos, y se quiere obtener el módulo 2^n . Hágallo utilizando una operación lógica.

- 23-** Discuta esta afirmación: algunos pretenden que la gran pirámide (pirámide de Keops) tiene un simbolismo oculto. Por ejemplo, su altura es de 150m y la distancia de la tierra al sol es de 150 millones de Km. Si esto es cierto, implicaría que los egipcios conocían la numeración posicional en base 10.

Es posible generalizar los sistemas de numeración posicional modificando algunas de las reglas usuales, tales como que los dígitos deben ser enteros o que deben ser positivos. En general, podemos considerar que un sistema de numeración posicional está compuesto de dos sucesiones: una sucesión d_i de dígitos, y una

sucesión base B_i . El valor del número representado es: $\sum_{i=0}^n d_i * B_i$. En los sistemas de

numeración usuales, $B_i = b^i$, y $d_i \in \{ x \mid 0 \leq x < b \}$; a continuación exploraremos otras posibilidades.

- 24-** Numeración de Fibonacci. En este caso vamos a considerar que $d_i \in \{ 0, 1 \}$, y $B_i = f_i$, donde f_i es la serie de Fibonacci empezando en $f_0 = 1$, y $f_1 = 2$; por ejemplo, los números del 0 al 5 serían: 0, 1, 10, 100, 101, 1000.

a- Desarrolle un algoritmo para codificar un número cualquiera en la numeración de Fibonacci.

b- Note que en esta notación es posible expresar el mismo número de varias maneras, por ejemplo, $11 = 100$. Por esto, vamos a definir una forma normal: un número en esta notación no puede tener dos unos seguidos. Desarrolle un algoritmo para convertir un número cualquiera en notación de Fibonacci a la forma normal. Use la definición de los números de Fibonacci: $f_{n+2} = f_{n+1} + f_n$.

- 25-** Base 3 simétrica. En este caso $d_i \in \{-1, 0, 1\}$, y $B_i = 3^i$; por ejemplo, los números del 0 al 5 serían: $0, 1, \bar{1}\bar{1}, 10, 11, \bar{1}\bar{1}\bar{1}$ (el -1 se nota por $\bar{1}$)

a- Desarrolle un algoritmo para codificar un número cualquiera en la base 3 simétrica.

b- En la base 3 simétrica los números negativos son connaturales a la notación. Desarrolle un algoritmo para codificar un número negativo cualquiera en base 3 simétrica.

c- Desarrolle un algoritmo para sumar dos números en esta representación.