3

Conversão de Bases e Aritmética Computacional

3.1 NOTAÇÃO POSICIONAL — BASE DECIMAL

Desde os primórdios da civilização o Homem vem adotando formas e métodos específicos para representar números, tornando possível, com eles, contar objetos e efetuar operações aritméticas (de soma, subtração etc.).

A forma mais empregada de representação numérica é a chamada *notação posicional*. Nela, os algarismos componentes de um número assumem valores diferentes, dependendo de sua posição relativa no número. O valor total do número é a soma dos valores relativos de cada algarismo. Desse modo, é a posição do algarismo ou dígito que determina seu valor.

A formação de números e as operações com eles efetuadas dependem, nos sistemas posicionais, da quantidade de algarismos diferentes disponíveis no referido sistema. Há muito tempo a cultura ocidental adotou um sistema de numeração que possui dez diferentes algarismos — 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 — e, por essa razão, foi chamado de ***sistema decimal***. (Ver mais detalhes no Apêndice A — Sistemas de Numeração.)

A quantidade de algarismos disponíveis em um dado sistema de numeração é chamada de ***base***; a base serve para contarmos grandezas maiores, indicando a noção de grupamento. O sistema de dez algarismos, mencionado anteriormente, tem base 10; um outro sistema que possua apenas dois algarismos diferentes (0 e 1) é de base 2, e assim por diante.

Vamos exemplificar o conceito de sistema posicional. Seja o número 1303, representado na base 10, escrito da seguinte forma:

$1\ 3\ 0\ 3_{10}$

Em base decimal, por ser a mais usual, costuma-se dispensar o indicador da base, escrevendo-se apenas o número:

1303

Neste exemplo, o número é composto de quatro algarismos:

1, 3, 0 e 3

e cada algarismo possui um valor correspondente à sua posição no número.

Assim, o primeiro 3 (algarismo mais à direita) representa 3 unidades. Neste caso, o valor absoluto do algarismo (que é 3) é igual ao seu valor relativo (que também é 3), por se tratar da 1.ª posição (posição mais à direita, que é a ordem das unidades). Considerando-se o produto três vezes a potência 0 da base 10 ou

$3 \times 10^0 = 3$

enquanto o segundo 3 vale três vezes a potência 2 da base 10 ou

$3 \times 10^2 = 300$

E o último à esquerda vale uma vez a potência 3 da base 10, ou $1 \times 10^3 = 1000$.

O valor total do número seria então:

$1000 + 300 + 0 + 3 = 1303_{10}$

$1 \times 10^3 + 3 \times 10^2 + 0 \times 10^1 + 3 \times 10^0 = 1303_{10}$

Generalizando, num sistema qualquer de numeração posicional, um número N é expresso da seguinte forma:

$$N = (d_{n-1}\, d_{n-2}\, d_{n-3} \ldots d_1\, d_0)_b \tag{3.1}$$

onde:

d indica cada algarismo do número;

n − 1, n − 2, 1, 0 (índice) indicam a posição de cada algarismo;

b indica a base de numeração;

n indica o número de dígitos inteiros.

O valor do número pode ser obtido do seguinte somatório:

$$N = d_{n-1} \times b^{n-1} + d_{n-2} \times b^{n-2} + \ldots + d_1 \times b^1 + d_0 \times b^0 \tag{3.2}$$

Desse modo, na base 10, podemos representar um número:

$N = 3748$

onde:

$n = 4$ (quatro dígitos inteiros).

Utilizando a fórmula indicada na Eq. 3.1:

$d_{n-1} = 3$ ou $d_3 = 3$; $d_2 = 7$; $d_1 = 4$; $d_0 = 8$

ou obtendo seu valor de acordo com a fórmula mostrada em (3.2):

$N = 3 \times 10^3 + 7 \times 10^2 + 4 \times 10^1 + 8 \times 10^0 =$

$= 3000 + 700 + 40 + 8 = 3748_{10}$

Observação: Números fracionários são apresentados em detalhe no Apêndice A.

3.2 OUTRAS BASES DE NUMERAÇÃO

Vejamos, em seguida, como representar números em outra base de numeração.

Entre as bases diferentes de 10, consideremos apenas as bases 2 e potências de 2, visto que todo computador digital representa internamente as informações em algarismos binários, ou seja, trabalha em base 2. Como os números representados em base 2 são muito extensos (quanto menor a base de numeração, maior é a quantidade de algarismos necessários para indicar um dado valor) e, portanto, de difícil manipulação visual, costuma-se representar externamente os valores binários em outras bases de valor mais elevado. Isso permite maior compactação de algarismos e melhor visualização dos valores. Em geral, usam-se as bases octal ou hexadecimal, em vez da base decimal, por ser mais simples e rápido converter valores binários (base 2) para valores em bases múltiplas de 2.

Utilizando-se a notação posicional indicada na Eq. 3.1, representam-se números em qualquer base:

$(1011)_2$ — na base 2

$(342)_5$ — na base 5

$(257)_8$ — na base 8

No entanto, nas bases diferentes de 10, o valor relativo do algarismo (valor dependente de sua posição no número) é normalmente calculado usando-se os valores resultantes de operações aritméticas em base 10 e não na base do número (ver Apêndice A para mais detalhes) e, portanto, o valor total do número na base usada será expresso em termos de grandeza na base 10.

Exemplo 3.1

Seja o número na base 2: $(1011)_2$ (usou-se a descrição da Eq. 3.1).

Se aplicássemos a Eq. 3.2, teríamos:

$1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 =$

$= 8 + 0 + 2 + 1 = (11)_{10}$

Este valor 11 está expresso na base 10 e não na base 2. Portanto, será $(11)_{10}$.

Exemplo 3.2

$$\begin{aligned}(1043)_5 &= 1 \times 5^3 + 0 \times 5^2 + 4 \times 5^1 + 3 \times 5^0 = \\ &= 125 + 0 + 20 + 3 = (148)_{10}\end{aligned}$$

Sobre o assunto, podemos concluir:

a) O número máximo de algarismos diferentes de uma base é igual ao valor da base.

Exemplo:

- na base 10 temos 10 dígitos: de 0 a 9;
- na base 2 temos apenas dois dígitos: 0 e 1;
- na base 5 temos cinco dígitos: de 0 a 4.

b) O valor do algarismo mais à esquerda (mais significativo) de um número de **n** algarismos inteiros é obtido pela multiplicação de seu valor absoluto (algarismo d_{n-1}) pela base elevada à potência **(n − 1)**, ou seja, $(d_{n-1} \times b^{n-1})$.

c) O valor total do número é obtido somando-se **n** valores, cada um expressando o valor relativo de um dos **n** algarismos componentes do número.

Exemplo 3.3

a) 375_{10}

$n = 3$	(3 algarismos)
$3 \times 10^2 + 7 \times 10^1 + 5 \times 10^0 =$	(3 produtos)
$= 300 + 70 + 5$	(3 valores a somar)

b) 11101_2 (5 algarismos)

$\underline{1 \times 2^4}$	$+ \underline{1 \times 2^3}$	$+ \underline{1 \times 2^2}$	$+ \underline{0 \times 2^1}$	$+ \underline{1 \times 2^0}$	(5 produtos − 5 valores)
1.º prod.	2.º prod.	3.º prod.	4.º prod.	5.º prod.	
16	+ 8	+ 4	+ 0	$+ 1 = 29_{10}$	

A base do sistema binário é 2 e, conseqüentemente, qualquer número, quando representado nesse sistema, consiste exclusivamente em dígitos 0 e 1. O termo dígito binário é chamado ***bit***, contração do termo inglês *bi*nary digi*t*.

Por exemplo, o número binário 11011 possui cinco dígitos, ou algarismos binários. Diz-se que o referido número é constituído de 5 bits.

Em bases de valor superior a 10, usam-se letras do alfabeto para a representação de algarismos maiores que 9. Uma dessas bases é especialmente importante em computação — trata-se da base 16 ou hexadecimal, por ser de valor potência de 2 (como a base 8).

Nessa base, os "algarismos" A, B, C, D, E e F representam, respectivamente, os valores (da base 10): 10, 11, 12, 13, 14 e 15.

Na base 16 (hexadecimal), dispomos de 16 algarismos (não números) diferentes:

0, 1, 2, 3, ..., 9, A, B, C, D, E e F

Um número nessa base é representado na forma da Eq. 3.1:

$(1A7B)_{16}$

O seu valor na base 10 será obtido usando-se a Eq. 3.2:

$1 \times 16^3 + 10 \times 16^2 + 7 \times 16^1 + 11 \times 16^0 = 4096 + 2560 + 112 + 11 = 6779_{10}$

Observemos que na Eq. 3.2 foram usados os valores 10 (para o algarismo A) e 11 (para o algarismo B) para multiplicar as potências de 16. Por isso, obtivemos o valor do número na base 10.

Em outras palavras, utilizamos valores e regras de aritmética da base 10 e, por isso, o resultado encontrado é um valor decimal. A Tabela 3.1 mostra a representação de números nas bases 2, 8, 10 e 16.

Pela tabela, podemos observar que os dígitos octais e hexadecimais correspondem a combinações de 3 (octais) e 4 (hexadecimais) bits (algarismos binários). Sendo a base desses sistemas de valor maior que a base 2 e tendo em vista essa particularidade na representação de números nas bases 8 e 16 em relação à base 2, verifica-se que é possível converter rapidamente números da base 2 para as bases 8 ou 16, ou vice-versa.

Por exemplo, o número (101111011101), na base 2, possui 12 algarismos (bits), mas pode ser representado com quatro algarismos octais ou com apenas três algarismos hexadecimais:

$(101111011101)_2 = (5735)_8$

porque 101 = 5; 111 = 7; 011 = 3 e 101 = 5

$(101111011101)_2 = (BDD)_{16}$

porque 1011 = B; 1101 = D; 1101 = D.

Tabela 3.1

Base 2	Base 8	Base 10	Base 16
0	0	0	0
1	1	1	1
10	2	2	2
11	3	3	3
100	4	4	4
101	5	5	5
110	6	6	6
111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1111	17	15	F
10000	20	16	10
10001	21	17	11

3.3 CONVERSÃO DE BASES

Uma vez entendido como representar números em notação posicional e como esta notação é aplicável em qualquer base inteira, podemos exercitar a conversão de números de uma base para outra.

Interessa-nos, principalmente, verificar o processo de conversão entre bases múltiplas de 2, e entre estas e a base 10, e vice-versa.

3.3.1 Conversão entre Bases Potência de 2

3.3.1.1 Entre as Bases 2 e 8

Como $8 = 2^3$, um número binário (base 2) pode ser facilmente convertido para o seu valor equivalente na base 8 (octal). Se o número binário for inteiro, basta dividi-lo, da direita para a esquerda, em grupos de 3 bits (o último grupo, à esquerda, não sendo múltiplo de 3, preenche-se com zeros à esquerda). Então, para cada grupo, acha-se o algarismo octal equivalente, conforme mostrado na Tabela 3.1.

A conversão de números da base 8 para a 2 é realizada de forma semelhante, no sentido inverso; substitui-se cada algarismo octal pelos seus 3 bits correspondentes (ver Tabela 3.1).

Exemplo 3.4

1) $(111010111)_2 = (\quad)_8$

$(111)\ (010)\ (111)_2 = (727)_8$

7 2 7

2) $(1010011111)_2 = (\quad)_8$

$(001)\ (010)\ (011)\ (111)_2 = (1237)_8$

1 2 3 7

3) $(327)_8 = (\quad)_2$

$(011)\ (010)\ (111)_2 = (011010111)_2$ ou $(11010111)_2$ Obs.: Naturalmente, despreza-se o(s) zero(s) à esquerda do número.

3 2 7

4) $(673)_8 = (\quad)_2$

$(110)\ (111)\ (011)_2 = (110111011)_2$

6 7 3

3.3.1.2 Entre as Bases 2 e 16

O procedimento de conversão entre números binários e hexadecimais (base 16) é idêntico ao da conversão entre as bases 2 e 8, exceto que, neste caso, a relação é $16 = 2^4$.

Desse modo, um algarismo hexadecimal é representado por 4 bits (ver Tabela 3.1). Converte-se um número binário em hexadecimal dividindo-se este número em grupos de 4 bits da direita para a esquerda.

A conversão de hexadecimal para binário é obtida substituindo-se o algarismo hexadecimal pelos 4 bits correspondentes, de acordo com os valores indicados na Tabela 3.1.

Exemplo 3.5

1) $(1011011011)_2 = (\quad)_{16}$

$(0010)\ (1101)\ (1011)_2 = (2DB)_{16}$

2 D B

2) $(10011100101101)_2 = (\ )_{16}$

$(0010)\ (0111)\ (0010)\ (1101)_2 = (272D)_{16}$

$\ \ 2 \quad\quad 7 \quad\quad 2 \quad\quad D$

3) $(306)_{16} = (\ )_2$

$(0011)\ (0000)\ (0110)_2 = (1100000110)_2$

$\ \ 3 \quad\quad 0 \quad\quad 6$

4) $(F50)_{16} = (\ )_2$

$(1111)\ (0101)\ (0000)_2 = (111101010000)_2$

$\ \ F \quad\quad 5 \quad\quad 0$

3.3.1.3 Entre as Bases 8 e 16

O processo de conversão utiliza os mesmos princípios antes apresentados. No entanto, como a base de referência para as substituições de valores é a base 2, esta deve ser empregada como intermediária no processo. Ou seja, convertendo-se da base 8 para a base 16, deve-se primeiro efetuar a conversão para a base 2 (como mostrado nos subitens anteriores) e depois para a base 16. E o mesmo ocorre se a conversão for da base 16 para a base 8.

Exemplo 3.6

1) $(3174)_8 = (\ )_{16}$

Primeiro, converte-se o número da base 8 para a base 2:

$(011)\ (001)\ (111)\ (100)_2 = (011001111100)_2$

Em seguida, converte-se da base 2 para a base 16, separando-se os algarismos de 4 em 4, da direita para a esquerda:

$(0110)\ (0111)\ (1100) = (67C)_{16}$

$\ \ 6 \quad\quad 7 \quad\quad C$

2) $(254)_8 = (\ )_{16}$

$= (010)\ (101)\ (100)_2 = (010101100)_2$

$= (1010)\ (1100)_2 = (AC)_{16}$

3) $(2E7A)_{16} = (\ )_8$

$= (0010)\ (1110)\ (0111)\ (1010)_2 = (0010111001111010)_2 =$

$= (010)\ (111)\ (001)\ (111)\ (010)_2 = (27172)_8$

4) $(3C7)_{16} = (\ )_8$

$= (0011)\ (1100)\ (0111)_2 = (1111000111)_2 =$

$= (001)\ (111)\ (000)\ (111)_2 = (1707)_8$

3.3.2 Conversão de Números de uma Base B para a Base 10

A conversão de um número, representado em uma base B qualquer, para seu correspondente valor na base 10 é realizada empregando-se a Eq. 3.2. A melhor maneira de compreender o processo de conversão consiste na realização de alguns exemplos práticos, onde se indica, detalhadamente, a aplicação da referida equação.

Os exemplos apresentados referem-se apenas a números inteiros. No Apêndice A — Sistemas de Numeração, são detalhados os diversos processos de conversão de números inteiros e fracionários.

Exemplo 3.7

1) $(101101)_2 = (\ \ )_{10}$

Substituindo, na Eq. 3.2, as letras pelos valores do exemplo, teremos:

$b = 2$ (a base origem do número a ser convertido)

$n = 6$ (6 algarismos)

$n - 1 = 5$ (expoente do 1.º produto mais à esquerda)

$d_{n-1} = 1$ (algarismo mais à esquerda)

1.º produto: $d_{n-1} \times b^{n-1} = 1 \times 2^5$

Os demais produtos seguem a seqüência da Eq. 3.2, resultando em:

$1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 =$

$= 32 + 0 + 8 + 4 + 0 + 1 = (45)_{10}$

2) $(27)_8 = (\ \ )_{10}$

Da mesma maneira, substitui-se na Eq. 3.2:

$b = 8$

$n = 2$

$n - 1 = 1$ $\qquad \left|\begin{array}{c|c|c|c|c|c|c} d_{n-1} & \times & B^{n-1} & + & d_0 & \times & B^0 \\ 2 & \times & 8^1 & + & 7 & \times & 8^0 \end{array}\right|$

$d_{n-1} = 2$

Valor total:

$2 \times 8^1 + 7 \times 8^0 = 16 + 7 = (23)_{10}$

3) $(2A5)_{16} = (\ \ )_{10}$

$2 \times 16^2 + 10 \times 16^1 + 5 \times 16^0 =$

$= 512 + 160 + 5 = (677)_{10}$

4) $(6734)_8 = (\ \ )_{10}$

$6 \times 8^3 + 7 \times 8^2 + 3 \times 8^1 + 4 \times 8^0 =$

$= 3072 + 448 + 24 + 4 = (3548)_8$

5) $(27)_8 = (\ \ )_{10}$

$2 \times 8^1 + 7 \times 8^0 = 23_{10}$

Observação: No desenvolvimento foram suprimidos os produtos em que os algarismos eram 0, visto que o resultado seria também sempre zero.

6) $(457)_9 = (\ \ )_{10}$

$4 \times 9^2 + 5 \times 9^1 + 7 \times 9^0 =$

$= 324 + 45 + 7 = (376)_{10}$

7) $(243)_5 = (\ \ )_{10}$

$2 \times 5^2 + 4 \times 5^1 + 3 \times 5^0 =$

$= 50 + 20 + 3 = (73)_{10}$

3.3.3 Conversão de Números Decimais para uma Base B

A conversão de números, representados na base 10, para seus valores equivalentes em uma base B qualquer é efetuada através de um processo inverso ao do subitem anterior (base B para base 10).

A conversão é obtida dividindo-se o número decimal pelo valor da base desejada; o resto encontrado é o algarismo menos significativo do valor na base B (mais à direita). Em seguida, divide-se o quociente encontrado pela base B; o resto é o algarismo seguinte (à esquerda); e assim, sucessivamente, vão-se dividindo os quocientes pelo valor da base até se obter quociente de valor zero. Em cada divisão, o resto encontrado é um algarismo significativo do número na nova base; o primeiro resto encontrado é o valor do algarismo menos significativo (mais à direita), e o último resto será o algarismo mais significativo (mais à esquerda).

Na realidade, o algoritmo de conversão pode ser definido com vários critérios de parada, tais como:

a) Enquanto o quociente for diferente de zero:
   - dividir dividendo por divisor;
   - extrair resto como algarismo e colocá-lo à esquerda do anterior;
   - repetir.

   Quando o quociente for igual a zero, parar.

b) Enquanto o dividendo for maior que o divisor:
   - dividir dividendo por divisor;
   - extrair resto como algarismo e colocá-lo à esquerda do anterior;
   - repetir.

Usar o dividendo (que agora é menor que o divisor) como último algarismo à esquerda (algarismo mais significativo).

No Apêndice A — Sistemas de Numeração, são detalhados os procedimentos de conversão de números inteiros e fracionários.

Exemplo 3.8

1) $(3964)_{10} = (\ \ )_8$

   $3964/8 = 495$ resto$_0$ = 4 (algarismo menos significativo)

   $495/8 = 61$ resto$_1$ = 7

   $61/8 = 7$ resto$_2$ = 5

   $7/8 = 0$ resto$_3$ = 7 (algarismo mais significativo)

   Observa-se que o primeiro resto encontrado (algarismo 4) é o algarismo mais à direita do número.

   O número é, então, $(7574)_8$.

2) $(483)_{10} = (\ \ )_8$

   $483/8 = 60$ resto$_0$ = 3 (algarismo menos significativo, mais à direita)

   $60/8 = 7$ resto$_1$ = 4

   $7/8 = 0$ resto$_3$ = 7 (algarismo mais significativo, mais à esquerda)

   O número é $(743)_8$.

   Para verificar, façamos o processo inverso, isto é: converter $(743)_8$ para a base 10.

   $7 \times 8^2 + 4 \times 8^1 + 3 \times 8^0 =$

   $= 448 + 32 + 3 = (483)_{10}$

3) $(45)_{10} = (\ )_2$

$45/2 = 22$ $resto_0 = 1$ (algarismo menos significativo, mais à direita)
$22/2 = 11$ $resto_1 = 0$
$11/2 = 5$ $resto_2 = 1$
$5/2 = 2$ $resto_3 = 1$
$2/2 = 1$ $resto_4 = 0$
$1/2 = 0$ $resto_5 = 1$ (algarismo mais significativo, mais à esquerda)

O número é 101101_2.

4) $(97)_{10} = (\ )_2$

$97/2 = 48$ $resto_0 = 1$ (algarismo menos significativo)
$48/2 = 24$ $resto_1 = 0$
$24/2 = 12$ $resto_2 = 0$
$12/2 = 6$ $resto_3 = 0$
$6/2 = 3$ $resto_4 = 0$
$3/2 = 1$ $resto_5 = 1$
$1/2 = 0$ $resto_6 = 1$ (algarismo mais significativo)

O número é $(1100001)_2$.

5) $(2754)_{10} = (\ )_{16}$

$2754/16 = 172$ $resto_0 = 2$ algarismo 2_{16} (algarismo menos significativo)
$172/16 = 10$ $resto_1 = 12$ algarismo C_{16}
$10/16 = 0$ $resto_2 = 10$ algarismo A_{16} (algarismo mais significativo)

O número é $(AC2)_{16}$.

6) $(490)_{10} = (\ )_{16}$

$490/16 = 30$ $resto_0 = 10_{10}$ algarismo A_{16} (algarismo menos significativo)
$30/16 = 1$ $resto_1 = 14_{10}$ algarismo E_{16}
$1/16 = 0$ $resto_2 = 1_{10}$ algarismo 1_{16} (algarismo mais significativo)

O número é $(1EA)_{16}$.

É possível simplificar o processo de conversão de valores da base 2 para a base 10 e vice-versa. Para tanto, basta considerar o seguinte:

a) A Eq. 3.2 estabelece o valor de um número pela soma de produtos:

$d_{n-1} \times b^{n-1} + \ldots$

b) Cada produto é constituído de duas parcelas: a primeira é o algarismo correspondente à posição em que se encontra e a segunda é a potência da base, cujo índice indica a posição.

c) No caso de a base ser 2, os algarismos só podem assumir o valor 0 ou 1. Dessa forma, o resultado do produto somente pode ser 0 ou o próprio valor da potência de 2.

Exemplos:

$101 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

O primeiro produto, 1×2^2, tem valor igual a $2^2 = 4$. Isto é, como o algarismo é 1, então 1×2^2 ou apenas 2^2 tem mesmo valor. No caso do segundo produto, 0×2^1 é igual a zero. O terceiro produto, igual ao primeiro, é 1×2^0 ou $2^0 = 1$.

d) As potências de 2, da direita para a esquerda, crescem da seguinte forma:

$2^0 = 1$ (potência zero, correspondente à posição mais à direita)

$2^1 = 2$; $2^2 = 4$; $2^3 = 8$; $2^4 = 16$ etc.

Ou seja:

... 6	5	4	3	2	1	0	←	posição
... 2^6	2^5	2^4	2^3	2^2	2^1	2^0	←	potência
... 64	32	16	8	4	2	1	←	valor

Em conseqüência, converter um número da base 2 para a base 10 consiste essencialmente em somar as potências de 2 correspondentes às posições onde o algarismo é igual a 1, desprezando as potências onde o algarismo é zero.

Exemplo 3.9

Efetuar as seguintes conversões:

1) $(110011)_2 = (\ )_{10}$

5	4	3	2	1	0	←	posição
1	1	0	0	1	1	←	algarismo
2^5	2^4	—	—	2^1	2^0	←	potências válidas para somar
32	16	—	—	2	1	←	valores

Valor em base 10: $32 + 16 + 2 + 1 = (51)_{10}$

2) $(100111)_2 = (\ )_{10}$

32	16	8	4	2	1	←	potências
1	0	0	1	1	1	←	algarismos

Valor em base 10: somam-se as potências válidas, correspondentes à posição onde o algarismo é 1.

$32 + 4 + 2 + 1 = (39)_{10}$

3.4 ARITMÉTICA NÃO-DECIMAL

Neste item serão apresentados procedimentos para realização das quatro operações aritméticas (adição, subtração, multiplicação e divisão) de números não-decimais (qualquer outro sistema de base diferente de 10), essencialmente os de base 2 e potência de 2, que interessam aos sistemas de computação.

Os números serão inteiros, sem limite de tamanho e positivos (sem sinal).

No Apêndice A — Sistemas de Numeração, são detalhados procedimentos para execução de operações aritméticas, com números binários, octais e hexadecimais, incluindo valores inteiros e fracionários, porém ainda sem sinal.

No Cap. 7, são detalhados procedimentos para execução de operações aritméticas com números positivos e negativos (inclusão do sinal nos números), inteiros e fracionários, bem como aqueles expressos na forma BCD (Binary Coded Decime). Os procedimentos estão relacionados ao processo efetivamente realizado no interior da unidade de processamento dos computadores. Já, neste capítulo, procura-se descrever procedimentos apenas matemáticos, para familiarizar o leitor com operações matemáticas não-decimais.

Finalmente, não se está levando em conta qualquer limite dos números, ou seja, a quantidade máxima de algarismos permitida para um dado número, o que é uma efetiva preocupação no caso dos computadores. Trata-se do problema de *overflow* ou estouro do limite, quando uma operação aritmética resulta em um valor acima do limite máximo possível (ver Cap. 7).