# [O que é bit e byte](http://www.infowester.com/bit.php)

Os computadores "entendem" impulsos elétricos, positivos ou negativos, que são representados por 1 e 0, respectivamente. A cada impulso elétrico, damos o nome de Bit (**BI**nary digi**T**). Um conjunto de 8 bits reunidos como uma única unidade forma um **Byte**.

Na verdade, se pudéssemos entrar no computador e ver seu funcionamento, não haveria letras A, nem B, nem C, nem números, dentro do computador existe apenas **ELETRICIDADE**, e esta pode assumir apenas dois estados: LIGADO e DESLIGADO (convencionou-se que **0** representa **desligado** e **1** representa **ligado**).

Cada caractere tem um código binário associado a ele. Vamos supor que a letra **A** seja **01000001**, nenhum outro caractere terá o mesmo código. Este código de caracteres é formado pela união de 8 **“zeros”** e **“uns”**. Cada **0** e **1** é chamado de **BIT**, e o conjunto de oito deles é chamado **BYTE**. Um **BYTE** consegue armazenar apenas um **CARACTERE (letras, números, símbolos, pontuação, espaço em branco e outros caracteres especiais)**.

Para os computadores, representar 256 números binários é suficiente. Por isso, os bytes possuem 8 bits. Basta fazer os cálculos. Como um bit representa dois valores (1 ou 0) e um byte representa 8 bits, basta fazer 2 (do bit) elevado a 8 (do byte) que é igual a 256.

Os bytes representam todas as letras (maiúsculas e minúsculas), sinais de pontuação, acentos, sinais especiais e até sinais que não podemos ver, mas que servem para comandar o computador e que podem, inclusive, serem enviados pelo teclado ou por outro dispositivo de entrada de dados e instruções.

Para que isto aconteça, os computadores utilizam uma tabela que combina números binários com símbolos: a tabela **ASCII** (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange). Nesta tabela, cada byte representa um caractere ou um sinal.

A partir daí, foram criados vários termos para facilitar a compreensão humana da capacidade de armazenamento, processamento e manipulação de dados nos computadores. No que se refere aos bits e bytes, tem-se as seguintes medidas:

**1 Byte = 8 bits**

**1 Kilobyte (ou KB) = 1024 bytes**

**1 Megabyte (ou MB) = 1024 kilobytes**

**1 Gigabyte (ou GB) = 1024 megabytes**

**1 Terabyte (ou TB) = 1024 gigabytes**

**1 Petabyte (ou PB) = 1024 terabytes**

**1 Exabyte (ou EB) = 1024 petabytes**

**1 Zettabyte (ou ZB) = 1024 exabytes**

**1 Yottabyte (ou YB) = 1024 zettabytes**

Calculadora que converte grandezas de Bytes

http://www.gwebtools.com.br/converter-bit

É também através dos bytes que se determina o comprimento da palavra de um computador, ou seja, a quantidade de bits que ele utiliza na composição das instruções internas, como por exemplo:

**8 bits - palavra de 1 byte**

**16 bits - palavra de 2 bytes**

**32 bits - palavra de 4 bytes**

Na transmissão de dados entre computadores, geralmente usa-se medições relacionadas a bits e não a bytes. Assim, existem também os seguintes termos:

**1 Kilobit (ou Kb) = 1024 bits**

**1 Megabit (ou Mb) = 1024 Kilobits**

**1 Gigabit ou (Gb) = 1024 Megabits**

[Note que quando a medição é feita em bytes, o B da sigla é maiúsculo (como em GB). Quando a medição é feita em bits, o B da sigla fica em minúsculo (como em Gb).](http://www.infowester.com/bit.php)

**Números decimais.** A maneira mais fácil de se compreender os bits é compará-los a algo que você já conhece: os **dígitos**. Um dígito é um local que pode conter valores numéricos entre 0 e 9. Dígitos normalmente são combinados em grupos para criar números maiores. Por exemplo, 6.357 possui quatro dígitos. Sabe-se que, no número 6.357, o 7 ocupa a posição de unidade, enquanto o 5 ocupa a posição de dezena, o 3 ocupa a posição de centena e o 6 ocupa a posição de milhar. Assim, caso queira ser explícito, poderá expressar esse número da seguinte maneira:

(6 \* 1000) + (3 \* 100) + (5 \* 10) + (7 \* 1) = 6000 + 300 + 50 + 7 = 6357

Uma outra maneira de expressá-lo seria utilizando **potências de 10**. Suponhamos que o conceito de "elevado à potência de" seja representado pelo símbolo "^" ("10 ao quadrado" seria escrito como "10^2"). Assim uma outra maneira de expressar esse número é:

(6 \* 10^3) + (3 \* 10^2) + (5 \* 10^1) + (7 \* 10^0) = 6000 + 300 + 50 + 7 = 6357

O que se pode perceber nessa expressão é que cada dígito é um **marcador de posição** para a próxima potência de 10, começando no primeiro dígito com 10 elevado à potência de zero.

Isso deve ser considerado cômodo, já que trabalhamos com dígitos decimais todos os dias. Mas o interessante sobre sistemas numéricos é que não existe nada que o force a ter 10 valores diferentes em um dígito. Nosso **sistema decimal** provavelmente se desenvolveu porque possuímos 10 dedos - caso viéssemos a [evoluir](http://informatica.hsw.uol.com.br/evolucao.htm) para apenas oito, poderíamos ter um sistema baseado na mesma quantidade de dígitos. Você pode criar sistemas baseados em qualquer quantidade de dígitos. Na verdade, existem várias boas razões para utilizar diferentes bases em diferentes situações.

**Bits**

Os computadores operam utilizando o sistema numérico baseado em 2 dígitos, também conhecido como **sistema numérico binário,** assim como o sistema numérico baseado em 10 dígitos é conhecido como sistema numérico decimal. A razão pela qual os computadores utilizam o sistema baseado em 2 dígitos é que isso torna muito mais fácil implementá-los com a tecnologia eletrônica atual. É possível conectar e montar computadores que operam na base de 10 dígitos, mas eles seriam extremamente caros. Por outro lado, os computadores binários são relativamente baratos.

Por isso os computadores usam números binários e, conseqüentemente, **dígitos binários** no lugar de dígitos decimais. A palavra **bit** é a abreviação das palavras "Binary digIT" (dígito binário). Enquanto os dígitos decimais possuem 10 valores possíveis, que vão de 0 a 9, os bits possuem apenas dois: 0 e 1. Portanto, um número binário é composto apenas de 0s e 1s, como por exemplo: 1011. De que maneira se descobre qual é o valor do número binário 1011? Você o faz da mesma forma que fizemos anteriormente para 6357, mas utilizará a base de 2 dígitos ao invés de 10. Assim:

(1 \* 2^3) + (0 \* 2^2) + (1 \* 2^1) + (1 \* 2^0) = 8 + 0 + 2 + 1 = 11

Você pode observar que em números binários cada bit comporta o valor das potências crescentes de 2. Isso torna a contagem em binários consideravelmente fácil. Contando em decimais e binários, começando em zero e indo até 20, fica assim:

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

Observando essa seqüência, você percebe que 0 e 1 são os mesmos para os sistemas numéricos decimal e binário. No número 2, no entanto, observa a primeira repetição no sistema binário. Se um bit é 1, e você soma 1 a ele, o bit torna-se 0 e o próximo torna-se 1. Na transição de 15 para 16 esse efeito passa por 4 bits, transformando 1111 em 10000.

**Bytes**

Os bits dificilmente estão sozinhos nos computadores. Normalmente são agrupados em conjuntos de 8 bits, chamados **bytes**. Por que existem 8 bits em um byte? Seria o mesmo que perguntar: "por que há 12 ovos em uma dúzia". O byte de 8 bits é algo que as pessoas estabeleceram através de tentativas e erros durante os últimos 50 anos.

Com 8 bits em um byte é possível representar 256 valores, de 0 a 255, como mostrado abaixo:

0 = 00000000  
 1 = 00000001  
 2 = 00000010  
 ...  
 254 = 11111110  
 255 = 11111111

**Bytes: o padrão ASCII**

Os bytes são freqüentemente utilizados para comportar caracteres individuais em um documento de texto. No **sistema de caracteres ASCII**, cada valor binário entre 0 e 127 está associado a um caractere específico. A maioria dos computadores estende o sistema ASCII, para utilizar a gama completa dos 256 caracteres disponíveis em um byte. Os últimos 128 caracteres comportam elementos especiais, como caracteres acentuados de diversas línguas como o português.

Você pode ver os 127 códigos padrão da tabela ASCII, abaixo. Computadores armazenam documentos de texto, tanto no [disco](http://informatica.hsw.uol.com.br/discos-rigidos.htm) quanto na [memória](http://informatica.hsw.uol.com.br/memoria-do-computador.htm), utilizando esses códigos. Se você utiliza o Bloco de Notas do Windows 95/98 para criar um arquivo de texto contendo as palavras "Four score and seven years ago" (quatro pontos e sete anos atrás), o Bloco de Notas usaria 1 byte de memória por caractere (incluindo 1 byte para cada espaço entre as palavras, caractere 32 da tabela ASCII). Quando o Bloco de Notas armazena a sentença em um arquivo no disco, o arquivo também irá conter 1 byte por caractere e por espaço.

Faça esse experimento: abra um novo arquivo no Bloco de Notas e insira a frase "Four score and seven years ago". Salve o arquivo no disco com o nome de **getty.txt**. Utilize então o Explorer e veja o tamanho do arquivo. Você irá descobrir que o arquivo ocupa um espaço de 30 bytes no disco: 1 byte para cada caractere. Se você adicionar outra palavra ao final da sentença e salvar novamente, o tamanho do arquivo irá subir para o número referente de bytes. Cada caractere consome um byte.

Se você olhasse para um arquivo como um computador o faz, veria que cada byte contém um número, não uma letra; o número é o código ASCII que corresponde ao caractere (veja abaixo). Desse modo, os números para o arquivo no disco são:

F o u r a n d s e v e n  
 70 111 117 114 32 97 110 100 32 115 101 118 101 111

Olhando a tabela ASCII, você pode observar uma correspondência entre cada caractere e o código ASCII utilizado. Observe o uso do 32 para espaço - 32 é o código ASCII para espaço. Poderíamos expandir esses números decimais para números binários (assim, 32=00100000), se quiséssemos ser tecnicamente corretos. Na realidade, é assim que o computador lida com as coisas.

**Tabela ASCII padrão**  
Os 32 primeiros valores (0 até 31) são códigos para itens como o retorno para o início de uma linha (CR - Carriage Return) e o avanço de linha (LF - Line Feed). O caractere de espaço é o 33º valor, seguido de caracteres de pontuação, dígitos, caracteres maiúsculos e caracteres minúsculos.

0 NUL  
 1 SOH  
 2 STX  
 3 ETX  
 4 EOT  
 5 ENQ  
 6 ACK  
 7 BEL  
 8 BS  
 9 TAB  
 10 LF  
 11 VT  
 12 FF  
 13 CR  
 14 SO  
 15 SI  
 16 DLE  
 17 DC1  
 18 DC2  
 19 DC3  
 20 DC4  
 21 NAK  
 22 SYN  
 23 ETB  
 24 CAN  
 25 EM  
 26 SUB  
 27 ESC  
 28 FS  
 29 GS  
 30 RS  
 31 US  
 **32** 33 !  
 34 "  
 35 #  
 36 $  
 37 %  
 38 &  
 39 '  
 40 (  
 41 )  
 42 \*  
 43 +  
 44 ,  
 45 -  
 46 .  
 47 /  
 48 0  
 49 1  
 50 2  
 51 3  
 52 4  
 53 5  
 54 6  
 55 7  
 56 8  
 57 9  
 58 :  
 59 ;  
 60 <  
 61 =  
 62 >  
 63 ?  
 64 @  
 65 A  
 66 B  
 67 C  
 68 D  
 69 E  
 70 F  
 71 G  
 72 H  
 73 I  
 74 J  
 75 K  
 76 L  
 77 M  
 78 N  
 79 O  
 80 P  
 81 Q  
 82 R  
 83 S  
 84 T  
 85 U  
 86 V  
 87 W  
 88 X  
 89 Y  
 90 Z  
 91 [  
 92   
 93 ]  
 94 ^  
 95 \_  
 96 `  
 97 a  
 98 b  
 99 c  
 100 d  
 101 e  
 102 f  
 103 g  
 104 h  
 105 i  
 106 j  
 107 k  
 108 l  
 109 m  
 110 n  
 111 o  
 112 p  
 113 q  
 114 r  
 115 s  
 116 t  
 117 u  
 118 v  
 119 w  
 120 x  
 121 y  
 122 z  
 123 {  
 124 |  
 125 }  
 126 ~  
 127 DEL

**Unidades de transmissão em redes computacionais**

Pois bem, em uma medida tradicional de armazenamento, a união de 8 bits formará 1 Byte (pronuncia-se “bait”). A partir disso, temos os seguintes valores:

1.024 Bytes – 1 Kilobyte (KB)  
1.024 Kilobytes – 1 Megabyte (MB)  
1.024 Megabytes – 1 Gigabyte (GB)  
…

Essa forma de medida é empregada principalmente no que diz respeito ao armazenamento em discos rígidos, memórias RAM, além de quaisquer outras mídias, sejam elas removíveis ou não (tais como pendrives, CDs, DVDs, cartões de memória, etc).

No caso de Kilobits e Megabits, além da utilização do “b” minúsculo em suas abreviações, a diferenciação ocorre desde a união do número de bits iniciais. O byte, união de 8 bits, deixa de existir, e os cálculos são realizados com bases de 1.000 para 1 (nos sistemas de armazenamento, a base é de 1.024 para 1). Veja abaixo:

1.000 Bits – 1 Kilobit (Kb)  
1.000 Kilobits – 1 Megabit (Mb)  
1.000 Megabits – 1 Gigabit (Gb)  
…

As medidas baseadas nesse formato são utilizadas principalmente para avaliar o tráfego em redes de computadores, agregando suas abreviações à terminação “ps” (por segundo). Com isso, surgem as conhecidas medidas de transmissão mencionadas anteriormente: Kbps (Kilobits por segundo) e Mbps (Megabits por segundo).

**Velocidade “real” de sua internet**

Tendo como exemplo uma [conexão](http://www.dinx.com.br/2009/08/unidades-de-transmissao-em-redes-computacionais/) de 800 Kbps com a internet, poderíamos dividir o valor por 8 para chegarmos a um valor aproximado da velocidade “real” de download (mais ou menos 100 KB por segundo). Para descobrirmos o valor exato, teríamos que levar em consideração as bases diferentes (1.024 para 1 nos meios de armazenamento e 1.000 para 1 na transmissão de dados). Isso pode ser realizado com algumas simples regras:

**1. Primeiramente, calculamos o valor em bits da conexão:**

800 Kbps = 800 Kbps x 1.000 = 800.000 bits

**2. Em seguida, calculamos o valor em bits de 1 KB:**

1 KB = 1.024 Bytes x 8 = 8.192 bits

**3. Dividimos o valor da conexão pelo valor do KB (em bits)**

800.000 / 8.192 = 97,65 KBps

Ou seja, em uma conexão real de 800 Kbps, o internauta poderia realizar um download em uma velocidade máxima de 97,65 Kilobytes por segundo.

**Calcule o tempo estimado de seus downloads**

Supondo então que gostaríamos de saber o tempo estimado para realizar o download de um arquivo de 50 MB utilizando a mesma conexão do exemplo. Para isso, basta partirmos do valor encontrado nos cálculos anteriores:

**1. Multiplicamos o tamanho do arquivo (50 MB) por 1.024, obtendo assim seu valor em KB:**

50 MB x 1.024 = 51.200 KB

**2. Agora, dividimos o tamanho em KB do arquivo pela velocidade “real” encontrada anteriormente, obtendo assim o total de tempo em segundos:**

51.200 KB / 97,65 KBps = 524,32 segundos

**3. Por último, dividimos o total em segundos por 60, para obtermos o tempo estimado em minutos:**

524,32 segundos / 60 = 8,73 minutos (aproximadamente 8 minutos e 43 segundos)

Veja que, nos exemplos apresentados, 97,65 KBps seria o máximo possível em uma conexão de 800Kbps, ao menos em teoria. Mas devido a diversos outros fatores que influenciam nos serviços disponíveis atualmente, dificilmente esse valor é alcançado.

Exercícios de bits e Bytes

1. O que é bit?
2. Como é representado?
3. Como é formata as letras de um computador?
4. Quantos bits tem em 1 Byte?
5. Para que Serve a tabela ASCII
6. Quais os estados dos Bits 1 e o 0 ?

7- Quantos bytes eu tenho em 4 MB

8- Quantos Kylobytes eu tenho em 11 GB?

9- Quantos Gigabytes eu tenho em 34 MB?

10- Quantos Terabytes eu tenho em 557 GB?

11- quantos Megabytes eu tenho em 98 TB?

12- Quantos bytes eu tenho em 87 KB?

13- Quantos Zetabytes tenho em 5 YB?

14- Quantos Pettabytes tenho em 3 ZB?

15- Quantos bits tenho em 3 KB?

16- Quantos Terabytes tenho em 2 PB?

17- Quantas músicas com 3 MB cabem em um pen drive de 16 GB?

18- Quantos arquivos de 256 KB cabem em um pen drive de 1 GB?

19- eu tenho um HD com 20 TB, quantos jogos de 750 MB cabem nele?

20- tenho um celular com um cartão de memória de 2 GB, quantas musicas de 5 MB cabem?

21- Tenho uma maquina fotográfica com um cartão de 8 GB, quantos fotos de 780 KB cabem?

22- Em um servidor com 100 TB de espaço físico de armazenamentos cabem quantos arquivos de 1,5 GB?

23- Em um MP3 player com a capacidade de 10 GB cabem quantas músicas de 4 MB?

24- Tenho uma mídia de HD-DVD de 80 GB quantos cds de 780 MB cabem dentro do HD-DVD?

26- Em um zipdrive com 120 MB cabem quantos arquivos de 108 KB?

27- Quantas fotos de baixa resolução com 150KB cabem em uma maquina com a capacidade de 4 GB?

28- Em quanto tempo eu baixo um arquivo de 200 MB com uma conexão de 2 Mbps?

29- Em quanto tempo eu baixo um arquivo de 30 MB com uma conexão de 512Kbps?

30- Em quanto tempo eu baixo um arquivo de 2 TB com uma conexão de 5 Mbps?