Sistemas numéricos

Matemáticas Discretas

19/08/2013

Instituto Tecnológico de Nuevo Laredo
I.S.C. José Antonio Espino Lopez, M.A.N.

Contenido

Introducción

De acuerdo con la historia se cree que los primeros pobladores utilizaban rayas, círculos, figuras de animales u objetos para representar cantidades. Por ejemplo, una manada de siete animales podría estar representada por siete rayas o siete figuras de ese animal, pero para representar cantidades cada vez mayores se usó la agrupación de varios símbolos en uno solo, con la finalidad de compactar la información. Por ejemplo, los egipcios utilizaban símbolos para representar cantidades y algunos de ellos son | = 1, ∩ = 10, ? = 100; utilizando éstos, la representación de 134 es la siguiente:

?∩∩∩|||| = 134

Un sistema corno el anterior se conoce como *sistema aditivo* y en él se suman los valores de todos los símbolos para obtener la cantidad total, sin embargo este sistema es impráctico para la representación de cantidades grandes o muy pequeñas, ya que se necesitarían muchos símbolos para su representación.

Otro sistema aditivo es el sistema de numeración romano en el cual los símbolos I, V, X, L. C, D y M representan cantidades y una línea sobre el símbolo implica una multiplicación del número por mil. En ambos casos se suman los valores de los caracteres de acuerdo con sus propias reglas, pero en éstas no importa la posición sino únicamente el símbolo y es por eso que se les llama sistemas de *numeración aditivos*.

Se cree que los babilonios fueron uno de los primeros pueblos en usar un sistema posicional para la representación de cantidades, ya que con base en el movimiento de los astros usaban un sistema sexagesimal (60 caracteres diferentes, en donde cada uno representa un número) para indicar cantidades. Su sistema aún se utiliza para la medición de horas, minutos y segundos, sin embargo tiene problemas con la representación del cero.

Actualmente los sistemas para la representación de cantidades son posicionales, ya que éstos tienen muchas ventajas en relación con los aditivos Ejemplos de sistemas posicionales son los sistemas numéricos decimal, binario, octal y hexadecimal. Una característica de los sistemas posicionales es que el valor del símbolo lo determina la posición que ocupa y la base del sistema, que es la cantidad de símbolos diferentes usados en un sistema numérico.

En esta unidad se trata preferentemente la representación, conversión y operaciones aritméticas en los sistemas decimal, binario, octal y hexadecimal, pero es importante mencionar que el procedimiento de representación, conversión y operación es el mismo, independientemente del sistema numérico de que se trate.

Términos

Sistema Numérico

Se llama sistema numérico al conjunto ordenado de símbolos o dígitos y a las reglas con que se combinan para representar cantidades numéricas. Existen diferentes sistemas numéricos, cada uno de ellos se identifica por su base.

Dígito

Un dígito en un sistema numérico es un símbolo que no es combinación de otros y que representa un entero positivo.

Bit
Es un dígito binario (Abreviación del inglés binary digit) es decir, un 0 o un 1.

Base de un sistema numérico

La base de un sistema numérico es el número de dígitos diferentes usados en ese sistema.

A continuación se ejemplifican estas definiciones con los sistemas numéricos más comúnmente usados que son:

Base	Sistema	Dígitos
2	Binario	0, 1
8	Octal	0, 1, 2, 3, 4, 5, 6, 7
10	Digital	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
16	Hexadecimal	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Notación

En adelante, para distinguir entre los diferentes sistemas numéricos encerraremos entre paréntesis el número y le añadiremos un subíndice, indicando la base que se está usando.

Sin embargo, si no se usa subíndice se deberá entender que el número está en base diez, a menos que se diga lo contrario.

Ejemplos:
$35 = (35)_{10}$ = 35 base 10 (sistema decimal)
$(110100)_2$ = 110100 base 2 (sistema binario)
$(34)_{16}$ = 34H = 34 base 16 (sistema hexadecimal)

En general cualquier número entero consta de

Parte entera . Parte Fraccionaria

Ejemplos:

10 . 50
1526 . 36
1478 . 3695
586

1.1 Sistemas numéricos (Decimal, Binario, Octal, Hexadecimal)

Sistema decimal

El sistema decimal se usa en forma rutinario para la representación de cantidades mediante los siguientes 10 caracteres diferentes:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Con estas cifras se pueden expresar cantidades hasta el 9. Para expresar cantidades más allá de este número es necesario introducir la representación posicional, es decir, a cada cifra se le asigna un valor posicional determinado de acuerdo con el lugar que ocupa dentro del número. Por ejemplo: el número decimal 836.74 se compone en la parte entera de la cifra 8 con el valor posicional 100, la cifra 3 con el valor posicional 10 y la cifra 6 con el valor posicional 1, y en la parte fraccionaria de la cifra 7 con el valor posicional 0.1 y la cifra 4 con el valor posicional 0.01. Así se tiene que:

$$836.74 = 8 \times 100 + 3 \times 10 + 6 \times 1 + \frac{7}{10} + \frac{4}{100}$$

Usando exponentes esto se puede representar como:

$$836.74 = 8 \times 10^2 + 3 \times 10^1 + 6 \times 10^0 + 7 \times 10^{-1} + 4 \times 10^{-2}$$

A esta forma de representación se le llama representación exponencial.

La representación exponencial es especialmente importante porque por medio de ella se puede convertir una cantidad representada en cualquier sistema numérico al sistema decimal, como se estudiará más adelante. El valor de la posición lo determina el exponente en una sucesión ascendente de derecha a izquierda para los enteros a partir del punto decimal (el 6 se encuentra en la posición 0, el 3 en la posición 1 y el 8 en la posición 2), y de izquierda a derecha para la parte fraccionaria (el 7 está en la posición -1 y el 4 en la posición -2), usando como base el número 10, debido a que se está en el sistema decimal. También se dice que la base de este sistema aritmético es 10, tomando en cuenta los 10 símbolos disponibles para representar cantidades.

Sistema binario

En el sistema binario sólo hay dos cifras: 0 y 1. Como sucede en el sistema decimal, en este sistema binario también se utilizan exponentes para expresar cantidades mayores. Mientras que en el sistema decimal la base es 10, en el sistema binario la base es 2.

Sistema Decimal

Desde el punto de vista matemático, el sistema decimal no posee ninguna ventaja especial sobre cualesquier otro posible sistema de numeración y su uso generalizado se debe a razones completamente ajenas a las leyes generales de las matemáticas.

De acuerdo con la antropología, el origen del sistema decimal se encuentra en el hecho de que los seres humanos tenemos diez dedos en las manos.

Sistema Binario

El antiguo matemático indio Píngala presentó la primera descripción que se conoce de un sistema de numeración binario en el siglo III a. de C., lo cual coincidió con su descubrimiento del concepto del número cero.

El sistema binario moderno fue documentado en su totalidad por Leibniz, en el siglo XVII, en su artículo *"Explication the Arithmétique Binaire".* Leibniz usó el 0 y el 1, al igual que el sistema de numeración binario actual.

Como se mencionó anteriormente, la representación exponencial se utiliza para convertir una cantidad de un sistema numérico cualquiera al sistema decimal. A continuación se muestra la forma de hacer esto.

Ejemplo. Convertir el numero binario 10011.01 a decimales
Solución. Expresando el número propuesto en notación exponencial y realizando las operaciones correspondientes, se obtiene la siguiente conversión de binario a decimal:

$$10011.01_{(2)} = 1\times 2^4 + 0\times 2^2 + 1\times 2^1 + 1\times 2^0 + 0\times 2^{-1} + 1\times 2^{-2}$$
$$10011.01_{(2)} = 16 + 0 + 0 + 2 + 1 + 0 + 0.25 = 19.25_{(10)}$$

Sistema octal

La reglas descritas para los sistemas decimal y binario, también son aplicables al sistema octal. En los siguientes ejemplos se ilustra este planteamiento.

Ejemplo. Convertir $631.532_{(8)}$ a decimal.
Solución. Primero se convierte el número dado a decimal y luego de decimal a binario.

Para convertir una cantidad de cualquier sistema numérico a decimal, se plantea su representación en notación exponencial y se realizan las operaciones. En este caso particular se tiene que:

$$631.532_{(8)} = 6\times 8^2 + 3\times 8^1 + 1\times 8^0 + 5\times 8^{-1} + 3\times 8^{-2} + 2\times 8^{-3}$$
$$= 409.6758_{(10)}$$

Sistema hexadecimal

La base numérica del sistema hexadecimal es 16 y para representar cantidades en él se utilizan los diez dígitos del sistema decimal (O, 1, 2, 3, 4,5, 6, 7, 8, 9) así como las seis primeras letras del alfabeto (A, B, C, D, E, F).

Con esto pueden formarse números según el principio de valor posicional como en los demás sistemas aritméticos. Los caracteres válidos en hexadecimal son del 1 al 15, con la particularidad de que a las letras se les asigna el siguiente valor: A= 10, B = 11, C = 12, D = 13, E= 14 y F = 15.

Ejemplo. Convertir $E8A7.3D_{(16)}$ a decimal.
Solución. El número dado primero se convierte a decimal:

$$E8A7.3D_{(16)} = 14\times 16^3 + 8\times 16^2 + 10\times 16^1 + 7\times 16^0 + 3\times 16^{-1}$$
$$+ 13\times 16^{-2} = 59559.2383_{(10)}$$

Sistema hexadecimal

El uso del sistema hexadecimal está estrechamente relacionado con la informática y con las ciencias de la computación, ya que las computadoras suelen utilizar el byte u octeto como unidad básica de memoria.

1.2 Conversiones entre sistemas numéricos.

Sistema decimal

Usando exponentes esto se puede representar como:

$$836.74 = 8 \times 10^2 + 3 \times 10^1 + 6 \times 10^0 + 7 \times 10^{-1} + 4 \times 10^{-2}$$

Sistema binario

Ejemplo. Convertir el numero binario 10011.01 a decimales

Solución. Expresando el número propuesto en notación exponencial y realizando las operaciones correspondientes, se obtiene la siguiente conversión de binario a decimal:

$$10011.01_{(2)} = 1 \times 2^4 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$$
$$10011.01_{(2)} = 16 + 0 + 0 + 2 + 1 + 0 + 0.25 = 19.25_{(10)}$$

Ejemplo. Convertir el número $28.37_{(10)}$ a binario.

Solución. Parte entera:

	Resto	
28 / 2 = 14	0	Los restos se toman en orden inverso a como fueron encontrados.
14 / 2 = 7	0	
7 / 2 = 3	1	
3 / 2 = 1	1	
1 / 2 = 0	1	

Parte fraccionaria:

	Entero	
0.37 × 2 = 0.74	0	Los enteros se toman en el mismo orden en que fueron encontrados.
0.74 × 2 = 1.48	1	
0.48 × 2 = 0.96	0	
0.96 × 2 = 1.92	1	
0.92 × 2 = 1.84		

Se podría seguir aproximando para determinar más dígitos en la parte fraccionaria y obtener así un resultado más exacto, sin embargo para ilustrar el procedimiento es suficiente con cuatro dígitos después del punto que separa a la parte entera de la parte fraccionaria. De esta forma, el resultado es:

$$28.37_{(10)} = 11100.0101_{(2)}$$

Sistema octal

Ejemplo. Convertir $631.532_{(8)}$ a binario.

Solución. Primero se convierte el número dado a decimal y luego de decimal a binario.

Para convertir una cantidad de cualquier sistema numérico a decimal, se plantea su representación en notación exponencial y se realizan las operaciones. En este caso particular se tiene que:

$$631.532_{(8)} = 6 \times 8^2 + 3 \times 8^1 + 1 \times 8^0 + 5 \times 8^{-1} + 3 \times 8^{-2} + 2 \times 8^{-3} = 409.6758_{(10)}$$

La conversión del número obtenido a binario es la siguiente:

Parte entera	*Resto*	Parte fraccionaria	*Entero*
409/2 = 204	1		
204/2 = 102	0	$0.6758 \times 2 = 1.3516$	1
102/2 = 51	0	$0.3516 \times 2 = 0.7032$	0
51/2 = 25	1	$0.7032 \times 2 = 1.4064$	1
25/2 = 12	1	$0.4064 \times 2 = 0.8128$	0
12/2 = 6	0		
6/2 = 3	0		
3/2 = 1	1		
1/2 = 0	1		

La conversión de octal a binario y de binario a octal es relativamente fácil si se utiliza la siguiente tabla de equivalencias*:

Octal	Binario
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Nota:

* SE PUEDE APRECIAR QUE SE UTILIZAN TRES DÍGITOS EN BINARIO, PARA CADA NÚMERO EN OCTAL, DEBIDO A QUE LA CANTIDAD MAYOR VÁLIDA EN EL SISTEMA OCTAL ES EL NÚMERO 7, QUE OCUPA TRES BITS. POR LO TANTO, TODOS DEBERÁN USAR LA MISMA CANTIDAD DE BITS.

Ejemplo. Convertir $631.532_{(8)}$ a binario usando la tabla de equivalencias anterior.

Solución. En la siguiente tabla se presenta la conversión pedida:

$$\begin{array}{cccccccc} \mathbf{6} & \mathbf{3} & \mathbf{1} & \mathbf{.} & \mathbf{5} & \mathbf{3} & \mathbf{2}_{(8)} \\ \mathbf{110} & \mathbf{011} & \mathbf{001} & \mathbf{.} & \mathbf{101} & \mathbf{011} & \mathbf{010}_{(2)} \end{array}$$

Ejemplo. Convertir 11010100000111101011010.0001101$_{(2)}$ a octal usando tablas y verificar dicho resultado usando el método general.

Solución. Cuando se usan tablas, se deben separar los bits de la cantidad binaria en bloques de tres en tres, a partir del punto decimal hacia la izquierda en la parte entera, y del punto decimal a la derecha en la parte fraccionaria. Si los bloques no se completan, se agregan ceros en los extremos como se indica a continuación:

011	010	100	000	111	101	011	010	.	000	110	1**00**$_{(2)}$
3	2	4	0	7	5	3	2	.	0	6	4$_{(8)}$

Para verificar este resultado usando el método general, primero se convierte de binario a decimal de forma que:

$$11010100000111101011010.0001101_{(2)} = 1 \times 2^{22} + 1 \times 2^{21} + 1 \times 2^{19} + 1 \times 2^{17} + 1 \times 2^{11} + 1 \times 2^{10} + 1 \times 2^{9} + 1 \times 2^{8} + 1 \times 2^{6} + 1 \times 2^{4} + 1 \times 2^{3} + 1 \times 2^{1} + 1 \times 2^{-4} + 1 \times 2^{-5} \quad 1 \times 2^{-7} = 4194304 + 2097152 + 524288 + 131072 + 2048 + 1024 + 512 + 256 + 64 + 16 + 8 + 2 + 0.0625 + 0.0312 + 0.0078 = 6950746.1015_{(10)}$$

A continuación se convierte de decimal a octal:

Parte entera	*Resto*	Parte fraccionaria	*Entero*
6950746/8 = 868843	2		
868843/8 = 108605	3	0.1015 × 8 = 0.8120	0
108605/8 = 13575	5	0.8120 × 8 = 0.4960	6
13575/8 = 1696	7	0.4960 × 8 = 0.9680	3
1696/8 = 212	0	0.9680 × 8 = 0.7440	7
212/8 = 26	4		
26/8 = 3	2		
3/8 = 0	3		

Al comparar los resultados obtenidos por el método general y mediante la tabla de equivalencias, se observa que la parte entera coincide totalmente y que solamente existe una pequeña diferencia en la parte fraccionaria, debido a errores de redondeo.

Es importante tener cuidado cuando se encuentra el resto de una división. En el caso de este ejemplo se dividió 108605/8 = 13575.625, y el resto resultó de multiplicar la parte fraccionaria del cociente por la base, esto es, 0.625 X 8 = 5.

Sistema hexadecimal

Ejemplo. Convertir $E8A7.3D_{(16)}$ a octal.
Solución. El número dado primero se convierte a decimal:

$$\begin{aligned} E8A7.3D_{(16)} &= 14 \times 16^3 + 8 \times 16^2 + 10 \times 16^1 + 7 \times 16^0 + 3 \times 16^{-1} + 13 \times 16^{-2} \\ &= 59559.2383_{(10)} \end{aligned}$$

Ahora el número obtenido se convierte a octal:

Parte entera	*Resto*	Parte fraccionaria	*Entero*
59559/8 = 7444	7 ↑		
7444/8 = 930	4	0.2383 × 8 = 1.9064	1 ↓
930/8 = 116	2	0.9064 × 8 = 7.2512	7
116/8 = 14	4	0.2512 × 8 = 2.0096	2
14/8 = 1	6	0.0096 × 8 = 0.0768	0
1/8 = 0	1		

De igual manera que en la conversión de binario a octal, se puede obtener la siguiente tabla de equivalencias de binario a hexadecimal*:

Hexadecimal	Binario
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Nota:

* NUEVAMENTE EL NUMERO MAYOR DEL SISTEMA NUMÉRICO ES EL QUE MANDA EN RELACIÓN CON CUÁNTOS BITS SE DEBERÁN USAR PARA REPRESENTAR CADA UNO DE LOS CARACTERES. EN ESTE CASO F = 15 ES EL SÍMBOLO MAYOR Y OCUPA CUATRO BITS. POR LO TANTO TODOS LOS SÍMBOLOS DEBERÁN REPRESENTARSE POR CUATRO

Ejemplo. Convertir $E8A7.3D_{(16)}$ a octal usando tablas de equivalencias.
Solución. A diferencia del método general, en el que el sistema intermedio es el sistema decimal, cuando se utilizan tablas de equivalencias el sistema intermedio es el binario, por lo que primero se pasa la cantidad al sistema binario poniendo los cuatro bits correspondientes a cada uno de los caracteres del sistema hexadecimal:

$$\begin{array}{cccccccc} \mathbf{E} & \mathbf{8} & \mathbf{A} & \mathbf{7} & . & \mathbf{3} & \mathbf{D}_{(16)} \\ \mathbf{1110} & \mathbf{1000} & \mathbf{1010} & \mathbf{0111} & . & \mathbf{0011} & \mathbf{1101}_{(2)} \end{array}$$

1.3 Operaciones básicas (Suma, Resta, Multiplicación, División)

Las operaciones básicas de suma, resta, multiplicación y división que se realizan en el sistema decimal, también se pueden llevar a cabo en cualquier sistema numérico aplicando las mismas reglas y teniendo en cuenta la base en la que se encuentran los números con los que se efectúa la operación. Es importante observar que las cantidades que se estén operando se deben de encontrar en la misma base, y en caso de no ser así lo primero que se debe de hacer es la conversión correspondiente de cada una de ellas.

A continuación se realizarán operaciones básicas de suma, resta, multiplicación y división en los sistemas decimal, binario, octal y hexadecimal. El sistema decimal permite ilustrar el procedimiento a seguir en cada una de las operaciones aritméticas, gracias a la familiaridad que se tiene con él, y los sistemas binario, octal y hexadecimal son de gran utilidad en el área de la computación.

Suma

Ejemplo. Suma en el sistema decimal:

$$\begin{array}{rr} & 456.78_{(10)} \\ + & 17820.649_{(10)} \\ \hline & 18277.429_{(10)} \end{array}$$

Explicación por columnas:

0+9=9	El 9 es un dígito válido de base 10, por lo que se queda tal cual.
8 + 4 = 12	El 12 no es válido en decimal, ya que es una combinación del 1 y el 2. Cuando ocurre esto se deberá dividir entre la base (10), colocando el resto debajo de la línea y sumando el cociente a los números de la siguiente columna de la izquierda
1+7+6 = 14	El 14 no es válido por lo que se divide entre la base y se procede como se hizo anteriormente.
1+6+0=7	Dígito válido en decimal.
5+2=7	Dígito válido.
4 + 8 = 12	Aquí hay que dividir entre la base.
1+0+7=8	Dígito válido.

Los espacios vacíos de los extremos, como el de arriba del 9 en el ejemplo anterior, se consideran como 0, por esta razón se sumó 0 + 9. Esto mismo sucede en todos los sistemas numéricos, ya

Suma, resta y multiplicación

La suma, la resta y la multiplicación de números son ejemplos de operaciones binarias, esto es, operaciones entre pares de números.

En general una operación binaria definida en un conjunto, es una regla que asocia a cada par ordenado de elementos del conjunto algún elemento del mismo conjunto.

que el 0 es el dígito válido más pequeño. Cuando el resultado de la suma es un símbolo válido en ese sistema numérico normalmente no se divide entre la base, pero podría dividirse ya que el cociente es 0 y por lo tanto no afecta a la siguiente columna de la izquierda y el resto es el mismo número, por ejemplo, si el resultado es 6, el cual es válido en el sistema decimal, el cociente de dividir 6 entre 10 es 0 y el resto es 6.

Ejemplo. Suma en el sistema hexadecimal:

$$\begin{array}{rcccccccc} & A & 6 & F & C & 9 & . & 7 & B & 2_{(16)} \\ + & 4 & E & 7 & D & 0 & . & 7 & 3 & E_{(16)} \\ \hline & F & 5 & 7 & 9 & 9 & . & E & F & 0_{(16)} \end{array}$$

Explicación por columna:

2 + 14 = 16	Al dividir 16 entre la base se obtiene el cociente 1 y resto 0.
1 + 11 + 3 = 15	Dígito válido: 15 = F.
7 + 7 = 14	Dígito válido: 14 = F.
9 + 0 = 9	Dígito válido.
12 + 13 = 25	Al dividir 25 entre la base se obtiene el cociente 1 y resto 9.
1 + 15 + 7 = 23	Al dividir 23 entre la base se obtiene el cociente 1 y resto 7.
1 + 6 + 14 = 21	Al dividir 21 entre la base se obtiene el cociente 1 y resto 5.
1 + 10 + 4 = 15	Dígito válido: 15 = F

Suma de números binarios

La tabla de sumar para números binarios es la siguiente:

+	0	1
0	0	1
1	1	10

Las posibles combinaciones al sumar dos bits son:

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 10

Note que al sumar 1 + 1 es 102, es decir, llevamos 1 a la siguiente posición de la izquierda (acarreo). Esto es equivalente, en el sistema decimal a sumar 9 + 1, que da 10: cero en la posición que estamos sumando y un 1 de acarreo a la siguiente posición.

Ejemplo

Acarreo	1
	1 0 0 1 1 0 0 0
	+0 0 0 1 0 1 0 1
Resultado	**1 0 1 0 1 1 0 1**

Se puede convertir la operación binaria en una operación decimal, resolver la decimal, y después transformar el resultado en un (número) binario. Operamos como en el sistema decimal: comenzamos a sumar desde la derecha, en nuestro ejemplo, 1 + 1 = 10, entonces escribimos 0 en la fila del resultado y llevamos 1 (este "1" se llama acarreo o arrastre). A continuación se suma el acarreo a la siguiente columna: 1 + 0 + 0 = 1, y seguimos hasta terminar todas la columnas (exactamente como en decimal).

Resta

Ejemplo. Resta en el sistema decimal:

$$\begin{array}{r} 8\;1\;2\;7\;.\;5\;8\;0_{(10)} \\ 5\;8\;3\;1\;.\;9\;6\;4_{(10)} \\ \hline 2\;2\;9\;5\;.\;6\;1\;6_{(10)} \end{array}$$

Explicación por columna:

(0 + 10) - 4 = 6	Cuando el sustraendo es mayor que el minuendo, como ocurre en la primera columna, se deberá sumar la base al minuendo y después llevar a cabo la sustracción (minuendo + 10) - sustraendo = resultado.
8 - (6 1- 1)=1	Cuando en la columna anterior se sumó 10 a 1 minuendo, en la columna siguiente de la izquierda se deberá sumar 1 al sustraendo (minuendo - (sustraendo + 1)) =resultado. Si después de sumar 1 al sustraendo éste es mayor que el minuendo, entonces se deberá sumar la base al minuendo antes de llevar a cabo la resta.
(5 + 10) - 9 = 6	Debido a que sustraendo > minuendo, se suma la base al minuendo y se realiza la resta.
7 - (1 + 1) = 5	Como en la columna anterior se le sumó la base 10 al minuendo, en esta columna se le deberá sumar 1 al sustraendo.
(2 + 10) - 3 = 9	Como sustraendo > minuendo, se suma la base al minuendo.
(1 + 10) - (8 + 1) = 2	Como se le sumó la base a la columna anterior, se deberá aumentar en 1 el sustraendo y como sustraendo > minuendo, se deberá sumar la base al minuendo antes de llevar a cabo la resta. El orden en que se llevan a cabo los incrementos en este caso es muy importante.
8 - (5 + 1) = 2	Sumar 1 al sustraendo, ya que en la columna anterior se le sumó la base al minuendo, y después llevar a cabo la resta.

Al efectuar la resta es necesario revisar si el sustraendo es mayor que el minuendo, ya que en caso afirmativo se debe sumar la base al minuendo antes de llevar a cabo la resta de dos dígitos de una columna cualquiera. Una vez comenzada la operación de resta cuando al minuendo se le suma la base, entonces al sustraendo de la columna izquierda próxima se le deberá sumar 1 (ya que en este caso la base es 10) antes de hacer la comparación entre el minuendo y sustraendo. En el caso de otro sistema numérico, lo que se le suma al minuendo debe de ser la base que corresponda (8 en octal, 16 hexadecimal o 2 en binario), sin embargo cuando se le suma la base al minuendo invariablemente será 1 lo que se incrementa el sustraendo de la columna izquierda próxima, independientemente del sistema numérico de que se trate (ya que ese 1 significa que se le sumó una vez la base en la columna anterior).

Ejemplo. Resta en el sistema octal:

$$\begin{array}{cccccccc}
4 & 1 & 0 & 7 & 2 & . & 1 & 4_{(8)} & \\
3 & 6 & 0 & 4 & 3 & . & 7 & 1 & 3_{(8)} \\
\hline
0 & 3 & 0 & 2 & 6 & . & 2 & 2 & 5_{(8)}
\end{array}$$

Explicación por columna:

(0+ 8) - 3 = 5	Como sustraendo > minuendo, hay que sumar la base al minuendo.
4 - (1 + 1) = 2	Sumar 1 al sustraendo debido a que se sumó la base al minuendo en la columna anterior.
(1 + 8) -7 = 2	Sumar la base al minuendo, ya que el sustraendo > minuendo.
(2 + 8) - (3 + 1) = 6	Primero sumar 1 al sustraendo, ya que se aumentó la base en la columna anterior, después sumar la base al minuendo, ya que sustraendo > minuendo.
7 - (4 + 1) = 2	Sumar 1 al sustraendo debido a que se le sumó la base al minuendo en la columna anterior.
0 - 0 =0	Sin cambio, ya que no se cumple que sustraendo > minuendo, ni se sumó la base en la columna anterior.
(1 + 8) - 6 = 3	Sumar la base al minuendo, ya que sustraendo > minuendo.
4- (3 + 1) = 0	Sumar 1 al sustraendo, ya que se sumó la base al minuendo de la columna anterior.

Resta de números binarios

El algoritmo de la resta en sistema binario es el mismo que en el sistema decimal. Pero conviene repasar la operación de restar en decimal para comprender la operación binaria, que es más sencilla. Los términos que intervienen en la resta se llaman minuendo, sustraendo y diferencia.

Las restas básicas 0 - 0, 1 - 0 y 1 - 1 son evidentes:

0 - 0 = 0
1 - 0 = 1
1 - 1 = 0
0 - 1 = **1** (se transforma en 10 - 1 = 1) (en sistema decimal equivale a 2 - 1 = 1)

La resta 0 - 1 se resuelve, igual que en el sistema decimal, tomando una unidad prestada de la posición siguiente: 0 - 1 = **1** y *me llevo* 1, lo que equivale a decir en el sistema decimal, 2 - 1 = 1.

En decimal, por ejemplo tienes 100-19, obviamente a 0 no le puedes quitar 9, así que debemos tomar prestado 1 para volverlo un 10 (en decimal la base es 10), y así si 10-9=1.

En binarios pasa lo mismo, no le puedes quitar 1 a 0, debes de tomar 1 prestado al de un lado, pero cuidado aquí viene lo complicado tu número no se va a volver 10, recuerda que en binario la base es 2 y por lo tanto se volverá 2 en binario, y ahora sí a 2 le quitas 1, 2-1=1, y continuas restando pero recuerda que llevas 1, porque pediste prestado.

Ejemplo para que le entiendas mejor, vamos a restar 201 - 67, ya sabemos que es 134, vamos a hacerlo en binario:

```
  1 1 0 0 1 0 0 1.......................201
- 0 1 0 0 0 0 1 1.......................67
```

Tomamos los dos últimos números, 1-1 es igual a 0, y no llevamos nada (no pedimos prestado)

```
  1 1 0 0 1 0 0 1
- 0 1 0 0 0 0 1 1
------------------------
                0
```

Ahora la siguiente columna 0-1, ya dijimos que no se puede, así que va a tomar 1 prestado al de la columna del lado izquierdo, sé que vas a decir "es un cero, no nos puede prestar 1", lo que pasa es que ese cero le pide a su vez al de lado, y así hasta que encuentres un 1, pero no te fijes en eso, vamos a seguir restando y no nos vamos a preocupar por eso ahora, entonces ahora nos prestaron 1 (no importa quién) y tenemos un 1 0 (este número es 2 en binario no 10 en decimal, no te vayas a confundir), entonces en binario tienes 10-1, que en decimal es 2-1=1, y llevamos 1 (porque pedimos 1 prestado)

```
  1 1 0 0 1 0 0 1 arriba
- 0 1 0 0 0 0 1 1 abajo
------------------------
              1 0
```

Para la siguiente columna tenemos 0 - 0, pero recuerda que tomamos 1 prestado así que en realidad tenemos 0 - 1 (le sumamos el 1 al de abajo), de nuevo tenemos que pedir prestado y entonces tenemos en binaria 1 0 - 1 que en decimal es 2-1=1, y de nuevo llevamos 1

```
  1 1 0 0 1 0 0 1
- 0 1 0 0 0 0 1 1
------------------------
            1 1 0
```

Continuamos con 1 - 0, pero como llevamos 1 tenemos ahora 1 - 1, esto si lo podemos resolver 1 - 1 = 1 (en binario y decimal).

```
  1 1 0 0 1 0 0 1
- 0 1 0 0 0 0 1 1
------------------------
          0 1 1 0
```

Lo demás es muy fácil:
0 - 0=0
0 - 0=0
1 - 1=0
1 - 0=1

```
  1 1 0 0 1 0 0 1
- 0 1 0 0 0 0 1 1
------------------------
  1 0 0 0 0 1 1 0      que en decimal es 134.
```

Es lo mismo que la resta en decimal, pides prestado y llevas, nada más debes de ser cuidadoso y recordar que tu base es 2.

"En este mundo solo existen 10 tipos de personas, las que saben binario y las que no"

Multiplicación

La forma en que se multiplica en decimal es la misma en que se llevan a cabo las multiplicaciones en otros sistemas numéricos, la única diferencia es la base.

Ejemplo. Multiplicación en el sistema decimal:

$$
\begin{array}{r}
8057.23_{(10)} \\
\times \quad 53.7_{(10)} \\
\hline
5640061 \\
2417169 \\
4028615 \\
\hline
432673.251_{(10)}
\end{array}
$$

Explicación por columna:

7 X 3 = 21	Como el 21 no es un dígito válido en decimal, se divide entre la base para obtener cociente = 2 y resto = 1. El resto se coloca debajo de la línea v el cociente se suma en el producto de la siguiente columna.
7 X 2 + 2 = 16	Como el 16 no es un dígito válido en el sistema decimal, se realiza lo mismo que en el caso anterior para obtener cociente = 1 v resto = 6.

El procedimiento seguido en el sistema decimal es el que se realiza en cualquier sistema numérico, tomando en cuenta que cuando la cantidad resultante no es un dígito válido en dicho sistema entonces se debe dividir entre la base; en octal se debe dividir entre 8, en hexadecimal entre 16, en vigesimal entre 20 y así sucesivamente. La forma en que se suman en el sistema decimal los resultados obtenidos en las diferentes columnas, es la misma en que se suman en otro sistema. Como se muestra en el ejemplo en la primera columna se baja el 1 porque no tiene otro dígito con qué sumarse, en la segunda columna se suma 6 + 9 = 15 y como el 15 no es un dígito válido en decimal entonces se debe dividir entre la base para obtener cociente = 1 y resto = 5, el resto se pone debajo de la línea y el cociente se suma a los dígitos de la siguiente columna de la izquierda, y así se continúa hasta terminar.

Ejemplo. Multiplicación en el sistema binario:

		1	0	0	1	1	.	0	$1_{(2)}$
	×					1	.1	0	$1_{(2)}$
			1	0	0	1	1	0	1
		0	0	0	0	0	0	0	
	1	0	0	1	1	0	1		
1	0	0	1	1	0	1			
1	1	1	1	1	.0	1	0	0	$1_{(2)}$

Entre menor sea la base del sistema numérico es más sencillo realizar operaciones aritméticas, ya que el número de dígitos válidos también se reduce en la misma proporción. Como se muestra en el ejemplo, en el sistema binario solamente hay posibilidades de 0 ó 1 y esto simplifica el trabajo.

En cualquier sistema, al multiplicar una cantidad por 1 se obtiene la misma cantidad, por esa razón en el sistema binario al multiplicar 1 por el multiplicando resulta el mismo multiplicando y al multiplicar 0 por el multiplicando resulta una fila de ceros. Al sumar las columnas se pueden observar casos como el que ocurre en la tercera columna de derecha a izquierda, donde 1 + 0 + 1 = 2, pero como el2 no es un dígito válido en binario se debe dividir entre la base, obteniéndose cociente = 1 y resto = 0 por lo que el resto se coloca debajo de la línea y el cociente se suma con los dígitos de la siguiente columna, de forma que en la cuarta columna se deben sumar 1 + 1 + 0 + 0 + 1 = 3, donde el primer 1 es el cociente de la línea anterior. Como el 3 no es un dígito válido en binario, se divide entre la base y se obtiene cociente = 1 y resto = 1. En lo que sigue se procede de la misma manera hasta terminar de sumar todas las columnas. El punto que separa a la parte entera de la fraccionaria se coloca también de manera semejante a como se realiza en el sistema decimal: se cuentan los decimales tanto del multiplicando como del multiplicador.

Producto de números binarios

La tabla de multiplicar para números binarios es la siguiente:

X	0	1
0	0	0
1	0	1

El algoritmo del producto en binario es igual que en números decimales; aunque se lleva a cabo con más sencillez, ya que el 0 multiplicado por cualquier número da 0, y el 1 es el elemento neutro del producto.

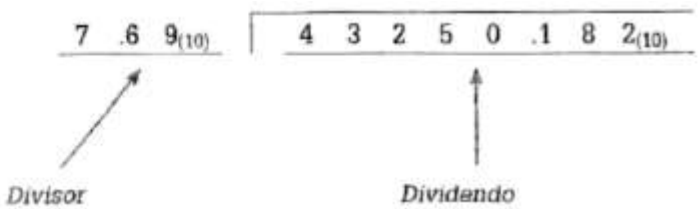
Por ejemplo, multipliquemos 10110 por 1001:

```
 10110  X  1001
                10110
              00000
            00000
          10110
          11000110
```

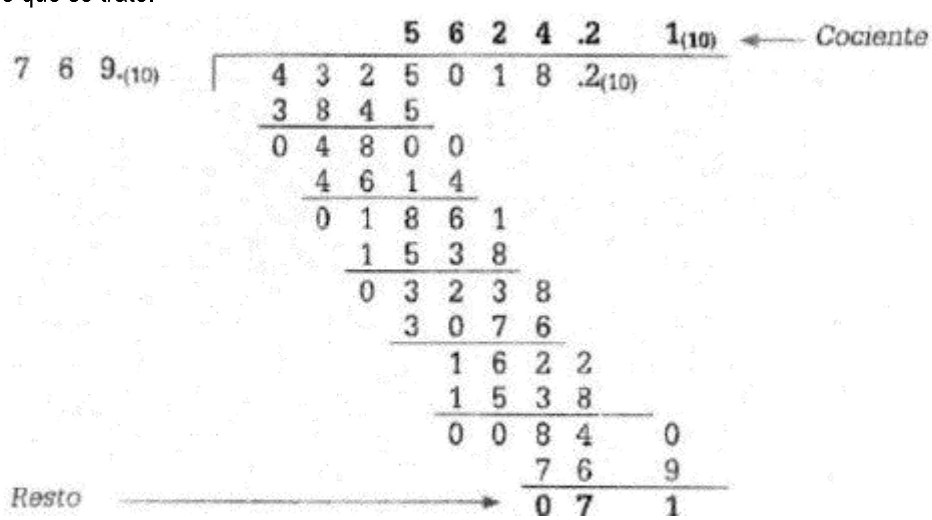
División

Se sabe que la división involucra operaciones de resta y multiplicación, por lo que es más complicada que las tres operaciones aritméticas anteriores. En este caso lo que se recomienda es usar lo que se conoce como división desarrollada, la cual permite realizar primero la multiplicación y después la resta, ya que de otra forma el tratar de llevar a cabo tanto la multiplicación como la resta en un sistema numérico con el que no se está familiarizado podría ser muy complicado.

Ejemplo. División en el sistema decimal:



En una división en el sistema decimal el dividendo puede tener o no punto decimal, pero el divisor no debe tenerlo o bien lo debe tener al final. En este ejemplo se debe recorrer el punto decimal dos posiciones, para mandar el punto decimal al extremo derecho del divisor. Si se recorre el punto decimal cierto número de posiciones en el divisor, también se debe recorrer esas mismas posiciones en el dividendo. En todos los casos en que sea necesario esto se debe de hacer antes de llevar a cabo la división, independientemente del sistema numérico de que se trate.



Después de recorrer el punto decimal al lado derecho del dígito menos significativo del divisor, v de recorrer ese mismo número de posiciones el punto decimal en el dividendo, se procede a llevar a cabo la división.

Como el divisor tiene tres dígitos y en este caso no cabe ninguna vez en tres dígitos del dividendo ya que $769_{(10)} > 432_{(10)}$, se toman cuatro dígitos del dividendo y se encuentra que cociente = 5, después se multiplica este cociente por el divisor, $5 \times 769_{(10)} = 3845_{(10)}$, y dicho resultado se resta de los cuatro dígitos del dividendo para obtener $4325_{(10)} - 3845_{(10)} = 0480_{(10)}$. luego se baja el siguiente dígito, se encuentra el cociente, se multiplica dicho cociente por el divisor y se resta del dividendo y así sucesivamente hasta terminar. Es importante mencionar que si después de bajar un dígito más del dividendo, el divisor no cabe ninguna vez en dicha cantidad, entonces se pondrá 0 como cociente y se bajará otro dígito. También, una vez que se han bajado todos los dígitos del dividendo y se desean más decimales, se deberán agregar más ceros a la cantidad que resulta de la resta. Todo esto ya se sabe, porque se está familiarizado con el sistema decimal y es válido en todos los sistemas.

División de números binarios

La división en binario es similar al decimal; la única diferencia es que a la hora de hacer las restas, dentro de la división, éstas deben ser realizadas en binario.

Ejemplo

Dividir 100010010 (274) entre 1101 (13):

```
 100010010 |1101
-0000        010101
 10001
 -1101
   01000
  - 0000
    10000
   - 1101
     00011
    - 0000
       01110
      - 1101
       00001
```

1.4 Algoritmos de Booth para la multiplicación y división en binario.

El algoritmo de Booth es un método rápido y sencillo para obtener el producto de dos números binarios con signo en notación complemento a dos.

Debemos saber que un número binario está formado por bits de ceros y unos, y que se puede traducir a decimal fácilmente de la siguiente forma:

128	64	32	16	8	4	2	1
0	1	0	1	0	1	1	0

Sabiendo que la posición de cada bit es 2^n (elevado a n) y partimos de n=0 de derecha a izquierda, sólo queda realizar la suma total de multiplicar por dicho bit, en este caso:

(0·2^7+1·2^6+0·2^5+1·2^4+0·2^3+1·2^2+1·2^1+0·2^0 = 86).

También debemos saber que el complemento a uno de un número binario es cambiar sus ceros por unos, y sus unos por ceros (complementar): (010010 -> ca1:101101) y que el complemento a dos de un número binario es el resultado de sumar 1 al complemento a uno de dicho número binario:

número binario:
0101 0110

$$
\begin{array}{lr}
\text{Ca1:} & 1010\ 1001 \\
 & +1 \\
\hline
\text{Ca2:} & 1010\ 1010
\end{array}
$$

Realizar una suma con dos números binarios es tarea fácil, pero la multiplicación resulta algo más complicada. Con el algoritmo de Booth, resulta mucho más sencillo de implementar. Partimos del ejemplo de la multiplicación 6·2=12:

```
0000 0110  (6)
0000 0010  (2)
└─8 bits─┘
```

```
   multiplicando                  bit extra
A=0000 0110  0000 0000  0
   multiplicando en ca2
S=1111 1010  0000 0000  0
                 multiplicador
P=0000 0000  0000 0010  0
```

Como se puede ver en la imagen superior, partiendo de los números binarios de la multiplicación 6·2 (multiplicando y multiplicador) creamos tres nuevos números binarios del doble de tamaño (16 en el ejemplo): A, S y P.

Partiendo del número P (producto) comenzamos a comparar los últimos 2 bits de la derecha, siguiendo los casos base del recuadro:

```
CASOS BASE
0 0 -> No se realiza
       ninguna acción
0 1 -> P = P + A
1 0 -> P = P + S
1 1 -> No se realiza
       ninguna acción
```

Se realizará esta comparación 8 veces en este ejemplo (número de bits de los operandos) y al final de cada comparación, realizamos un desplazamiento de un bit hacia la derecha, manteniendo el último bit de la izquierda, y descartando el último bit del lado contrario. Si hacemos una traza paso a paso nos quedarían los siguientes resultados:

```
0000 0000 0000 001[0 0]          ->
0000 0000 0000 000[1 0]   P=P+S
1111 1010 0000 000[1 0]          ->
1111 1101 0000 000[0 1]   P=P+A
0000 0011 0000 000[0 1]          ->
0000 0001 1000 000[0 0]          ->
0000 0000 1100 000[0 0]          ->
0000 0000 0110 000[0 0]          ->
0000 0000 0011 000[0 0]          ->
0000 0000 0001 100[0 0]          ->


0000 0000 0000 1100 [0]    (12)
```

Finalmente obtenemos el número en binario resultante (12 en este ejemplo), descartando el bit extra que hemos añadido al principio del procedimiento y que se encuentra en el extremo a la derecha.

1.5 Aplicación de los sistemas numéricos en la computación.

Existe una cantidad infinita de sistemas numéricos, sin embargo, para una computadora, únicamente existen 4, que son el Binario (con base 2), el octal (con base 8), el decimal (base 10) y hexadecimal (base 16). Detallaremos el uso década uno de ellos por la computadora.

Sistema Binario

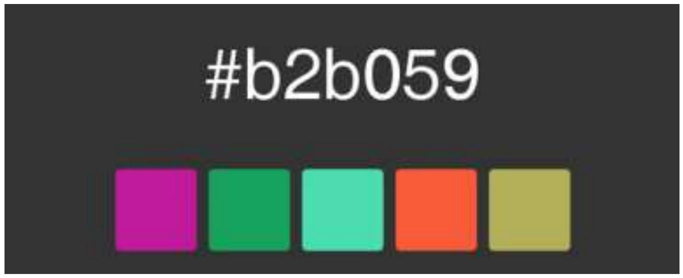
El Sistema Binario, por ser el sistema base de la computación y el único entendido de manera nativa por una computadora, es el sistema en el que está escrita toda instrucción, dato, etc. Está compuesto por dos únicos dígitos que 1 y 0 o como en realidad trabaja la computadora, "apagado" y "encendido" y se es como representa todos los datos con los que trabaja la computadora, desde sumas bajo nivel: el hardware. Estos dígitos son llamados bits

Sistema Octal

Para trabajar la computadora agrupa a los bits en grupos de ocho, a los cuales se denomina byte y es esta la razón por la que es tan importante el sistema octal, sin embargo una computadora no puede trabajar con el sistema octal como tal, sino que utiliza su conversión en sistema binario, usando tres bits para cada digito octal

Sistema Hexadecimal

El sistema hexadecimal es empleado al indexar la memoria o al representar un byte debido a que al contener más dígitos es posible usar menos números para representar números más grandes, haciendo posible que un byte, conformado por 8 bits o términos binarios, se represente con solo dos términos hexadecimales, lo que es un ahorro de información. Sin embargo, la computadora tampoco reconoce el sistema hexadecimal como tal y, al igual que el sistema octal, lo representa con términos binarios, empleando conjuntos de cuatro bits, para cada término hexadecimal. Sin embargo al presentar información al usuario es más factible presentar A9 que 10101001



Sistema Decimal

Por último el sistema decimal únicamente se utiliza al interactuar con el usuario, debido a que un usuario común no está acostumbrado a tratar con diferentes sistemas numéricos.