



# Sistemas de Numeração

## 1 – Sistemas de numeração

**S**istemas de numeração são sistemas para representação de números. Historicamente vários sistemas de numeração foram desenvolvidos de acordo com os avanços das técnicas matemáticas e da própria civilização. Uma das hipóteses de que tenhamos desenvolvido o sistema decimal de numeração está relacionada à contagem com os 10 dedos das mãos.

Na era digital acostumamo-nos a ouvir falar de *bits* ou *bytes*. Sabemos que **bit** é a menor unidade de informação na memória do computador (0 ou 1, aberto ou fechado, falso ou verdadeiro). Um **byte** corresponde a 8 bits, mas um byte consegue armazenar um código que corresponde somente a um único número, letra ou símbolo. A combinação de cadeias de bits é que permite representar diferentes números agrupando-se na forma de bytes.

É importante ter uma boa noção de notação e bases numéricas quando queremos estudar Computação porque isso tem muitos aproveitamentos na área.

## Notação e bases

A quantidade de algarismos disponíveis num sistema de numeração designa-se de base.



### Decimal (base 10)

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



Binário (base 2)

0, 1



Octal (base 8)

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



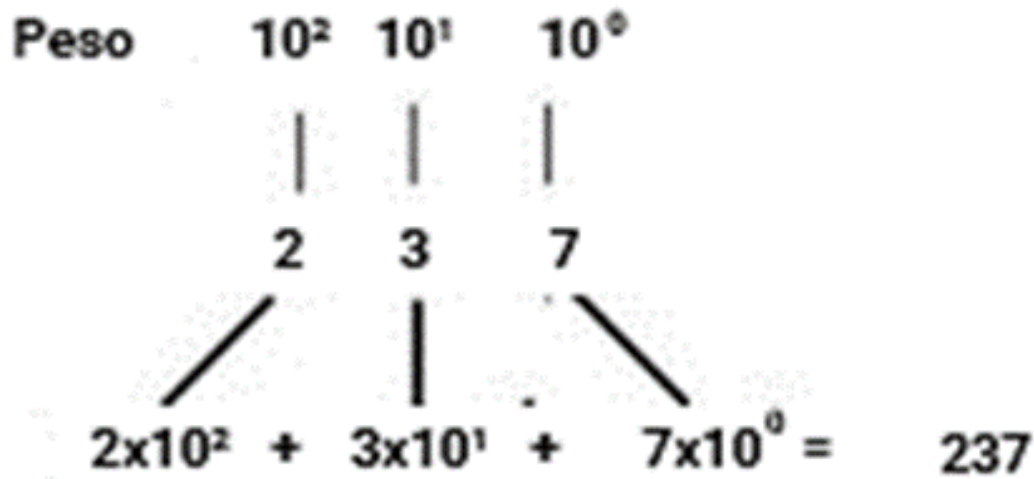
Hexadecimal (base 16)

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

O sistema decimal é o sistema comumente utilizado por nós seres humanos e é constituído por dez algarismos: **0,1,2,3,4,5,6,7,8,9.**

Sabemos que no sistema decimal cada algarismo tem um valor posicional, ou seja, cada algarismo tem um peso de acordo com a sua posição na representação de um dado valor.

Por exemplo, o número 237 na base 10 pode ser decomposto em:



O **sistema binário** é o sistema mais utilizado por máquinas atualmente uma vez que os sistemas digitais trabalham internamente com dois estados (ligado/desligado, verdadeiro/falso, aberto/fechado). O sistema binário utiliza os símbolos: **0, 1**, sendo cada símbolo designado por bit (*binary digit*). Um equipamento digital é um grande manipulador de 0's e 1's.

O **sistema octal** é um sistema de numeração de base 8, isto é, utiliza 8 símbolos (**0, 1, 2, 3, 4, 5, 6, 7**) para a representação de um determinado valor.

O **sistema hexadecimal** é muito utilizado na programação de microprocessadores (espécie de pequeno computador), especialmente nos equipamentos de estudo e sistemas de desenvolvimento. Utiliza os símbolos: **0, 1, 2, 3, 4, 5, 6, 7, 8, 9** do sistema decimal e ainda as letras **A, B, C, D, E, F**.

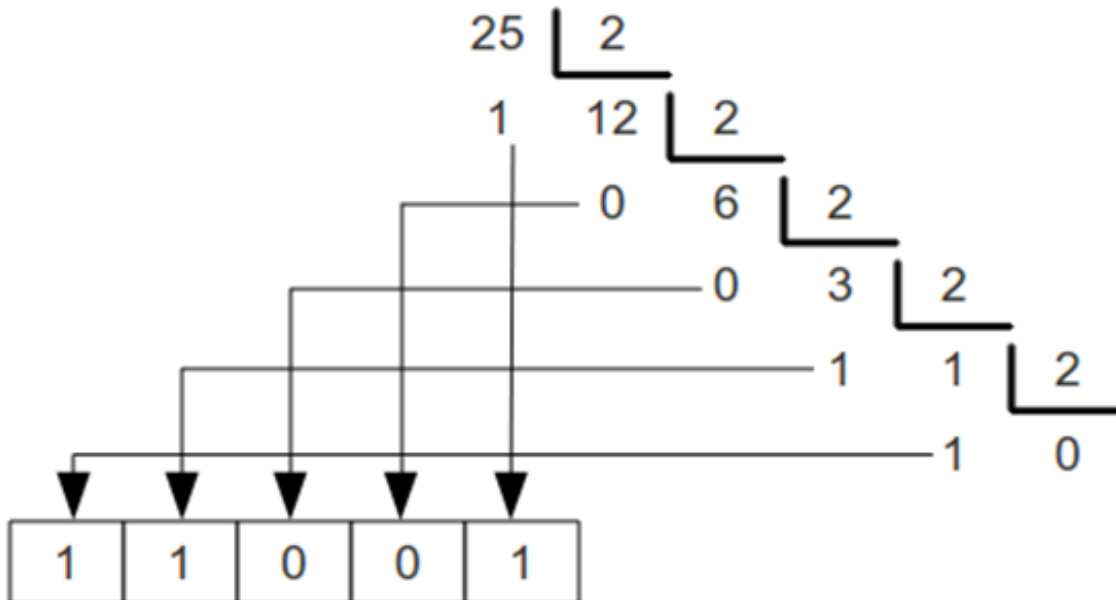
**Importante notar as equivalências:** A=10, B=11, C=12, D=13, E=14 e F=15.

## 2 – Conversões de base decimal

### Conversão de decimal para binário

Basta dividir sucessivamente por 2 o número decimal e guardar os quocientes que vão sendo obtidos, até que o quociente de uma das divisões seja 0.

O resultado é a sequência de baixo para cima de todos os restos obtidos, conforme ilustra a figura em que é apresentado 25 na base 10 que se torna 11001 na base 2.



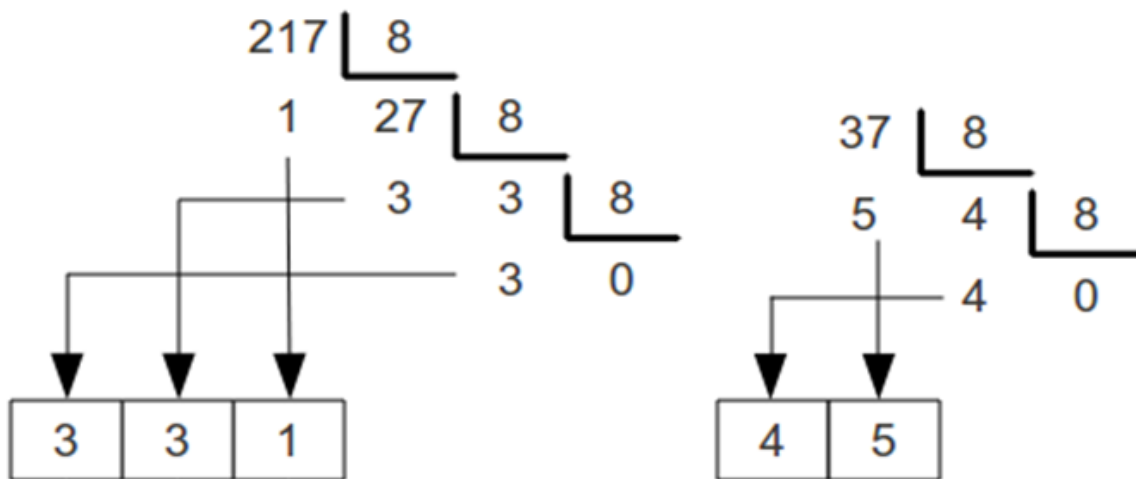
A relação entre números decimais e binários é extremamente comum e importante, dada resumidamente pela tabela abaixo para alguns valores:

Decimal	Binário		Decimal	Binário
0	0		11	1011
1	1		12	1100
2	10		13	1101
3	11		14	1110
4	100		15	1111
5	101		16	10000
6	110		17	10001
7	111		18	10010
8	1000		19	10011
9	1001		20	10100
10	1010		...	....

### Conversão de decimal para octal

Neste caso o processo de conversão se dá por divisões sucessivas por 8.

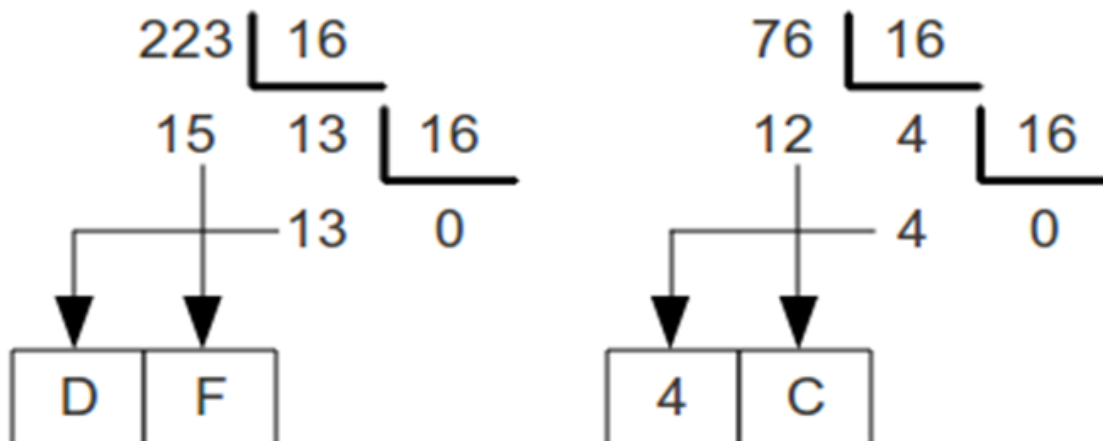
O resultado é a sequência de baixo para cima de todos os restos obtidos, conforme as imagens abaixo em que transformamos o número 217 (decimal) que vira 331 em octal e o número 37 (decimal) que vira 45 (octal)



### Conversão de decimal para hexadecimal

Desta vez o processo se dá por sucessivas divisões por 16, aproveitando-se o resto conforme abaixo.

O número 223(10) corresponde a DF(16) e o número 76(10) corresponde a 4C(16).



A tabela abaixo apresenta uma série de números em ordem crescente escritos em decimal e a representação dos mesmos na base hexadecimal:

Decimal	Hexa		Decimal	Hexa		Decimal	Hexa
0	0		11	B		22	16
1	1		12	C		23	17
2	2		13	D		24	18
3	3		14	E		25	19
4	4		15	F		26	1A
5	5		16	10		27	1B
6	6		17	11		28	1C
7	7		18	12		29	1D
8	8		19	13		30	1E
9	9		20	14		31	1F
10	A		21	15		32	20

### 3 - Conversões de base binária, octal e hexadecimal

#### Conversão de binário para decimal:

É apenas resultado da soma do dígito (0 ou 1) multiplicado pela base 2 com o expoente de acordo com sua posição, começando do zero a partir da direita.

Neste caso o número 11001 (base 2) corresponde a 25 (base 10), conforme ilustrado abaixo:

1	1	0	0	1
$1 \times 2^4$	$1 \times 2^3$	$0 \times 2^2$	$0 \times 2^1$	$1 \times 2^0$
$16 + 8 + 0 + 0 + 1 = 25$				

1	1	0	1	0	1
1	1	0	1	0	1
$1 \times 2^2$	$1 \times 2^1$	$0 \times 2^0$	$1 \times 2^2$	$0 \times 2^1$	$1 \times 2^0$
4 + 2 + 0			4 + 0 + 1		
$6_{10} = 6_8$			$5_{10} = 5_8$		
65					

	1	1	1	1	1
0	1	1	1	1	1
$0 \times 2^2$	$1 \times 2^1$	$1 \times 2^0$	$1 \times 2^2$	$1 \times 2^1$	$1 \times 2^0$
$0 + 2 + 1$			$4 + 2 + 1$		
$3_{10} = 3_8$			$7_{10} = 7_8$		
37					

### Conversão de binário para octal:

Neste caso, os dígitos do número binário são separados em grupos de 3 bits da direita para a esquerda. Cada grupo de 3 bits é um dígito em octal. Ao final, une-se os resultados.

Caso o número de dígitos do número binário não seja múltiplo de 3, completa-se os dígitos à esquerda com zeros (0).

Veja o exemplo abaixo para conversão do número 110101 (base binária) para o número 65 (base octal) e o número 1111 (base 2) para 37 (base 8).

### Binário -> hexadecimal

### Conversão de binário para hexadecimal:



Separa-se o número binário em grupos de 4 bits, da direita para a esquerda. Em seguida, transforma-se cada grupo de 4 bits em hexadecimal. Ao final, simplesmente une-se os resultados em um só. Caso o número de dígitos do número binário não seja múltiplo de 4, completa-se os dígitos à esquerda com zeros (0).

Veja o exemplo abaixo para conversão do número 01011011 (base 2 ou binária) para o número 5B (base 16 ou hexa) e o número 111101 (base 2) para 3D (base 16):

0	1	0	1	1	0	1	1
0	1	0	1	1	0	1	1
<b>0</b> x2 <sup>3</sup>	<b>1</b> x2 <sup>2</sup>	<b>0</b> x2 <sup>1</sup>	<b>1</b> x2 <sup>0</sup>	<b>1</b> x2 <sup>3</sup>	<b>0</b> x2 <sup>2</sup>	<b>1</b> x2 <sup>1</sup>	<b>1</b> x2 <sup>0</sup>
0 + 4 + 0 + 1 = 5 <sub>10</sub>				8 + 0 + 2 + 1 = 11 <sub>10</sub>			
5 <sub>10</sub> = 5 <sub>16</sub>				11 <sub>10</sub> = B <sub>16</sub>			
5B							

		1	1	1	1	0	1
0	0	1	1	1	1	0	1
$0 \times 2^3$	$0 \times 2^2$	$1 \times 2^1$	$1 \times 2^0$	$1 \times 2^3$	$1 \times 2^2$	$0 \times 2^1$	$1 \times 2^0$
$0 + 0 + 2 + 1 = 3_{10}$				$8 + 4 + 0 + 1 = 13_{10}$			
$3_{10} = 3_{16}$				$13_{10} = D_{16}$			
3D							

## 4 – Conversão de bases

### Conversão octal para decimal

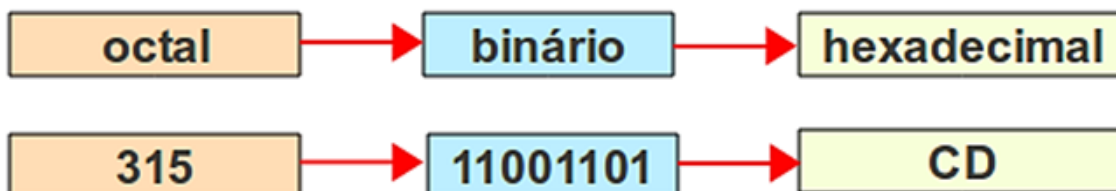
Obtida através da soma dos dígitos do número octal multiplicados pela base 8 elevada à posição colunar do dígito, começando em 0 da direita para a esquerda. Observe os exemplos para os números 331 (base 8) e 45 (base 8).

3	3	1
$3 \times 8^2$	$3 \times 8^1$	$1 \times 8^0$
$192 + 24 + 1 = 217$		

4	5
$4 \times 8^1$	$5 \times 8^0$
$32 + 5 = 37$	

### Conversão de octal para hexadecimal

Transforma-se primeiro o octal em binário e em seguida o binário em hexadecimal:



### Conversão hexadecimal para decimal

Realizada através da soma dos dígitos hexadecimais multiplicados pela base 16 elevada à posição colunar contando da direita para a esquerda, começando em 0, de forma semelhante à conversão de binários em decimais:

1	0	A
$1 \times 16^2$	$0 \times 16^1$	$10 \times 16^0$
$256 + 0 + 10 = 266$		

C	B
$12 \times 16^1$	$11 \times 16^0$
$192 + 11 = 203$	

Note que os caracteres que definem os dígitos hexadecimais A, B e C foram substituídos pelos valores equivalentes em decimais 10, 11 e 12 de acordo com a tabela da lição anterior para a realização do cálculo.

### Conversão hexadecimal para binário

Decompõem-se o número hexadecimal diretamente em binários de 4 dígitos. Os zeros mais à esquerda do resultado binário podem ser omitidos:

1				2				F			
$0 + 0 + 0 + 1 = 1$				$0 + 0 + 2 + 0 = 2$				$8 + 4 + 2 + 1 = 15$			
$0 \times 2^3$	$0 \times 2^2$	$0 \times 2^1$	$1 \times 2^0$	$0 \times 2^3$	$0 \times 2^2$	$1 \times 2^1$	$0 \times 2^0$	$1 \times 2^3$	$1 \times 2^2$	$1 \times 2^1$	$1 \times 2^0$
0	0	0	1	0	0	1	0	1	1	1	1
			1	0	0	1	0	1	1	1	1

### Conversão hexadecimal para octal

Transforma-se primeiro o hexadecimal em binário e em seguida o binário em octal:



#### 4 – Grandezas e tabela ASCII

Sempre vale a pena rever a ideia de grandezas e suas nomenclaturas:

Nome	Simbolo	Tamanho
Byte	B	8 Bit
KiloByte	KB	1024 Byte
MegaByte	MB	1024 KByte
GigaByte	GB	1024 MByte
TeraByte	TB	1024 GByte
PetaByte	PB	1024 TByte
ExaByte	EB	1024 PByte
ZettaByte	ZB	1024 EByte
YottaByte	YB	1024 ZByte

## Tabela ASCII

A tabela ASCII representa o equivalente de uma série de símbolos e caracteres presentes em inúmeros programas e aplicações:

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	<b>NUL</b> (null)	32	20	040	&#32;	<b>Space</b>	64	40	100	&#64;	<b>@</b>	96	60	140	&#96;	<b>`</b>
1	1	001	<b>SOH</b> (start of heading)	33	21	041	&#33;	<b>!</b>	65	41	101	&#65;	<b>A</b>	97	61	141	&#97;	<b>a</b>
2	2	002	<b>STX</b> (start of text)	34	22	042	&#34;	<b>"</b>	66	42	102	&#66;	<b>B</b>	98	62	142	&#98;	<b>b</b>
3	3	003	<b>ETX</b> (end of text)	35	23	043	&#35;	<b>#</b>	67	43	103	&#67;	<b>C</b>	99	63	143	&#99;	<b>c</b>
4	4	004	<b>EOT</b> (end of transmission)	36	24	044	&#36;	<b>\$</b>	68	44	104	&#68;	<b>D</b>	100	64	144	&#100;	<b>d</b>
5	5	005	<b>ENQ</b> (enquiry)	37	25	045	&#37;	<b>%</b>	69	45	105	&#69;	<b>E</b>	101	65	145	&#101;	<b>e</b>
6	6	006	<b>ACK</b> (acknowledge)	38	26	046	&#38;	<b>&amp;</b>	70	46	106	&#70;	<b>F</b>	102	66	146	&#102;	<b>f</b>
7	7	007	<b>BEL</b> (bell)	39	27	047	&#39;	<b>'</b>	71	47	107	&#71;	<b>G</b>	103	67	147	&#103;	<b>g</b>
8	8	010	<b>BS</b> (backspace)	40	28	050	&#40;	<b>(</b>	72	48	110	&#72;	<b>H</b>	104	68	150	&#104;	<b>h</b>
9	9	011	<b>TAB</b> (horizontal tab)	41	29	051	&#41;	<b>)</b>	73	49	111	&#73;	<b>I</b>	105	69	151	&#105;	<b>i</b>
10	A	012	<b>LF</b> (NL line feed, new line)	42	2A	052	&#42;	<b>*</b>	74	4A	112	&#74;	<b>J</b>	106	6A	152	&#106;	<b>j</b>
11	B	013	<b>VT</b> (vertical tab)	43	2B	053	&#43;	<b>+</b>	75	4B	113	&#75;	<b>K</b>	107	6B	153	&#107;	<b>k</b>
12	C	014	<b>FF</b> (NP form feed, new page)	44	2C	054	&#44;	<b>,</b>	76	4C	114	&#76;	<b>L</b>	108	6C	154	&#108;	<b>l</b>
13	D	015	<b>CR</b> (carriage return)	45	2D	055	&#45;	<b>-</b>	77	4D	115	&#77;	<b>M</b>	109	6D	155	&#109;	<b>m</b>
14	E	016	<b>SO</b> (shift out)	46	2E	056	&#46;	<b>.</b>	78	4E	116	&#78;	<b>N</b>	110	6E	156	&#110;	<b>n</b>
15	F	017	<b>SI</b> (shift in)	47	2F	057	&#47;	<b>/</b>	79	4F	117	&#79;	<b>O</b>	111	6F	157	&#111;	<b>o</b>
16	10	020	<b>DLE</b> (data link escape)	48	30	060	&#48;	<b>0</b>	80	50	120	&#80;	<b>P</b>	112	70	160	&#112;	<b>p</b>
17	11	021	<b>DC1</b> (device control 1)	49	31	061	&#49;	<b>1</b>	81	51	121	&#81;	<b>Q</b>	113	71	161	&#113;	<b>q</b>
18	12	022	<b>DC2</b> (device control 2)	50	32	062	&#50;	<b>2</b>	82	52	122	&#82;	<b>R</b>	114	72	162	&#114;	<b>r</b>
19	13	023	<b>DC3</b> (device control 3)	51	33	063	&#51;	<b>3</b>	83	53	123	&#83;	<b>S</b>	115	73	163	&#115;	<b>s</b>
20	14	024	<b>DC4</b> (device control 4)	52	34	064	&#52;	<b>4</b>	84	54	124	&#84;	<b>T</b>	116	74	164	&#116;	<b>t</b>
21	15	025	<b>NAK</b> (negative acknowledge)	53	35	065	&#53;	<b>5</b>	85	55	125	&#85;	<b>U</b>	117	75	165	&#117;	<b>u</b>
22	16	026	<b>SYN</b> (synchronous idle)	54	36	066	&#54;	<b>6</b>	86	56	126	&#86;	<b>V</b>	118	76	166	&#118;	<b>v</b>
23	17	027	<b>ETB</b> (end of trans. block)	55	37	067	&#55;	<b>7</b>	87	57	127	&#87;	<b>W</b>	119	77	167	&#119;	<b>w</b>
24	18	030	<b>CAN</b> (cancel)	56	38	070	&#56;	<b>8</b>	88	58	130	&#88;	<b>X</b>	120	78	170	&#120;	<b>x</b>
25	19	031	<b>EM</b> (end of medium)	57	39	071	&#57;	<b>9</b>	89	59	131	&#89;	<b>Y</b>	121	79	171	&#121;	<b>y</b>
26	1A	032	<b>SUB</b> (substitute)	58	3A	072	&#58;	<b>:</b>	90	5A	132	&#90;	<b>Z</b>	122	7A	172	&#122;	<b>z</b>
27	1B	033	<b>ESC</b> (escape)	59	3B	073	&#59;	<b>;</b>	91	5B	133	&#91;	<b>[</b>	123	7B	173	&#123;	<b>{</b>
28	1C	034	<b>FS</b> (file separator)	60	3C	074	&#60;	<b>&lt;</b>	92	5C	134	&#92;	<b>\</b>	124	7C	174	&#124;	<b> </b>
29	1D	035	<b>GS</b> (group separator)	61	3D	075	&#61;	<b>=</b>	93	5D	135	&#93;	<b>]</b>	125	7D	175	&#125;	<b>}</b>
30	1E	036	<b>RS</b> (record separator)	62	3E	076	&#62;	<b>&gt;</b>	94	5E	136	&#94;	<b>^</b>	126	7E	176	&#126;	<b>~</b>
31	1F	037	<b>US</b> (unit separator)	63	3F	077	&#63;	<b>?</b>	95	5F	137	&#95;	<b>_</b>	127	7F	177	&#127;	<b>DEL</b>

Source: [www.LookupTables.com](http://www.LookupTables.com)

## Dica

Um ponto importante a destacar é que existem inúmeras ferramentas de conversão de números entre diferentes bases. Nosso estudo até aqui teve o objetivo de mostrar como é o processo de conversão no sentido de você entender melhor o significado e a importância das bases numéricas. Uma delas é apresentada abaixo (<https://www.cjdinfo.com.br/utilitario-conversor-bases-numericas#:~:text=CJDinfo%20%2D%20Conversor%20de%20Bases%20Num%C3%A9ricas>).

**Conversor de Bases Numéricas**

Esta página faz a conversão de bases numéricas de 2 a 36, incluindo *binário*(2), *octal*(8), *decimal*(10) e *hexadecimal*(16), utilizando as rotinas genéricas apresentadas no item [Soluções](#).

Entre com a Base e o Valor de Entrada. Indique a Base de Saída e Clique em Converter.

Valores muito altos poderão provocar *Overflow* ou perda de dados na conversão.

**Base de Entrada:**

**Valor de Entrada:**

**Base de Saída:**

**Valor de Saída:**

Converter

### Referência Bibliográfica

BROOKSHEAR, J.G. **Ciência da Computação: uma visão abrangente**. Porto Alegre: Bookman, 2013.

FORBELLONE, A.L.V. & EBERSPACHER, H. F. **Lógica de Programação – A Construção de Algoritmos e Estruturas de Dados**. 3ª. Edição. São Paulo, SP: Prentice Hall, 2005.

**Ir para exercício**