



## DESCRIÇÃO

Organização e arquitetura de computadores. Utilização dos sistemas de numeração e representação de dados.

## PROPÓSITO

Introduzir os fundamentos básicos da representação de dados desde o *bit* (a menor unidade de representação dos dados em um sistema de computação) até os caracteres (apresentados em aplicações, em um nível de abstração mais próximo do usuário).

## OBJETIVOS

### MÓDULO 1

Identificar as unidades de informação utilizadas pelos sistemas de computação

### MÓDULO 2

Descrever os sistemas de numeração a partir da prática de operações aritméticas

### MÓDULO 3

Empregar a conversão entre os sistemas de numeração

### MÓDULO 4

# MÓDULO 1

---

## 🕒 Identificar as unidades de informação utilizadas pelos sistemas de computação

## INTRODUÇÃO

É muito importante dominar conceitos fundamentais que determinam qualquer área de conhecimento. Um dos conceitos fundamentais da computação está relacionado ao termo *bit*.

O computador é uma máquina concebida a partir de componentes eletrônicos. De um modo geral, o *bit* possibilita representar os sinais elétricos em informações que possuem significados úteis para a criação e o processamento de tarefas.

## UNIDADES DE INFORMAÇÃO

### VOCÊ JÁ COMPAROU COMO SÃO REALIZADOS OS PROCESSOS DE COMUNICAÇÃO ENTRE O COMPUTADOR E O SER HUMANO?

As pessoas normalmente podem criar textos formados por frases, as quais, por sua vez, utilizam palavras compostas por letras (caracteres).

Da mesma forma, um sistema de computação precisa usar representações simbólicas para que o processamento dos dados seja realizado corretamente e em conformidade com as expectativas de seus projetistas e usuários.



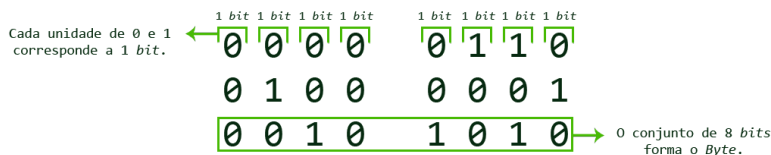
Foto: Shutterstock.com

Demonstraremos, a seguir, como as unidades de informação são fundamentais para a realização desse processo.

O computador armazena e move os dados eletronicamente sob a forma de voltagem ou corrente elétrica. A representação dos valores elétricos é feita na forma binária; portanto, ela utiliza somente dois valores: **0 e 1**.

Quando os *bits* são agrupados em um conjunto ordenado com **oito valores**, temos a unidade de medida denominada **Byte**.

Você pode ver no exemplo a seguir várias sequências de 8 *bits*, que formam o *Byte*. Essas sequências são entendidas pelo computador e decodificadas, dando origem ao caractere.



computersciencewiki.org

Por isso, a menor unidade de informação possível em um sistema de computação é o ***bit***.

## VAMOS ENTENDER COMO É FEITA A CONVERSÃO ENTRE ***BIT*** E ***BYTE***?

Sabemos que 8 *bits* = 1 *Byte*. Então, como convertemos 512 *bits* em *Bytes*?

Fazendo uma regra de três simples:

8 *bits* --- 1 *Byte*

512 *bits* --- x

$x = 512/8 = 64$  *Bytes*

Desse modo, a regra geral para a conversão é:

**De *bit* para *Byte***

**Dividimos** o valor por 8

**De *Byte* para *bit***

**Multiplicamos** o valor por 8

Apontaremos, a seguir, quatro conjuntos formados nas unidades de informação:

Um conjunto ordenado de *Bytes*, que representa uma informação útil para os computadores, constitui uma **palavra**.

Um conjunto estruturado de palavras forma um **registro**.

Um conjunto organizado de registros forma um **arquivo**.

Um conjunto organizado de arquivos forma um **banco de dados**.

Da mesma forma que fazemos no nosso cotidiano, podemos empregar **múltiplos e submúltiplos** para grandezas nos sistemas de unidades a fim de exprimir valores de grandeza maiores ou menores que uma unidade.

Por exemplo, para medir a grandeza **comprimento**, é usada, no Sistema Internacional de medidas, a unidade de base **metro**.



## EXEMPLO

Medido na Via Dutra, o comprimento entre Rio de Janeiro e São Paulo é de aproximadamente 434.000 metros.

Esse valor pode ser abreviado usando o prefixo **quilo**, que é múltiplo do metro no fator de  $10^3$  (1.000).

Assim, dizemos que 434 km (quilômetros) é igual a  $434 \times 1.000 = 434.000$  metros.

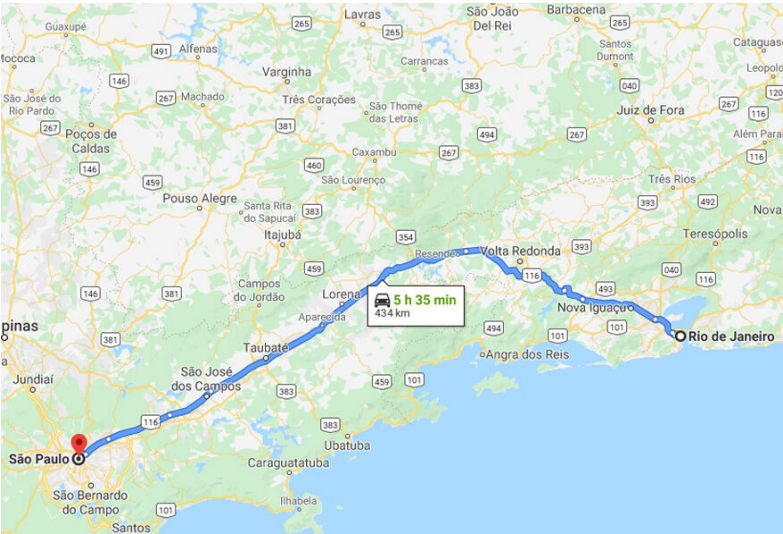


Imagem: Fabio Henrique Silva



Também podemos utilizar prefixos de múltiplos e submúltiplos das unidades *bit* e *Byte*. Em geral, a abreviação simbólica do *bit* é realizada com o “b” minúsculo e a do *Byte*, com o “B” maiúsculo.

Observe estas representações:

4.000 Kb ( $4.000 \times 10^3 = 4.000.000 \text{ bits}$ )

567 MB ( $567 \times 10^6 = 567.000.000 \text{ Bytes}$ )

## 📢 ATENÇÃO

No exemplo, a representação dos múltiplos e submúltiplos foi realizada com uma base numérica decimal, que possui 10 números (de 0 a 9), já que a utilizamos normalmente no nosso dia a dia. Entretanto, seus resultados expressam valores aproximados (e não exatos) da quantidade de *bits* ou *Bytes* medidos em um computador.

Como vimos, o *bit* é a menor unidade de informação dos **computadores atuais**. Como só possui dois valores possíveis (0 e 1), ele é representado pela base numérica **binária**.

Se reescrevermos o exemplo anterior representando os valores decorrentes do cálculo com a base 2, obteremos:

4.000 Kb ( $4.000 \times 2^{10} = 4.096.000 \text{ bits}$ )

567 MB ( $567 \times 2^{20} = 594.542.592 \text{ Bytes}$ )

A tabela a seguir mostra alguns prefixos empregados na computação a fim de abreviar valores medidos em uma unidade base (como, por exemplo, *bit* ou *Byte*) para valores em potências de 2 (binário) e de 10 (decimal):

Unidade	Valor em potência de 2	Valor unitário	Valor em potência de 10	Valor unitário
1 K (quilo)	$2^{10}$	1024	$10^3$	1.000
1 M (mega)	$2^{20}$	1.048.576	$10^6$	1.000.000
1 G (giga)	$2^{30}$	1.073.741.824	$10^9$	1.000.000.000

Unidade	Valor em potência de 2	Valor unitário	Valor em potência de 10	Valor unitário
1 T (tera)	$2^{40}$	1.099.511.627.776	$10^{12}$	1.000.000.000.000

Esses são os prefixos tipicamente adotados na computação com unidades de medida para exprimir valores de grandezas muito maiores ou menores que aqueles da unidade usada sem um prefixo.

## 📢 ATENÇÃO

Os valores unitários podem ser obtidos a partir da notação dos múltiplos da grandeza em:

**Potência de 2** - Expressam os valores (em decimal) que, de fato, são manipulados pelo computador em binário.

**Potência de 10** - Exprimem os valores (em decimal) equivalentes àqueles manipulados pelo computador em binário, embora empreguem os múltiplos no sistema decimal comumente utilizado no nosso dia a dia.

Suponha que você queira mudar o prefixo do valor a seguir:

**52,9 GB → TB?**

Usando como referência os valores mostrados na tabela anterior, uma maneira de realizar isso é:

1

Escrever os prefixos em sua sequência (B, KB, MB etc.).

2

Colocar o valor (no caso, 52,9) embaixo da unidade com o prefixo dado. Este será seu ponto de partida ou ponto de referência relacionado ao novo valor obtido.

3

Desenhar uma **seta** no sentido do prefixo pretendido. Do lado esquerdo dela, fica o operador matemático "+"; do direito, o sinal "-". O sinal resultante fica ao lado do valor do expoente da base utilizada.

## SETA

O sentido da seta pode ter duas direções:

**Esquerda** - Determina o múltiplo do valor (expoente positivo);

**Direita** - Indica o submúltiplo do valor (expoente negativo).

Assim, você terá:



Conforme os valores mostrados na tabela anterior, observemos sua aplicação com múltiplos em:

#### Potência de 10

$$52,9 \times 10^{-3} = 0,0529 \text{ TB}$$

Isso ocorrerá se considerarmos os múltiplos em potência de 10.

#### Potência de 2

$$52,9 \times 2^{-10} \text{ TB} \sim 0,05166$$

### DICA

Como o cálculo do valor na potência de 2 não é tão “trivial” quanto o cálculo do valor na potência de 10, uma sugestão é deixá-lo no formato de notação científica. Se, ainda assim, você quiser realizá-lo, ele será de aproximadamente **0,05166 TB**.

## ENTRE DOIS PREFIXOS DE UNIDADES DE MEDIDA VIZINHOS TIPICAMENTE UTILIZADOS NA COMPUTAÇÃO, O EXPOENTE DA BASE DECIMAL VARIA, EM GERAL, DE 3 EM 3 (OU DE 10 EM 10, SE A BASE FOR BINÁRIA).

Assim, por exemplo, entre Giga (G) e Mega (M), o valor para a potência de 10 é  $10^3$ , e para a potência de 2 é  $2^{10}$ .

Do mesmo modo, entre Tera (T) e Giga (G), permanecem os valores para as potências de 10 e 2:  $10^3$  e  $2^{10}$ .

Já entre Giga (G) e Kilo (K), registramos o seguinte valor:  $10^3 \times 10^3 = 10^6$ , ou  $2^{10} \times 2^{10} = 2^{20}$ .

Esta imagem ilustra alguns exemplos:

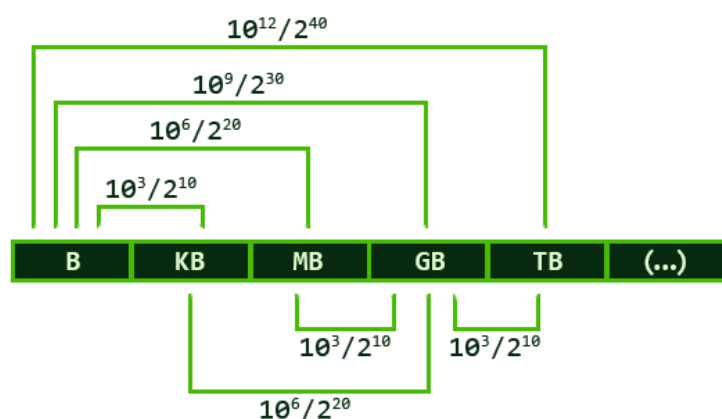


Imagem: Fabio Henrique Silva

Suponhamos que você deseje novamente mudar o prefixo do valor a seguir:

## 143 GB → KB?

Para isso, você deverá fazer o seguinte:

B	KB	MB	GB	TB	(...)
	?		143		
		$10^6$ $2^{20}$			
		←			

Imagem: Fabio Henrique Silva

Vejamos, por fim, os cálculos em potência de:

**10**

$$143 \times 10^6 = 143.000.000 \text{ KB}$$

**2**

$$143 \times 2^{20} = 149.946.368 \text{ KB}$$

## REPRESENTAÇÃO DAS INFORMAÇÕES NO COMPUTADOR

O primeiro passo **indispensável** para evoluir em qualquer área de atuação pretendida no ramo da computação é entender como um computador funciona, verificando como as informações são representadas dentro dele.

“

**TUDO O QUE É ADEQUADO PARA PROCESSAMENTO COM COMPUTADORES DIGITAIS SERÁ REPRESENTADO (CODIFICADO) COMO UMA SEQUÊNCIA DE 0S E 1S, SEJAM DADOS NUMÉRICOS, TEXTO, ARQUIVOS EXECUTÁVEIS, SEJAM IMAGENS, ÁUDIO OU VÍDEO.”**

(SEdgeWICK; WAYNE, 2017)

Por exemplo, para cada tipo de dado, serão amplamente utilizados os seguintes métodos padrão de codificação:



Imagem: Shutterstock.com



Imagem: Shutterstock.com

### **Padrão ASCII**

O padrão ASCII pode associar um número binário de 7 *bits* a cada um dos 128 caracteres distintos possíveis.



Imagem: Shutterstock.com

### **Formato de arquivo MP3**

O formato de arquivo MP3 especifica o modo de codificar um arquivo de áudio como uma sequência de 0s e 1s.



Imagem: Shutterstock.com

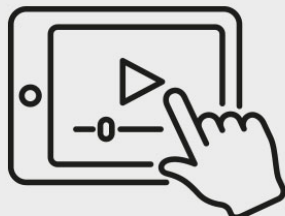


### Formato de imagem .png

O formato de imagem .png especifica os *pixels* das imagens digitais como uma sequência de 0s e 1s.

Neste vídeo, o professor Fabio Henrique Silva reforça alguns conceitos da Representação de Dados e pratica a conversão entre as unidades de medida.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



## MÃO NA MASSA

**1. CONSIDEREMOS QUE PREFIXOS POSSAM SER ADICIONADOS A UMA UNIDADE PARA INDICAR MÚLTIPLOS E SUBMÚLTIPLOS DOS SEUS VALORES. TENDO COMO PONTO DE PARTIDA A PRÓPRIA UNIDADE (SEM PREFIXO, COMO, POR EXEMPLO, O *BIT* ), O VALOR  $2^{20}$ , QUE ESTÁ EM POTÊNCIA DE 2, EQUIVALE A QUAL PREFIXO USADO PARA, NESSE CASO, ABREVIAR VALORES EM COMPUTAÇÃO?**

- A) Quilo
- B) Mega
- C) Giga
- D) Tera

---

### GABARITO

**1. Consideremos que prefixos possam ser adicionados a uma unidade para indicar múltiplos e submúltiplos dos seus valores. Tendo como ponto de partida a própria unidade (sem prefixo, como, por exemplo, o *bit* ), o valor  $2^{20}$ , que está em potência de 2, equivale a qual prefixo usado para, nesse caso, abreviar valores em computação?**

A alternativa "**B**" está correta.

A tabela a seguir lhe ajudará a memorizar os valores e as unidades equivalentes em relação a uma unidade base de medida:

Unidade	Valor em potência de 2	Valor em potência de 10
1 K (quilo)	$2^{10}$	$10^3$
1 M (mega)	$2^{20}$	$10^6$
1 G (giga)	$2^{30}$	$10^9$
1 T (tera)	$2^{40}$	$10^{12}$

# VERIFICANDO O APRENDIZADO

1. A QUANTOS *BITS* EQUIVALEM 256 *BYTES* ?

- A) 32 *bits*
- B) 256 *bits*
- C) 1024 *bits*
- D) 2048 *bits*

2. JÚLIO ESTÁ ESCRREVENDO UM RELATÓRIO EM DETERMINADO APLICATIVO DE EDIÇÃO DE TEXTO. AO TENTAR SALVÁ-LO, ELE REPARA QUE O ARQUIVO POSSUI O TAMANHO DE 2.456 *BITS* . COMO ESSE VALOR PODE SER ABREVIADO?

- A) 2.456 B
- B) 2.456 KB
- C) 2.456 b
- D) 2.456 Kb

3. FUNCIONÁRIA DE UMA COMPANHIA DE SEGUROS, MANOELA É USUÁRIA DE UM TIPO DE PROGRAMA FEITO PARA GERENCIAR UM CONJUNTO ORGANIZADO DE ARQUIVOS. ASSINALE A ALTERNATIVA CUJO TERMO REPRESENTA A CONSTITUIÇÃO DESSE CONJUNTO:

- A) Banco de dados
- B) Palavra
- C) Registro
- D) Arquivo

4. PASSE O NÚMERO A SEGUIR PARA O PREFIXO DE MÚLTIPLO OU SUBMÚLTIPLO DA UNIDADE INDICADA:

0,876 MB → TB

SEU RESULTADO É:

- A) 0,000000876 TB
- B) 0,00876 TB
- C) 876 TB
- D) 876.000 TB

5. A QUANTOS *BYTES* EQUIVALEM 128 *BITS* ?

- A) 128 Bytes
- B) 16 Bytes
- C) 1.024 Bytes
- D) 8 Bytes

6. PASSE O NÚMERO A SEGUIR PARA O PREFIXO DE MÚLTIPLO OU SUBMÚLTIPLO DA UNIDADE INDICADA:

7,8 TB → GB

SEU RESULTADO É:

- A) 0,0078 GB
- B) 0,780 GB
- C) 780 GB
- D) 7800 GB

---

## GABARITO

1. A quantos *bits* equivalem 256 Bytes ?

A alternativa "D " está correta.

Considerando que 8 *bits* é igual a 1 Byte e fazendo uma regra de três simples, temos o seguinte resultado:

8 *bits* --- 1 Byte

x --- 256 Bytes

$x = 256 * 8 = 2.048 \text{ bits}$

2. Júlio está escrevendo um relatório em determinado aplicativo de edição de texto. Ao tentar salvá-lo, ele repara que o arquivo possui o tamanho de 2.456 *bits* . Como esse valor pode ser abreviado?

A alternativa "D " está correta.

Em geral, a abreviação simbólica do *bit* é realizada com o “b” minúsculo e a do Byte , com o “B” maiúsculo.

3. Funcionária de uma companhia de seguros, Manoela é usuária de um tipo de programa feito para gerenciar um conjunto organizado de arquivos. Assinale a alternativa cujo termo representa a constituição desse conjunto:

A alternativa "A " está correta.

Banco de dados é um conjunto organizado de arquivos.

4. Passe o número a seguir para o prefixo de múltiplo ou submúltiplo da unidade indicada:

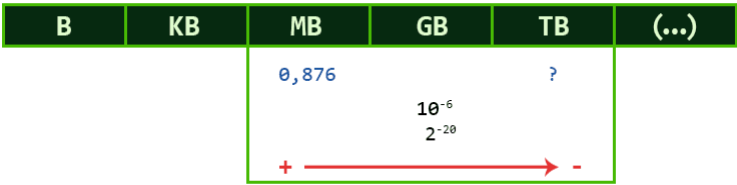
0,876 MB → TB

Seu resultado é:

A alternativa "A " está correta.

Note que estamos considerando os múltiplos em potências de 10. Desse modo, devemos fazer o cálculo segundo os valores mostrados na tabela que estudamos de prefixos adotados na computação:  $0,876 \times 10^{-6} = 0,000000876$  TB.

Além disso, devemos escrever os prefixos em sequência e colocar o valor de partida embaixo do prefixo dado, desenhando a seta no sentido do prefixo desejado. Se o sentido dela for para a esquerda, multiplicaremos o valor fornecido pela base elevada ao expoente positivo. Caso ela se volte para a direita, a multiplicação desse valor será feita com o expoente negativo. Veja:



5. A quantos Bytes equivalem 128 bits ?

A alternativa "B " está correta.

Sabemos que 8 bits = 1 Byte . Fazendo uma regra de três simples, obtemos este resultado:

8 bits --- 1 Byte  
128 bits --- x  
x = 128/8 = 16 Bytes

6. Passe o número a seguir para o prefixo de múltiplo ou submúltiplo da unidade indicada:

7,8 TB → GB

Seu resultado é:

A alternativa "D " está correta.

Note que estamos considerando os múltiplos em potências de 10. Desse modo, devemos fazer o cálculo segundo os valores mostrados na tabela que estudamos de prefixos adotados na computação:  $7,8 \times 10^3 = 7800$  GB

Além disso, devemos escrever os prefixos em sequência e colocar o valor de partida embaixo do prefixo dado, desenhando a seta no sentido do prefixo desejado. Se o sentido dela for para a esquerda, multiplicaremos o valor fornecido pela base elevada ao expoente positivo. Caso ela se volte para a direita, a multiplicação desse valor será feita com o expoente negativo. Veja:



# MÓDULO 2

# INTRODUÇÃO

Verificaremos, a partir de agora, os fundamentos dos sistemas de numeração posicionais. Eles são indispensáveis para o entendimento da formação dos números nas chamadas bases de numeração, ou seja, a quantidade de símbolos diferentes possíveis.

Particularmente, daremos um enfoque especial às seguintes bases:

## Decimal

Base numérica mais utilizada no nosso cotidiano.

## Binária

Base usada pelos computadores no processamento de dados.

## Octal e hexadecimal

Múltiplos da base binária.

# SISTEMAS DE NUMERAÇÃO

Para entendermos o funcionamento dos sistemas de numeração, precisaremos apresentá-los.

## 1. POSICIONAL

Cada algarismo tem um valor relativo (diferente) de acordo com sua posição no número. Observe:

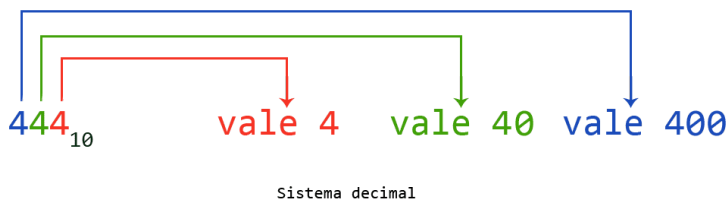


Imagem: Fabio Henrique Silva

Neste exemplo, à medida que os algarismos são acrescentados à esquerda, o número cresce de valor em grupos de 10.

## UMA BASE É A QUANTIDADE DE SÍMBOLOS DE UM SISTEMA POSICIONAL. SEU NÚMERO INDICA O TOTAL DE SÍMBOLOS DELA.

## EXEMPLO

---

Base 10 (decimal) – 0, 1, 2, 3, 4, 5, 6, 7, 8, 9;

Base 2 (binária) – 0, 1;

Base 8 (octal) – 0, 1, 2, 3, 4, 5, 6, 7;

Base 16 (hexadecimal) – 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.



Cada algarismo de um número demonstra o valor de sua **posição** em potências da base. A unidade de uma posição de um sistema de base "z" tem valor equivalente a "z" unidades da posição imediata à direita.

Observe:

$$\begin{array}{c} \text{36}_{10} \text{ 3} \end{array} \begin{array}{c} \text{ * } 10^1 \end{array} + \begin{array}{c} \text{ 6 } \end{array} \begin{array}{c} \text{ * } 10^0$$

$\downarrow$                        $\downarrow$   
10 é a base                      10 é a base  
1 é a posição                      0 é a posição

Todo número cresce de valor da direita para a esquerda

computersciencewiki.org

Em qualquer sistema posicional, o **número** é formado da seguinte maneira:

#### Da direita para a esquerda

Cresce a partir do valor 0 (seguido do 1 até o último algarismo válido).

**Exemplo:** Bases 3 (algarismos 0 a 2) e 10 (algarismos 0 a 9).

#### Retorna a 0

Quando a contagem chega ao último algarismo válido de uma posição, ela retorna a 0 e cresce uma unidade para a esquerda.

**Exemplo:** Base 10 → Cresce de 0 a 9 → Depois, 10, 11, ..., 19 → Na direita, retorna a 0 e cresce 1 para a esquerda: 20.

## 📢 ATENÇÃO

Os movimentos descritos neste exemplo repetem-se infinitamente.

## 2. NÃO POSICIONAL

Todo algarismo tem valor fixo independentemente de sua posição no número.

No **sistema de numeração romano**, o algarismo **X** vale sempre **10**:

XXX = 30

XL = 40

LX = 60

# SISTEMA DE NUMERAÇÃO ROMANO

Este sistema de numeração é não posicional.

## OPERAÇÕES ARITMÉTICAS COM NÚMEROS INTEIROS EM QUALQUER BASE

Agora, vamos estudar as operações aritméticas em bases numéricas diferentes. Dentro de uma unidade especializada (**unidade lógica e aritmética**), o processador usa seus circuitos lógicos digitais para realizar operações lógicas e aritméticas, tendo o *bit* como unidade para representação de dados.

Dentro de uma unidade especializada (**unidade lógica e aritmética**), o processador usa seus circuitos lógicos digitais para realizar operações lógicas e aritméticas, tendo o *bit* como unidade para representação de dados.

Veremos, a seguir, as operações de:

### 1. ADIÇÃO

As parcelas são somadas algarismo por algarismo (pares da mesma coluna), partindo da posição mais à direita até a última posição à esquerda.

Se o resultado da soma de dois algarismos (fazendo em decimal) for igual ou maior que o valor da base, o excesso será subtraído do valor da base, e o resultado, colocado na respectiva posição. Passamos uma unidade (“vai 1”) para a posição imediata à esquerda, que será somada aos dois algarismos seguintes. Observe os exemplos a seguir para entender melhor esse conceito:

#### Na base 10

Base numérica mais utilizada no nosso cotidiano.

$$\begin{array}{r} 11 \\ 574 \\ + 893 \\ \hline 1467 \end{array}$$

#### Na base 2

Na base binária,  $1 + 1 = 0$ , com o “vai 1” = 10

Logo:

$$\begin{array}{r} \text{“vai 1”} \leftarrow 11 \\ 1101 \\ + 1110 \\ \hline 11011 \end{array}$$

#### Na base 16

Antes de abordarmos as regras para cálculos com números hexadecimais, é importante sabermos a **equivalência dos números decimais e seus respectivos hexadecimais**.

111

CD94

+ 8E77

15C0B

# EQUIVALÊNCIA DOS NÚMEROS DECIMAIS E SEUS RESPECTIVOS HEXADECIMAIS

Base 10	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Base 16	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Base 10	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Base 16	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F

Base 10	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
Base 16	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F

- Uma maneira de pensar a adição em qualquer base é:
- Somar os números na posição na base 10;
  - Diminuir o número obtido nesta base pelo valor dela se ele passar do seu último algarismo;
  - Não se esquecer do “vai 1”.



## EXEMPLO

Na base 8:





Em 4 - 7, "pedimos emprestado" o valor da base para o dígito vizinho:  $16 + 4 = 20$ . Logo:  $20 - 7 = 13$ . Em hexadecimal, 13 (na base 10) é igual a D (na base 16).

#### Na 2ª posição:

O valor 9 vale 8 porque "emprestou" ao dígito na 1ª posição do minuendo:  $8 - 7 = 1$ .

#### Na 3ª posição:

O valor D na base 16 vale 13 na base 10, enquanto o valor E na base 16 vale 14 na base 10. "Pedindo emprestado", o dígito da 2ª posição no minuendo será:  $13 + 16 = 29$ . Logo:  $29 - 14 = 15$ , que, na base 16, é representado por F.

#### Na 4ª posição:

O valor C na base 16 vale 12 na base 10. Já o valor 12 vale 11, porque "emprestou" ao dígito na 1ª posição do minuendo:  $11 - 8 = 3$ .

## NA BASE 2

No binário, a operação parece ser mais complicada. Vejamos a seguir:

$0 - 1 = 1$  (e "pede emprestado" do dígito seguinte);

$1 - 1 = 0$ ;

$1 - 0 = 1$ ;

$0 - 1 = 0$ .

Quando temos  $0 - 1$ , precisamos "pedir emprestado" do primeiro algarismo diferente de 0. Esse empréstimo **vale 2 (valor dois em decimal)** por ser um número binário.

Então, no caso da coluna  $0 - 1 = 1$ , a operação feita é  $2 - 1 = 1$ . Como esse processo se repete, o elemento que cedeu o "empréstimo" e valia 1 passa a valer 0.

$$\begin{array}{r} \textcircled{0 \ 1 \ 1 \ 2} \\ 1 \ 0 \ 0 \ 0 \ 1 \\ - 0 \ 1 \ 1 \ 1 \ 0 \\ \hline 0 \ 0 \ 0 \ 1 \ 1 \end{array}$$

É possível verificar que, quando encontramos **0 - 1**:

O zero desta coluna vale **2**.

Todos os zeros à esquerda até o primeiro "número 1" valem 1.

Este "primeiro número 1" vale **0 (zero)**.

Se acontecer novamente  $0 - 1$ , o processo se repetirá.

O "empresta 1" no binário tem valor 2.

## COMPUTAÇÃO ÚTIL

Com recursos de computação cada vez mais baratos e ricos em possibilidades, nossa tendência é não nos preocuparmos tanto com seus aspectos fundamentais. Entretanto, o domínio desse estado da arte lhe possibilitará pensar como um projetista de máquinas e linguagens de programação, podendo desenvolver, no futuro, protótipos capazes de fazer a diferença do ponto de vista tecnológico.



**PODEMOS REALMENTE FAZER COMPUTAÇÃO ÚTIL COM 4.096 *BITS* DE MEMÓRIA? [...] O AVANÇO DA TECNOLOGIA MINIMIZOU ESSAS RESTRIÇÕES. [...] EM UM NÍVEL PRÁTICO, VOCÊ VERÁ QUE É BASTANTE VIÁVEL DESENVOLVER IMPLEMENTAÇÕES EM LINGUAGEM DE MÁQUINA QUE REALIZAM TODOS OS TIPOS DE TAREFAS."**

(SEdgeWICK; WAYNE, 2017)

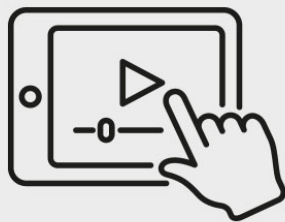
A maior parte dos primeiros programas aplicativos foi implementada dessa maneira. Afinal, por muitos anos, a penalidade sofrida pelo desempenho do uso de uma linguagem de alto nível era grande demais para poder ser paga. A maioria desses códigos foi escrita em *assembly*, cujo funcionamento é similar ao da linguagem de máquina, exceto pelo fato de permitir nomes simbólicos para códigos de operação, registros e localizações de memória.

## COMENTÁRIO

Na década de 1970, era comum ver programas em linguagem *assembly* que se estendiam a dezenas de milhares de linhas de código.

Neste vídeo, o professor Fabio Henrique Silva explica os fundamentos dos sistemas de numeração posicionais e não posicionais, bem como realiza as operações aritméticas de adição e subtração com números inteiros em qualquer base.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



## MÃO NA MASSA

**1. QUAL É O RESULTADO DA OPERAÇÃO ARITMÉTICA  $101101_2 + 100111_2$ ?**

- A) 0101011
- B) 201212
- C) 1010100
- D) 110

**2. QUAL É O RESULTADO DA OPERAÇÃO ARITMÉTICA  $1436_8 - 556_8$ ?**

- A) 2214
- B) 107
- C) 1992
- D) 660

**3. A PARTIR DO VALOR BINÁRIO (BASE 2) 110011, APONTE OS QUATRO VALORES SEGUINTE:**

- A) 110100; 110101; 110110; 110111
- B) 110011; 110010; 110001; 110000
- C) 110100; 110110; 110101; 110001
- D) 101100; 101101; 101110; 101111

**4. A PARTIR DO VALOR OCTAL (OCTAL) 1365, OS CINCO VALORES SEGUINTE SÃO:**

- A) 1365; 1366; 1367; 1368; 1370
- B) 1366; 1367; 1368; 1370; 1371
- C) 1366; 1367; 1370; 1371; 1372
- D) 1365; 1366; 1366; 1366; 1366

---

## GABARITO

**1. Qual é o resultado da operação aritmética  $101101_2 + 100111_2$ ?**

A alternativa "C " está correta.

Vamos aos cálculos:

$$\begin{array}{r} \phantom{+} 1 \phantom{0000} \\ \phantom{+} 1 \phantom{0} 1 \phantom{0} 1 \phantom{0} 1 \\ \phantom{+} 1 \phantom{0} 1 \phantom{0} 1 \phantom{0} 1 \phantom{0} 1 \\ + \phantom{0} 1 \phantom{0} 0 \phantom{0} 1 \phantom{0} 1 \phantom{0} 1 \\ \hline 1 \phantom{0} 0 \phantom{0} 1 \phantom{0} 0 \phantom{0} 1 \phantom{0} 0 \phantom{0} \end{array}$$

Você pode realizar a operação de adição normalmente, como é feito na base 10. Uma confusão comum entre os iniciantes é a operação  $1 + 1$ , que é igual a 10.

**2. Qual é o resultado da operação aritmética  $1436_8 - 556_8$ ?**

A alternativa "D " está correta.

Vamos aos cálculos:

$$\begin{array}{r}
 0 \ 11 \ 11 \\
 1 \ 4 \ 3 \ 6 \\
 - \ 5 \ 5 \ 6 \\
 \hline
 6 \ 6 \ 0
 \end{array}$$

Note que o 3 “pede emprestado” o valor da base em decimal (8), que é somado a ele mesmo (também em decimal). Assim:  $3 + 8 = 11$ . O cálculo continua normalmente como se fosse em decimal.

### 3. A partir do valor binário (base 2) 110011, aponte os quatro valores seguintes:

A alternativa "A " está correta.

Vamos à análise:

#### 1º número

Partiremos do número **1 1 0 0 1 1**. Lembre-se de que, quando a contagem chega ao último algarismo válido de uma posição, ela retorna a 0 e cresce uma unidade para a esquerda.

Na base binária, temos os seguintes algarismos: **0; 1**. Concentre-se no número mais à direita (“posição 0”), pois ele vale 1.

Na base 2, quais são os valores possíveis para cada posição? Resposta: 0 e 1. Como o número mais à direita vale 1, ele já está no “limite” de valores possíveis para a posição, ou seja, não existe valor numérico após 1. Logo, ele deverá “voltar” para o primeiro algarismo possível, que é **0**.

“Ande” uma posição para a esquerda (“posição 1”). O valor é **1**. O valor deve “voltar” para o primeiro algarismo possível, que é **0**. “Andando” para a próxima posição (“posição 2”), seu valor numérico é 0. Logo, ele passa a valer **1**.

Temos, assim, a sequência: **100**. Basta, agora, repetir os demais algarismos, ficando: **110100**.

#### 2º número

Você deve voltar para a “posição 0” mais à esquerda, que vale **0**.

Qual é o algarismo possível após o 0 na base 2? É o **1**. Dessa forma, o próximo número da sequência será: **110101**.

#### 3º número

A posição 0 mais à esquerda vale **1**. Não existe valor numérico após **1**. Portanto, ela retorna a 0 e cresce uma unidade para a esquerda, ficando: **110110**.

#### 4º número

O valor da “posição 0” mais à esquerda corresponde a 0, que passará a valer 1. Temos, então, **110111**.

### 4. A partir do valor octal (octal) 1365, os cinco valores seguintes são:

A alternativa "C " está correta.

Vamos à análise:

#### 1º número

Partiremos do número **1365**. Lembre-se de que, quando a contagem chega ao último algarismo válido de uma posição, ela retorna a 0 e cresce uma unidade para a esquerda.

Na base octal, temos os seguintes algarismos: **0; 1; 2; 3; 4; 5; 6; 7**. Não existe número 8!

O valor na posição mais à direita corresponde a **5**. Você pode mudá-lo para 6. Então, ele fica igual a **1366**.

### 2º número

Concentre-se no número mais à direita (“posição 0”). Ele vale **6**. Podemos “subir” mais um algarismo, passando de 6 para o valor 7. Resultado: **1367**.

### 3º número

A posição mais à esquerda possui o maior valor possível na base 8, que é 7. O valor deve “voltar” para o primeiro de algarismo possível, que é **0**, e crescer uma unidade para a esquerda. Resultado: **1370**.

### 4º e 5º números

Voltando à “posição 0”, mais à direita, comece a contagem normalmente a partir do 0. Resultado: **1371**; **1372**.

## VERIFICANDO O APRENDIZADO

**1. A PARTIR DO VALOR 2 4 5 6 EM BASE 7, OS CINCO VALORES SEGUINTE SÃO:**

- A) 2456; 2457; 2458; 2459; 2460
- B) 2456; 2457; 2460; 2461; 2461
- C) 2457; 2460; 2461; 2462; 2463
- D) 2460; 2461; 2462; 2463; 2464

**2. A PARTIR DO VALOR BINÁRIO 1100101, OS QUATRO NÚMEROS SEGUINTE, SALTANDO DE DOIS EM DOIS, SÃO:**

- A) 1100111; 1101001; 1101011; 1101101
- B) 1100110; 1100111; 1101000; 1101001
- C) 1100101; 1100111; 1100101; 1100111
- D) 1100111; 1100112; 1100113; 1100114

**3. QUAL É O RESULTADO DA OPERAÇÃO ARITMÉTICA  $4C7BE8_{16} - 1E927A_{16}$ ?**

- A) 6B0E62
- B) 6B0E63
- C) 2DE96E
- D) 6B0E6B

**4. QUAL É O RESULTADO DA OPERAÇÃO ARITMÉTICA  $7312_8 - 3465_8$ ?**

- A) 1941

- B) 3625
- C) 123567
- D) 795

**5. A PARTIR DO VALOR HEXADECIMAL (BASE 16) 2BEED, OS CINCO VALORES SEGUINTE SÃO:**

- A) 3BEED; 4BEED; 5BEED; 6BEED; 7BEED
- B) 2BEEE; 2BEEF; 2BEF0; 2BEF1; 2BEF2
- C) 2CEED; 2DEED; 2EEED; 2FEED; 30EED
- D) 2BEEE; 2BEEF; 2BEEG; 2BEEH; 2BEEH

**6. QUAL É O RESULTADO DA OPERAÇÃO ARITMÉTICA  $110000001101_2 - 10110011101_2$ ?**

- A) 11001100000
- B) 10110110000
- C) 11100010000
- D) 11001110000

---

## GABARITO

**1. A partir do valor 2 4 5 6 em base 7, os cinco valores seguintes são:**

A alternativa **"D "** está correta.

Vamos à análise:

### 1º número

Partiremos do número **2456**. Lembre-se de que, quando a contagem chega ao último algarismo válido de uma posição, ela retorna a 0 e cresce uma unidade para a esquerda.

Na base 7, temos os seguintes algarismos: **0; 1; 2; 3; 4; 5; 6. Não existe número 7!**

A contagem já está no último algarismo válido de uma posição. Ele retorna a 0 e cresce uma unidade para a esquerda: **2460**.

### 2º, 3º, 4º e 5º números

Comece a contagem normalmente a partir do 0, voltando à “posição 0” mais à direita: **2461; 2462; 2463; 2464**.

**2. A partir do valor binário 1100101, os quatro números seguintes, saltando de dois em dois, são:**

A alternativa **"A "** está correta.

Vamos à análise:

### 1º número

Partiremos do número **1100101**. Lembre-se de que, quando a contagem chega ao último algarismo válido de uma posição, ela retorna a 0 e cresce uma unidade para a esquerda.





Vamos à análise:

1º e 2º números

Partiremos do número **2BEED**. Lembre-se de que, quando a contagem chega ao último algarismo válido de uma posição, ela retorna a 0 e cresce uma unidade para a esquerda.

Na base hexadecimal, temos os seguintes algarismos: **0; 1; 2; 3; 4; 5; 6; 7; 8; 9; A; B; C; D; E; F**.

O valor na posição mais à direita corresponde a **D**. Você seguirá mudando as letras do alfabeto até a letra **F**, quando, então, o valor na posição voltará a **0**. Logo, os dois primeiros números são **2BEEE** e **2BEED**.

3º número

A contagem chegou ao último algarismo válido de uma posição. Ela retorna a 0 e cresce uma unidade para a esquerda: **2BEED** → **2BEF0**.

4º e 5º números

Comece a contagem normalmente a partir do 0. Com isso, você encontrará **2BEF1** e **2BEF2**.

6. Qual é o resultado da operação aritmética  $110000001101_2 - 10110011101_2$ ?

A alternativa **"D"** está correta.

Vamos aos cálculos:

0111112

11000001101

-10111001101

11001110000

Repare a sequência interessante de zeros que existe no minuendo. Você terá de “pedir emprestado” para o número 1, que está “bem distante”. Logo, todos os zeros corresponderão ao valor da base menos 1, enquanto o valor que “pediu emprestado” corresponderá à base 2.

# MÓDULO 3

🕒 Empregar a conversão entre os sistemas de numeração

## INTRODUÇÃO

Entenderemos, a partir de agora, como são convertidos os valores entre bases numéricas distintas. Além disso, realizaremos operações aritméticas de adição e subtração em outras bases diferentes da base decimal.

DOMINAR ESSE PROCEDIMENTO É IMPORTANTE PARA POSSIBILITAR A ALTERNÂNCIA ENTRE OS SISTEMAS DE NUMERAÇÃO DECIMAL

(UTILIZADOS NORMALMENTE EM NOSSO COTIDIANO) E TRÊS OUTROS SISTEMAS: BINÁRIO, OCTAL E HEXADECIMAL (USADOS PELO COMPUTADOR).

## CONVERSÃO ENTRE SISTEMAS DE NUMERAÇÃO

Embora os sistemas de computação expressem seus valores na base 2 (binária), eles poderão ser representados em outras bases quando houver a necessidade de promover uma melhor visualização e compactar os seus valores. Geralmente, são usadas as bases 8 (octal) e 16 (hexadecimal), pois ambas são múltiplas de 2.

Esta tabela mostra a equivalência dos valores nas bases 2, 8, 10 e 16:

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

Base 2	Base 8	Base 10	Base 16
1111	17	15	F
10000	20	16	10
10001	21	17	11

Equivalência dos valores nas bases 2, 8, 10 e 16.

Em bases de valor superior a 10, usamos letras do alfabeto para representação de algarismos maiores que 9. As tabelas a seguir comparam os valores entre as bases 10 e 16:

Base 10	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Base 16	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Base 10	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Base 16	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F

Base 10	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
Base 16	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F

Comparação dos valores entre as bases 10 e 16

Conheceremos duas maneiras de realizar a conversão de valores:

1. DE UMA BASE X PARA OUTRA Y

A conversão sempre precisa ser realizada **tendo a base 10** como intermediária, pois ela é a utilizada para efetuar cálculos aritméticos.

Após a conversão da base X (origem) para a base 10, o resultado será convertido para a base Y (destino).

Conseguiremos entender isso melhor com o seguinte passo a passo:

1

2

3

4

5

6

7

Vamos converter 234 da base 6 para um valor equivalente na base 8. O número a ser convertido é expresso em  $N$  produtos, em que  $N$  é igual à quantidade algarismo do número. Assim, temos:

$$234_6 = 2 * 6^2 + 3 * 6^1 + 4 * 6^0$$

Calcularemos, agora, os produtos usando a aritmética da base 10. Desse modo, o resultado **será expresso em valores decimais**.

Continuando a realizar o cálculo, teremos:

$$(2 * 36) + 18 + 4 = 94_{10}$$

Dessa vez, o resultado (decimal 94) será convertido para a base desejada (8). Esse processo é o inverso do anterior – afinal, o inverso da multiplicação é a divisão. Dividimos o valor (94) pela base desejada (8). O resto obtido é o primeiro algarismo do número desejado (aquele mais à direita):

$$94 / 8 = 11 \text{ e resto} = 6$$

O quociente obtido na divisão é novamente dividido por 8. A seguir, o novo resto é acrescentado à esquerda do primeiro algarismo:

$$11 / 8 = 1 \text{ e resto} = 3$$

O número a ser obtido na base 8 é: 36.

Novamente, faremos a divisão até obtermos um quociente 0 (zero):

$$1 / 8 = 0 \text{ e resto} = 1$$

O número obtido, portanto, é: 136<sub>8</sub>.

## 2. EM CASOS ESPECIAIS

As conversões de valores entre as **bases 2 e 8, 2 e 16 e 8 e 16** podem ser realizadas com este método. Tenha em mente o seguinte: se você tomar o **valor da base binária** e o elevar à sua posição (em decimal), obterá o valor relativo da posição da base 2 representado em decimal.

A seguir, analisaremos dois exemplos.

### VALOR DA BASE BINÁRIA

Como este valor está representado em decimal, escreva 2.

1. Observe estes três *bits*:

$2^2$	$2^1$	$2^0$
=	=	=
4	2	1

Toda vez que o dígito 1 for inserido em uma posição, ela corresponderá ao valor relativo da posição (escrito em decimal). Em outras palavras, toda vez que você colocar:

1

“Liga” o *bit* na posição

0

“Desliga” o *bit* na posição

Logo, o valor final em decimal será a soma de todos os valores (escritos em decimal) relativos das posições que estiverem “ligados”.

2. Para “ligar” algumas posições, colocaremos, agora, o valor 1 nelas:

1 0 1

Foram colocados valores “1” nas posições 0 e 2. Somando os relativos às posições em decimal, temos:  $4 + 1 = 5$ .

Desse modo, o **binário 101** corresponde ao **valor 5 em decimal**.

A tabela a seguir mostra algumas correspondências:

$2^0 = 1$	$2^7 = 128$
$2^1 = 2$	$2^8 = 256$
$2^2 = 4$	$2^9 = 512$
$2^3 = 8$	$2^{10} = 1024$
$2^4 = 16$	$2^{11} = 2048$
$2^5 = 32$	$2^{12} = 4096$
$2^6 = 64$	(...)

**Valores relativos às posições em decimal para os números correspondentes em binário.**

Com essas informações, podemos observar, dentro dos casos especiais citados, como são realizadas as conversões entre as seguintes bases:

## BASES 2 E 8

Como  $8 = 2^3$ , dividimos o valor, partindo da direita para a esquerda, em grupos de três *bits* :

	$(111010111)_2$		$=()_8$
$(111)$	$(010)$	$(111)_2$	$=(727)_8$

7	2	7	

BASES 2 E 16

Como  $16 = 2^4$ , dividimos o valor, partindo da direita para a esquerda, em grupos de quatro *bits* :

	$(1011011011)_2$		$= ( )_{16}$
$(0010)$	$(1101)$	$(1011)_2$	$= (2DB)_{16}$
2	D	B	

BASES 8 E 16

Converte-se primeiramente para a base 2 e, em seguida, para a desejada:

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

Observemos, agora, as conversões para as bases:

$2 \rightarrow (011)(001)(111)(100)_2 = (011001111100)_2$

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

O resultado da conversão, portanto, será: 67C.

LÓGICA BOOLEANA

Além de sabermos que um computador representa seus dados usando *bits*, já compreendemos como são realizadas algumas operações matemáticas, bem como a conversão de valores para outras bases numéricas responsáveis por compactar os números binários.

OS VALORES BINÁRIOS SEGUEM A *LÓGICA BOOLEANA*. NELA, DOIS VALORES SÃO REFERIDOS COMO VERDADEIRO E FALSO, SIM E NÃO OU 0 E 1. NÃO IMPORTA A ESCOLHA FEITA PARA ESSA REFERÊNCIA: SEU CONCEITO É O MESMO.

LÓGICA BOOLEANA

Expressão cuja origem remete a George Boole (1815-1864), filósofo britânico responsável pela criação da álgebra booleana. Ela foi fundamental para o desenvolvimento da computação moderna.

A conexão íntima entre a lógica booleana e os circuitos que executam tarefas computacionais é um conceito fundamental na estruturação da infraestrutura computacional da qual desfrutamos atualmente.

Entender essa conexão significa compreender a pergunta de Sedgewick e Wayne (2017): "**Como os circuitos calculam?**". Afinal, quando você passa a saber e *sentir* como é o funcionamento dos circuitos lógicos, também consegue entender a importância da representação dos valores binários 0 e 1 em outras bases numéricas.

Neste vídeo, o professor Fabio Henrique Silva explica como os valores entre bases numéricas distintas são convertidos.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



## MÃO NA MASSA

### 1. INDIQUE O RESULTADO DESTA OPERAÇÃO ARITMÉTICA:

$$6513_7 - 133_5 = \underline{\hspace{2cm}}_3$$

- A) 10011200
- B) 10010002
- C) 10010001
- D) 10011103

### 2. QUAL É O VALOR DECIMAL EQUIVALENTE AO MAIOR NÚMERO DE SETE ALGARISMOS QUE PODE SER CRIADO NA BASE 2?

- A) 7
- B) 31
- C) 127
- D) 255

### 3. CONVERTA O SEGUINTE VALOR PARA A BASE INDICADA:

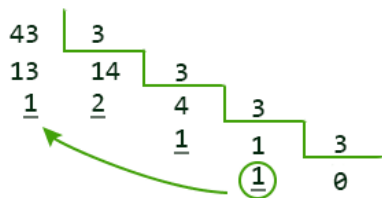
$$2317_8 = ( \ )_2$$

#### O RESULTADO É:

- A) 110011010000
- B) 010011001111
- C) 110010110010

- Da base 10 para a base 3:





O resultado é:

$$\begin{array}{r}
 \begin{array}{cccccccc}
 & 1 & 0 & 0 & 1 & 1 & \overset{1}{2} & \overset{2}{0} & \overset{3}{0} \\
 (3) & & & & & 1 & 1 & 2 & 1 & - \\
 \hline
 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 
 \end{array}
 \end{array}$$

Tiremos, por fim, a prova real para checar o resultado:

$$\begin{array}{r}
 \begin{array}{cccccccc}
 & 1 & 0 & 0 & 1 & 0 & \overset{1}{0} & \overset{1}{0} & 2 \\
 (3) & & & & & 1 & 1 & 2 & 1 & + \\
 \hline
 & 1 & 0 & 0 & 1 & 1 & 2 & 0 & 0 & 
 \end{array}
 \end{array}$$

2. Qual é o valor decimal equivalente ao maior número de sete algarismos que pode ser criado na base 2?

A alternativa "C " está correta.

Distribua as posições nas quais você escreverá os valores binários na quantidade de sete posições:

$\overline{2^6}$	$\overline{2^5}$	$\overline{2^4}$	$\overline{2^3}$	$\overline{2^2}$	$\overline{2^1}$	$\overline{2^0}$
=	=	=	=	=	=	=
64	32	16	8	4	2	1

Para saber a combinação máxima entre as possibilidades de valores, coloque o valor 1 em todas as posições e some os valores das relações entre os valores binário e decimal de cada posição:

$\frac{1}{2^6}$	$\frac{1}{2^5}$	$\frac{1}{2^4}$	$\frac{1}{2^3}$	$\frac{1}{2^2}$	$\frac{1}{2^1}$	$\frac{1}{2^0}$
=	=	=	=	=	=	=
64	32	16	8	4	2	1

$$64 + 32 + 16 + 8 + 4 + 2 + 1 = 127$$

3. Converta o seguinte valor para a base indicada:

$$2317_8 = ( )_2$$

O resultado é:

A alternativa "B " está correta.

Vamos fazer a conversão de  $2317_8$  da base 8 (octal) para a base 2 (binária):

2			3			1			7		
0	1	0	0	1	1	0	0	1	1	1	1

Observação: O zero à esquerda pode ser omitido.

4. Indique o resultado desta operação aritmética:

$$235_8 + 100110110_2 = \text{_____}_{16}$$

A alternativa está correta.

Vamos fazer a conversão de  $235_8$ .

- Da base 8 (octal) para a base 2 (binária):

2			3			5		
0	1	0	0	1	1	1	0	1

- Da base 2 (binária) para a base 16 (hexadecimal):

0	0	0	0	1	0	0	1	1	1	0	1
0				9				13 → D			

Agora, é a vez de fazermos a conversão de  $100110110_2$  da base 2 (binário) para a base 16 (hexadecimal):

0	0	0	1	0	0	1	1	0	1	1	0
1				3				6			

Realizando a operação, temos:

Base 10	0	(...)	9	10	11	12	13	14	15	16	17	18	19	(...)
Base 16	0	(...)	9	A	B	C	D	E	F	10	11	12	13	(...)

O resultado é:

$$\begin{array}{r}
 \text{(16)} \quad \begin{array}{ccc} & \overset{1}{3} & 6 \\ 1 & 9 & D \\ \hline 1 & D & 3 \end{array} +
 \end{array}$$

# VERIFICANDO O APRENDIZADO

1. CONVERTA O SEGUINTE VALOR PARA A BASE INDICADA:

$$10011110001011_2 = ( \quad )_{16}$$

- A) 2E5C
- B) 7834
- C) AB3D
- D) 278B

2. CONVERTA O SEGUINTE VALOR PARA A BASE INDICADA:

$$2BEF5_{16} = ( \quad )_8$$

- A) 573241
- B) 852387
- C) 635421
- D) 537365

3. CONVERTA O SEGUINTE VALOR DA BASE INDICADA PARA A OUTRA À DIREITA:

$$2464_9 = ( \quad )_4$$

- A) 130300
- B) 300203
- C) 130004
- D) 323013

4. A MAIORIA DAS PESSOAS SÓ PODE CONTAR ATÉ 10 UTILIZANDO SEUS DEDOS. NO ENTANTO, QUEM TRABALHA COM COMPUTADORES PODE FAZER ISSO MELHOR. IMAGINE CADA UM DE SEUS DEDOS COMO UM DÍGITO BINÁRIO. CONSIDERE QUE O DEDO ESTENDIDO SIGNIFICA O ALGARISMO 1 E O RECOLHIDO, O ALGARISMO 0.

**ATÉ QUAL NÚMERO (EM VALOR DECIMAL) VOCÊ PODE CONTAR USANDO AS DUAS MÃOS?**

- A) 10
- B) 100
- C) 1000
- D) 1023

**5. CONVERTA O SEGUINTE VALOR DA BASE INDICADA PARA A OUTRA À DIREITA:**

$$234_6 = ( \quad )_8$$

**O RESULTADO É:**

- A) 234
- B) 136
- C) 365
- D) 154

**6. EXPRESSE O VALOR DECIMAL 100 NA BASE 2:**

- A) 1100100
- B) 1001000
- C) 0000100
- D) 1000000

---

## **GABARITO**

**1. Converta o seguinte valor para a base indicada:**

$$10011110001011_2 = ( \quad )_{16}$$

A alternativa "**D**" está correta.

Vamos fazer a conversão de  $10011110001011_2$  da base 2 (binária) para a base 16 (hexadecimal).

Divida o binário da seguinte maneira, começando da direita para a esquerda:

10 0111 1000 1011

Acrescente zeros à esquerda para completar o último grupo:

0010 0111 1000 1011

0	0	1	0	0	1	1	1	1	0	0	0	1	0	1	1
2				7				8				11 → B			

2. Converta o seguinte valor para a base indicada:

$2BEF5_{16} = ( )_8$

A alternativa "D " está correta.

Vamos fazer a conversão de  $2BEF5_{16}$ .

- Da base 16 (hexadecimal) para a base 2 (binária):

2				B				E				F				5			
0	0	1	0	1	0	1	1	1	1	1	0	1	1	1	1	0	1	0	1

- Da base 2 (binária) para a base 8 (octal):

Reagrupe os *bits* , começando **da direita para a esquerda**, incluindo zeros à esquerda se for necessário para completar a posição. Assim, você terá:

0	0	0	1	0	1	0	1	1	1	1	1	0	1	1	1	1	0	1	0	1
0			5			3			7			3			6			5		

3. Converta o seguinte valor da base indicada para a outra à direita:

$2464_9 = ( )_4$

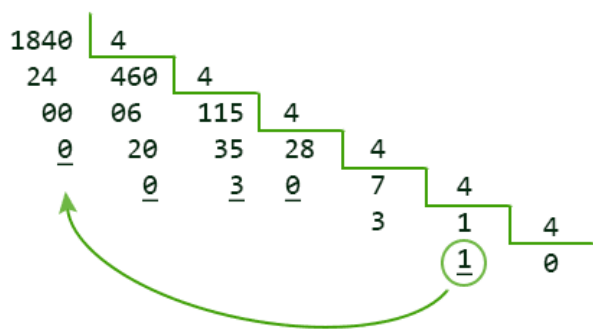
A alternativa "A " está correta.

Vamos fazer a conversão de  $2464_9$ .

- Da base 9 para a base 10:

$2 \times 9^3$	$+ 4 \times 9^2$	$+ 6 \times 9^1$	$+ 4 \times 9^0$	
1458	324	54	4	= 1840

- Da base 10 para a base 4:



4. A maioria das pessoas só pode contar até 10 utilizando seus dedos. No entanto, quem trabalha com computadores pode fazer isso melhor. Imagine cada um de seus dedos como um dígito binário. Considere que o dedo estendido significa o algarismo 1 e o recolhido, o algarismo 0.

Até qual número (em valor decimal) você pode contar usando as duas mãos?

A alternativa "D " está correta.

Distribua as posições nas quais você escreverá os valores binários. Para cada posição, tome o valor da base binária (ele está representado em decimal; logo, escreva 2) e o eleve à sua posição (em decimal).

Desse modo, você obtém o valor relativo da posição da base 2 representado em decimal:

$2^9$	$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
=	=	=	=	=	=	=	=	=	=
512	256	128	64	32	16	8	4	2	1

Aqui, vemos a representação dos 10 dedos: do “dedo 0” ao “dedo 9”. Quando um dedo da mão estiver recolhido, indicará o valor 0. Se ficar estendido, indicará o valor 1.

Para saber a combinação máxima entre as possibilidades de valores de dedos recolhidos e estendidos, basta colocar o valor 1 em todas as posições e somar os valores das relações entre os valores binário e decimal de cada posição:



$1$	$1$	$1$	$1$	$1$	$1$	$1$	$1$	$1$	$1$
$2^9$	$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
=	=	=	=	=	=	=	=	=	=
512	256	128	64	32	16	8	4	2	1

Resposta:  $512 + 256 + 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 1023$

5. Converta o seguinte valor da base indicada para a outra à direita:

$$234_6 = ( )_8$$

O resultado é:

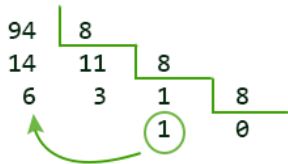
A alternativa "B " está correta.

Vamos fazer a conversão de  $234_6$ .

- Da base 6 para a base 10:

$2 \times 6^2$	$+ 3 \times 6^1$	$+ 4 \times 6^0$	
72	18	4	= 94

- Da base 10 para a base 8:



## 6. Expresse o valor decimal 100 na base 2:

A alternativa "A " está correta.

Distribua as posições nas quais você escreverá os valores binários. Para cada posição, tome o valor da base binária (ele está representado em decimal; logo, escreva 2) e o eleve à sua posição (em decimal).

Comece a escrever isso da direita para a esquerda. Não se preocupe sobre a quantidade de posições necessárias, pois, quando o valor final for alcançado, todas as que não forem representativas terão zeros (à esquerda).

Neste exercício, já foram colocadas as posições suficientes:

$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
=	=	=	=	=	=	=
64	32	16	8	4	2	1

Lembre-se: toda vez que você colocar 1 na posição, ela terá o valor relativo ao da base em decimal. O valor final (em decimal) será a soma de todos os valores relativos às posições em decimal nas quais houver o valor 1.

O somatório dos valores "1" colocados terá de ser 100. Comece da esquerda para a direita, ou seja, do maior para o menor:

1						
$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
=	=	=	=	=	=	=
64	32	16	8	4	2	1

"Ligamos" a posição 6, cuja relação existente entre o valor da posição em binário e em decimal é esta:  $2^6 = 64$ . Em seguida, subtraímos 64 de 100 para saber quanto falta até 100:  $100 - 64 = 36$ . Observe:

$\frac{1}{2^6}$	$\frac{1}{2^5}$	$\frac{1}{2^4}$	$\frac{1}{2^3}$	$\frac{1}{2^2}$	$\frac{1}{2^1}$	$\frac{1}{2^0}$
=	=	=	=	=	=	=
64	32	16	8	4	2	1

“Ligamos” a posição 5, cuja relação existente entre o valor da posição em binário e em decimal é a seguinte:  $2^5 = 32$ . Depois, subtraímos 32 de 36 para saber quanto falta até 100:  $36 - 32 = 4$ .

Não conseguimos “ligar” os valores relativos às posições 4 ( $2^4 = 16$ ) e 3 ( $2^3 = 8$ ), pois a soma dos valores extrapola o resultado desejado. Portanto, colocamos 0 (“desligar”) nelas.

Enfim, conseguimos “ligar” os valores relativos à posição 2 ( $2^2 = 4$ ):

$\frac{1}{2^6}$	$\frac{1}{2^5}$	$\frac{0}{2^4}$	$\frac{0}{2^3}$	$\frac{1}{2^2}$	$\frac{1}{2^1}$	$\frac{1}{2^0}$
=	=	=	=	=	=	=
64	32	16	8	4	2	1

Já temos, agora, o valor desejado:  $4 - 4 = 0$ . Basta preencher, então, o restante das posições com os zeros:

$\frac{1}{2^6}$	$\frac{1}{2^5}$	$\frac{0}{2^4}$	$\frac{0}{2^3}$	$\frac{1}{2^2}$	$\frac{0}{2^1}$	$\frac{0}{2^0}$
=	=	=	=	=	=	=
64	32	16	8	4	2	1

## MÓDULO 4

### ⦿ Categorizar as tabelas de representação de dados

## INTRODUÇÃO

Os *bits* constituem a forma básica da representação de dados de um computador. Suas informações são obtidas conforme padrões de representações. Afinal, o computador precisa utilizar valores numéricos para representar os sinais gerados no *hardware*.

No entanto, estamos habituados a representar dados e informações por meio de letras, palavras, frases etc. Esse processo se repete quando um usuário comum deseja receber uma informação em caracteres.

Apontaremos, a seguir, de que maneira ocorre a representação de conjuntos de *bits* em caracteres, além de apresentarmos tabelas de representação de dados.



# TABELAS DE REPRESENTAÇÃO DE DADOS

Em uma tabela de representação de dados, cada símbolo possui uma correspondência com um grupo de *bits* que identifica exclusivamente determinado **caractere**.



## EXEMPLO

Quando alguém escreve um texto no teclado, os caracteres do alfabeto são convertidos em outros codificados em *bits*.



Os tipos primitivos de dados podem ser classificados em:

### Caractere

Representa símbolos (não numéricos). Modo primário de introduzir dados no computador. Serve para escrever um texto em algum idioma.

### Lógico

Representa verdadeiro ou falso.

### Numérico

Representa os números.

Apresentaremos, a seguir, dois conjuntos de códigos de caracteres:

## A) ASCII

*American Standard Code for Information Interchange* (em português, Código Padrão Americano para Troca de Informações).

Cada caractere tem sete *bits*. O ASCII possui um total de 128 caracteres ( $2^7$ ) que podem ser expressos em hexadecimal.

Os códigos 0 a 1F, por sua vez, não são impressos, pois ambos são caracteres de controle.

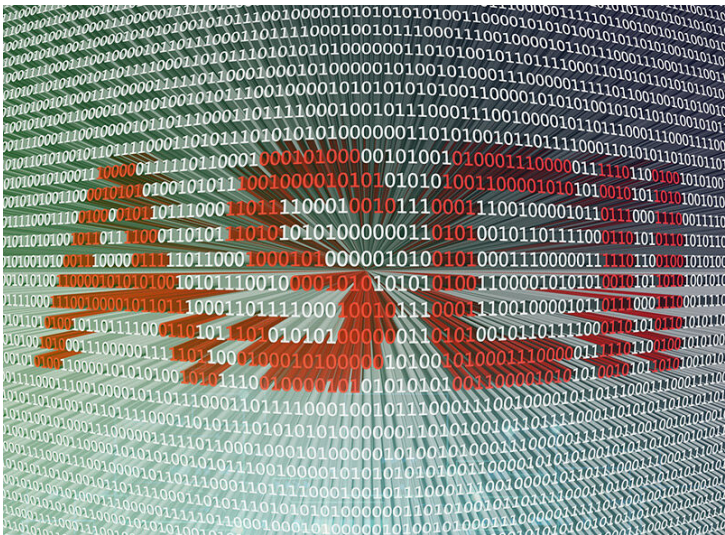


Foto: Shutterstock.com

Tanembaum (2007) informa que o ASCII ainda possui:

### Letras maiúsculas

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

### Letras minúsculas

a b c d e f g h i j k l m n o p q r s t u v w x y z

Sinais de pontuação

\* ~ . , ; " ' + = ! ? @ / # ( ) { } [ ]

Símbolos matemáticos

+ \* / - =

As tabelas a seguir exibem o conjunto de caracteres ASCII:

Hex	Nome	Significado	Hex	Nome	Significado
0	NUL	Null	10	DLE	Data Link Scape
1	SOH	Start Of Heading	11	DC1	Device Control 1
2	STX	Start Of Text	12	DC2	Device Control 2
3	ETX	End Of Text	13	DC3	Device Control 3
4	EOT	End Of Transmission	14	DC4	Device Control 4
5	ENQ	Enquiry	15	NAK	Negative ACKnowledgement
6	ACK	ACKnowledgement	16	SYN	SYNchronous Idle
7	BEL	BELI	17	ETB	End Of Transmission Block
8	BS	BackSpace	18	CAN	CANcel
9	HT	Horizontal Tab	19	EM	End Of Medium
A	LF	Line Feed	1a	SUB	SUBstitute
B	VT	Vertical Tab	1B	ESC	ESCape
C	FF	Form Feed	1C	FS	File Separator
D	CR	Carriage Return	1D	GS	Group Separator
E	SO	Shift Out	1E	RS	Record Separator
F	SI	Shift In	1F	US	Unit Separator

Hex	Car.	Hex	Car.	Hex	Car.	Hex	Car.	Hex	Car.	Hex	Car.
20	(Space)	30	0	40	@	50	P	60	`	70	p

Hex	Car.	Hex	Car.	Hex	Car.	Hex	Car.	Hex	Car.	Hex	Car.
21	!	31	1	41	A	51	Q	61	a	71	q
22	"	32	2	42	B	52	R	62	b	72	r
23	#	33	3	43	C	53	S	63	c	73	s
24	\$	34	4	44	D	54	T	64	d	74	t
25	%	35	5	45	E	55	U	65	e	75	u
26	&	36	6	46	F	56	V	66	f	76	v
27	'	37	7	47	G	57	W	67	g	77	w
28	(	38	8	48	H	58	X	68	h	78	x
29	)	39	9	49	I	59	Y	69	i	79	y
2A	*	3A	:	4A	J	5A	Z	6A	j	7A	z
2B	+	3B	;	4B	K	5B	[	6B	k	7B	{
2C	,	3C	<	4C	L	5C	\	6C	l	7C	
2D	-	3D	=	4D	M	5D	]	6D	m	7D	}
2E	.	3E	>	4E	N	5E	^	6E	n	7E	~
2F	/	3F	?	4F	O	5F	_	6F	o	7F	DEL

Tabela: Conjunto de caracteres ASCII

Fonte: TANEMBAUM, 2007, p. 73.

## B) UNICODE

O ASCII é um bom conjunto de caracteres para textos de língua inglesa, mas não tanto para os demais idiomas, como japonês, chinês, árabe etc. Isso levou um grupo de empresas a criar o Unicode, padrão internacional no qual cada caractere possui um único valor: 16 *bits*.

Com isso, podemos ter um total de 65.536 símbolos (2<sup>16</sup>).



Foto: Shutterstock.com

Além do alfabeto e dos símbolos de pontuação, existem códigos para:

#### Símbolos monetários

£ ¥ ₣ £ € \$

#### Símbolos matemáticos

+ \* / - =

#### Formas geométricas

■ □ △ ○ ◆ ▽

#### Emojis

😊 😊 😊 🤖 🤖

Segundo Tanenbaum (2007), o consórcio Unicode estuda e decide todas as novas propostas de inclusão de novos caracteres.



## EXEMPLO

Quando um usuário digita no teclado a palavra SABER (em letras maiúsculas), seus caracteres são convertidos conforme a tabela a seguir:

	S	A	B	E	R
ASCII	01010011	01000001	01000010	01000101	01010010
Unicode	U+0053	U+0041	U+0042	U+0045	U+0052

#### Conjuntos de códigos de caracteres ASCII e Unicode



# EVOLUÇÃO DOS CARACTERES

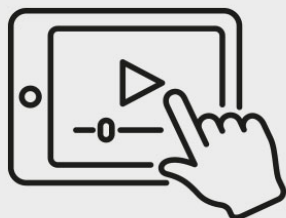
Os dispositivos que você usa hoje em dia suportam muitos caracteres especiais. Se você acompanhou a evolução daqueles apresentados pelos programas, lembra de algumas incompatibilidades que aconteciam entre aplicativos diferentes ou que estavam relacionadas a idiomas distintos.

## ESSAS INCOMPATIBILIDADES OCORRIAM QUANDO NÃO HAVIA PADRONIZAÇÕES (COMO AS QUE EXISTEM ATUALMENTE) NEM UMA ADEQUAÇÃO ÀS APLICAÇÕES. ISSO PREJUDICA TANTO A POSSIBILIDADE DA MELHOR EXPLORAÇÃO DE UM NEGÓCIO QUANTO SUA ACESSIBILIDADE E USABILIDADE.

Tenha sempre em mente a preocupação de promover a maior compatibilidade possível entre os conjuntos de caracteres, caso você venha a trabalhar com isso em funções como programador de aplicativos de rede, programador *web*, *webdesigner* etc.

Neste vídeo, o professor Fabio Henrique Silva comenta a diferença entre os conjuntos de códigos ASCII e Unicode.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



## MÃO NA MASSA

**1. DETERMINADO PADRÃO DE CONJUNTOS DE CARACTERES SUPORTA 512 CARACTERES. ISSO SIGNIFICA QUE CADA CARACTERE POSSUI UM ÚNICO VALOR DE:**

- A) 16 *bits*
- B) 2 *bits*
- C) 8 *bits*
- D) 9 *bits*

---

### GABARITO

**1. Determinado padrão de conjuntos de caracteres suporta 512 caracteres. Isso significa que cada caractere possui um único valor de:**

A alternativa "D " está correta.

Se o padrão suporta 512 caracteres, significa que cada caractere possui um único valor de 9 *bits* , pois:  $2^9 = 512$ .

# VERIFICANDO O APRENDIZADO

1. ASSINALE A ALTERNATIVA QUE APRESENTA UM PADRÃO DE CÓDIGOS DE CARACTERES:

- A) SCHOOL
- B) ZYXFG
- C) Codee
- D) Unicode

2. O TRECHO DE UMA TABELA POSSUI AS SEGUINTES CODIFICAÇÕES PARA CARACTERES:

Código decimal	Caractere
65	Letra A maiúscula
77	Letra M maiúscula
79	Letra O maiúscula
82	Letra R maiúscula

COMO É A CODIFICAÇÃO DOS CARACTERES ROMA EM BINÁRIO?

- A) 1101101 0001101 0011001 1010110
- B) 1111011 0111001 1010011 0101100
- C) 1010010 1001111 1001101 1000001
- D) 1100100 0000001 0111001 0110010

3. UM TRECHO DE UMA TABELA POSSUI AS SEGUINTES CODIFICAÇÕES PARA CARACTERES:

Código octal	Caractere
104	Letra D maiúscula
105	Letra E maiúscula
122	Letra R maiúscula

COMO É A CODIFICAÇÃO DOS CARACTERES REDE EM BINÁRIO?

- A) 001010010 001000101 001000100 001000101
- B) 100100110 111001001 001101101 111001001
- C) 011111001 001010110 110110010 001010110
- D) 101001011 110011001 011000001 110011001

4. DETERMINADO PADRÃO DE CONJUNTOS DE CARACTERES SUPORTA 8.192 CARACTERES. ISSO SIGNIFICA QUE CADA CARACTERE POSSUI UM ÚNICO VALOR DE:

- A) 8 bits
- B) 1.024 bits
- C) 13 bits
- D) 65.536 bits

5. UM USUÁRIO DIGITA A LETRA “P” NO TECLADO. ELA SERÁ CONVERTIDA PARA O CÓDIGO HEXADECIMAL 70 USANDO ASCII. ISSO SIGNIFICA QUE:

- A) O caractere digitado será convertido em caractere codificado em bits .
- B) O usuário usa um programa em que, ao digitar “70”, surge uma tabela ASCII na tela.
- C) A afirmação está errada, pois a conversão é para binário, e não para hexadecimal.
- D) Faltou um comando adicional para que a letra “p” fosse devidamente codificada.

6. O TRECHO DE UMA TABELA POSSUI AS SEGUINTE CODIFICAÇÕES PARA CARACTERES:

Código hexadecimal	Caractere
41	Letra A maiúscula
43	Letra C maiúscula
53	Letra S maiúscula

COMO É A CODIFICAÇÃO DOS CARACTERES CASA EM BINÁRIO?

- A) 01000010 01000001 01001010 01011100
- B) 01101001 01101001 01101001 01101001
- C) 01101111 01111101 10001001 10100101
- D) 01000011 01000001 01010011 01000001

GABARITO

1. Assinale a alternativa que apresenta um padrão de códigos de caracteres:

A alternativa "D " está correta.

Devido à limitação na quantidade de caracteres suportados, um novo padrão internacional para a elaboração de um conjunto de códigos deles foi criado. Visando ao suporte a caracteres de inúmeros idiomas e símbolos, no Unicode, cada caractere possui um único valor: 16 bits .

2. O trecho de uma tabela possui as seguintes codificações para caracteres:

Código decimal	Caractere
65	Letra A maiúscula
77	Letra M maiúscula
79	Letra O maiúscula
82	Letra R maiúscula

Como é a codificação dos caracteres ROMA em binário?

A alternativa "C " está correta.

Convertendo os códigos decimais para binário, temos:

65 (A)						
1	0	0	0	0	0	1
2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
=	=	=	=	=	=	=
64	32	16	8	4	2	1
77 (M)						
1	0	0	1	1	0	1
2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
=	=	=	=	=	=	=
64	32	16	8	4	2	1
79 (O)						
1	0	0	1	1	1	1
2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
=	=	=	=	=	=	=
64	32	16	8	4	2	1
82 (R)						
1	0	1	0	0	1	0
2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
=	=	=	=	=	=	=
64	32	16	8	4	2	1

3. Um trecho de uma tabela possui as seguintes codificações para caracteres:

Código octal	Caractere
104	Letra D maiúscula
105	Letra E maiúscula
122	Letra R maiúscula

Como é a codificação dos caracteres REDE em binário?

A alternativa "A " está correta.

Convertendo os códigos octais para binário, temos:



1	0	4
001	000	100

1	0	5
001	000	101

1	2	2
001	010	010

**4. Determinado padrão de conjuntos de caracteres suporta 8.192 caracteres. Isso significa que cada caractere possui um único valor de:**

A alternativa "C " está correta.

Se o padrão suporta 512 caracteres, isso significa que cada caractere possui um único valor de 13 *bits* , pois  $2^{13} = 8.192$ .

**5. Um usuário digita a letra “p” no teclado. Ela será convertida para o código hexadecimal 70 usando ASCII. Isso significa que:**

A alternativa "A " está correta.

Cada símbolo possui uma correspondência com um grupo de *bits* que identifica exclusivamente aquele caractere.

**6. O trecho de uma tabela possui as seguintes codificações para caracteres:**

Código hexadecimal	Caractere
41	Letra A maiúscula
43	Letra C maiúscula
53	Letra S maiúscula

**Como é a codificação dos caracteres CASA em binário?**

A alternativa "D " está correta.

Convertendo os códigos hexadecimais para binário, temos:

4	1
0100	0001

4	3
0100	0011

5	3
0101	0011

## CONCLUSÃO

## CONSIDERAÇÕES FINAIS

O estudo de assuntos mais profundos (do ponto de vista abstrato) do computador nos permite a assimilação – ainda que de forma primária – de tudo o que ocorre tanto no nível de *hardware* quanto em todos os seus desdobramentos em *software*.

Sabendo, por exemplo, como são os sistemas de numeração binário, octal e hexadecimal, podemos compreender melhor como os tamanhos dos arquivos são medidos, de que forma as linguagens de programação podem usar a memória do computador, ou como os endereços IP em redes de computadores são construídos.

Desse modo, ao entendermos a maneira como são realizados os cálculos aritméticos, estaremos capacitados para atuar, por exemplo, em estudos e projetos de *chips*, além de dispositivos embarcados e de automação. Também estaremos habilitados a realizar uma interseção com outras áreas de estudo, como a Engenharia, a Eletrônica e as Telecomunicações.

Para ouvir um *podcast* sobre o assunto, acesse a versão online deste conteúdo.



## REFERÊNCIAS

HARRIS, D.; HARRIS, S. **Digital design and computer architecture**. 2. ed. San Francisco: Morgan Kaufmann, 2012.

HENNESSY, J. L. **Organização e projeto de computadores: a interface hardware/software**. 2. ed. Rio de Janeiro: LTC, 2000.

INMETRO. **Resumo do Sistema Internacional de Unidades - SI**. Tradução da publicação do BIPM. Consultado em meio eletrônico em: 7 abr. 2020.

MONTEIRO, M. **Introdução à organização de computadores**. 5. ed. Rio de Janeiro: LTC, 2007.

MURDOCCA, M. J.; HEURING, V. P. **Introdução à arquitetura de computadores**. Rio de Janeiro: Campus, 2000.

PATTERSON, D. A. *et al.* **Computer architecture, a quantitative approach**. 5. ed. San Francisco: Morgan Kaufmann, 2011.

PATTERSON, D. A. *et al.* **Organização e projeto de computadores**. 2. ed. Rio de Janeiro: LTC, 2000.

SEDGEWICK, R.; WAYNE, K. **Computer science** – an interdisciplinary approach. New York: Pearson Education, 2017.

STALLINGS, W. **Arquitetura e organização de computadores**. 8. ed. São Paulo: Pearson Prentice Hall, 2010.

TANEMBAUM, A. S. **Organização estruturada de computadores**. 5. ed. Rio de Janeiro: LTC, 2007.

---

## EXPLORE+

Para conhecer os conjuntos de caracteres Unicode, pesquise na internet e acesse a página **Unicode – the world standard for text and emoji**.

---

## CONTEUDISTA

Fabio Henrique Silva

Mestre em Sistemas e Computação pelo Instituto Militar de Engenharia (IME) e Graduado em Tecnologia em Sistemas para a Internet pelo Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET/RJ).

Mais informações em: Plataforma Lattes – CNPQ.