

# **Codificação e Representação da Informação**

# Introdução

As informações precisam de meios para serem exibidas. Normalmente se utiliza de modelos que imitam a realidade.

“Quando você toca um CD, pode ouvir música mesmo não havendo músico algum dentro do aparelho de CD. O disco contém um modelo eletrônico do som. Os rádios e os aparelhos de CD existem porque alguém descobriu uma maneira mecânica ou eletrônica de imitar o som, e assim construiu equipamentos capazes de reproduzir os sons. O mesmo acontece com as imagens fornecidas pela televisão e filmes.”

Peter Norton

# Representação da Informação

A base de todas as informações no computador são as combinações dos símbolos "0" e "1" (Bit – Binary Digit).

As combinações possíveis destes números através de um sistema de codificação, é o que move toda a informática, gerando números, letras, imagens, sons e movimentos.

Os sistemas de computação funcionam de modo a criar um modelo eletrônico que imita a informação real. Esse modelo é numérico e aritmético.

Alguns pontos comuns entre os equipamentos de computação e a Matemática permitem essa imitação da realidade (Numeração e Aritmética Binária).

**B**inary digi**T** = BIT é a menor unidade de informação.

Os bits servem muito bem para a representação de números mas o sistema de computação não trabalha apenas com informações numéricas, mas também com letras e símbolos.

Como representar então, letras e símbolos?

# Denominações

Nybble = 4 bits

Byte = 8 bits

Word = 16 bits

Double Word = 32 bits

Quad Word = 64 bits

# Representação de Símbolos

Na década de 60, com a necessidade dos computadores se comunicarem, surgiram padrões de codificação, que consiste em tabelas com grupos de bits que representam determinados caracteres.

O sistema binário só consegue representar números (normalmente inteiros e maiores que zero).

O sistema de computação consegue trabalhar com outros tipos de dados utilizando uma combinação de bits.

Esta combinação, através de um sistema de codificação, diferencia o tipo de dado utilizado no momento.

Os principais tipos de codificação são:

## Os principais tipos de codificação são:

- BCD;
- EBCDIC;
- ASCII
- UNICODE.



# Codificação BCD

O código padrão BCD (Binary Coded Decimal) foi um dos primeiros a serem usados e foi criado pela IBM.

A codificação BCD utiliza 6 bits para representar os caracteres, permitindo 64 ( $2^6$ ) símbolos no máximo. Esse código representa apenas letras maiúsculas mais alguns poucos símbolos. Rapidamente se tornou inadequado pela necessidade de representação de letras minúsculas.

# Codificação EBCDIC

Para resolver o problema do BCD foi criado, também pela IBM, o EBCDIC (Extend Binary Coded with Decimal Interchange Code) que usa 8 bits para representar os caracteres, permitindo 256 ( $2^8$ ) símbolos.

Essa codificação ficou restrita aos equipamentos IBM.

# Codificação ASCII

O código ASCII (American Standard Coded for Information Interchange) criado pela ANSI (American Standards Institute), tinha intuito de ser um padrão para a indústria de computadores. Foi idealizado utilizando 8 bits, dos quais 7 eram usados para representar caracteres ( $2^7 = 128$  símbolos) e um era utilizado com a função especial (bit de paridade).

Com o tempo a necessidade do bit de paridade perdeu a importância e a IBM criou um novo código ASCII, com 8 bits, que permitiu representar até 256 símbolos.

# UNICODE

Com a disseminação de computadores surgiu a necessidade de representar caracteres diferentes daqueles representados pela tabela ASCII. Para isso, um consorcio de empresas (IBM, Apple, Microsoft, Nokia, HP, America On-line, etc), criaram o Unicode, utilizando 16 bits (2 bytes) e permitindo a representação de 65.536 ( $2^{16}$ ) símbolos.

Essa quantidade de símbolos é suficiente para representar a maioria dos caracteres e símbolos em vários alfabetos (Árabe, Russo, Japonês, etc), tornando-se ideal para um padrão mundial.

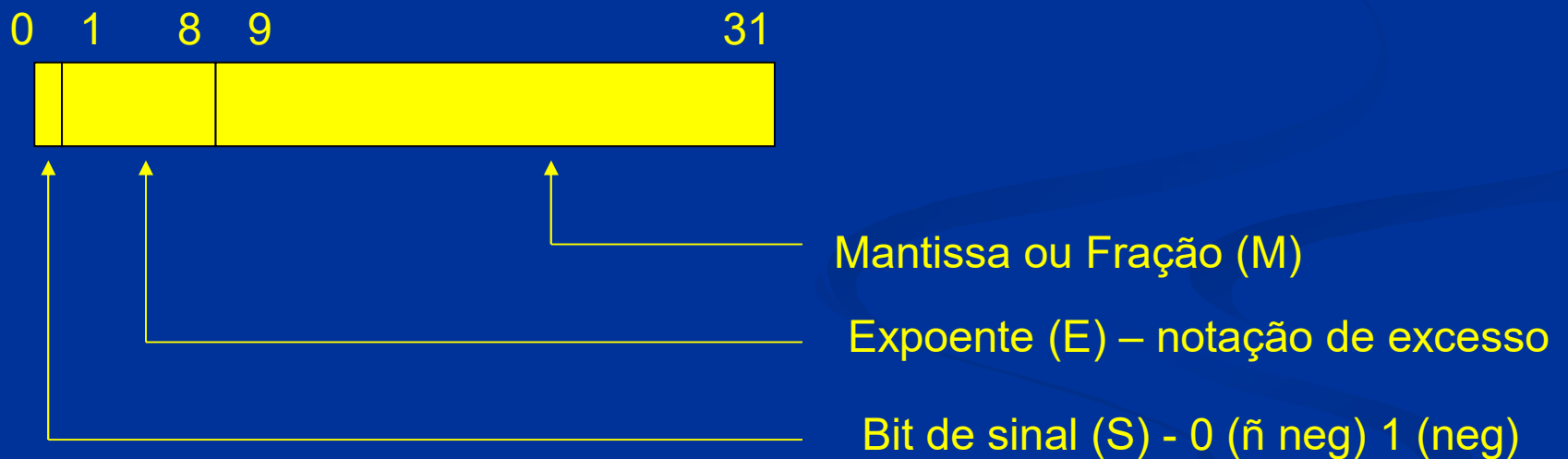
# Codificação de Valores Numéricos

# Representação de números não inteiros (Ponto Flutuante)

O Padrão 754 do ANSI/IEEE (1985) trata da representação de números na Notação Ponto Flutuante com 32 bits (precisão simples) e 64 bits (precisão dupla).

Para utilizar a notação ponto flutuante com precisão simples, é necessária uma palavra de 32 bits (numerados de 0 a 31) com as seguintes características:

# Notação de Ponto Flutuante com 32 bits (precisão simples)



# Codificação de Valores Numéricos

## Definições:

- BMS (Bit mais significativo) – é o bit mais a esquerda do número binário. Qualquer mudança no valor desse dígito ocasiona as maiores mudanças no valor representado pelo numeral binário.
- bms (bit menos significativo) – é o bit mais a direita do número binário. Mudança no valor deste dígito ocasiona menor mudança no valor representado pelo numeral binário (muda uma unidade).

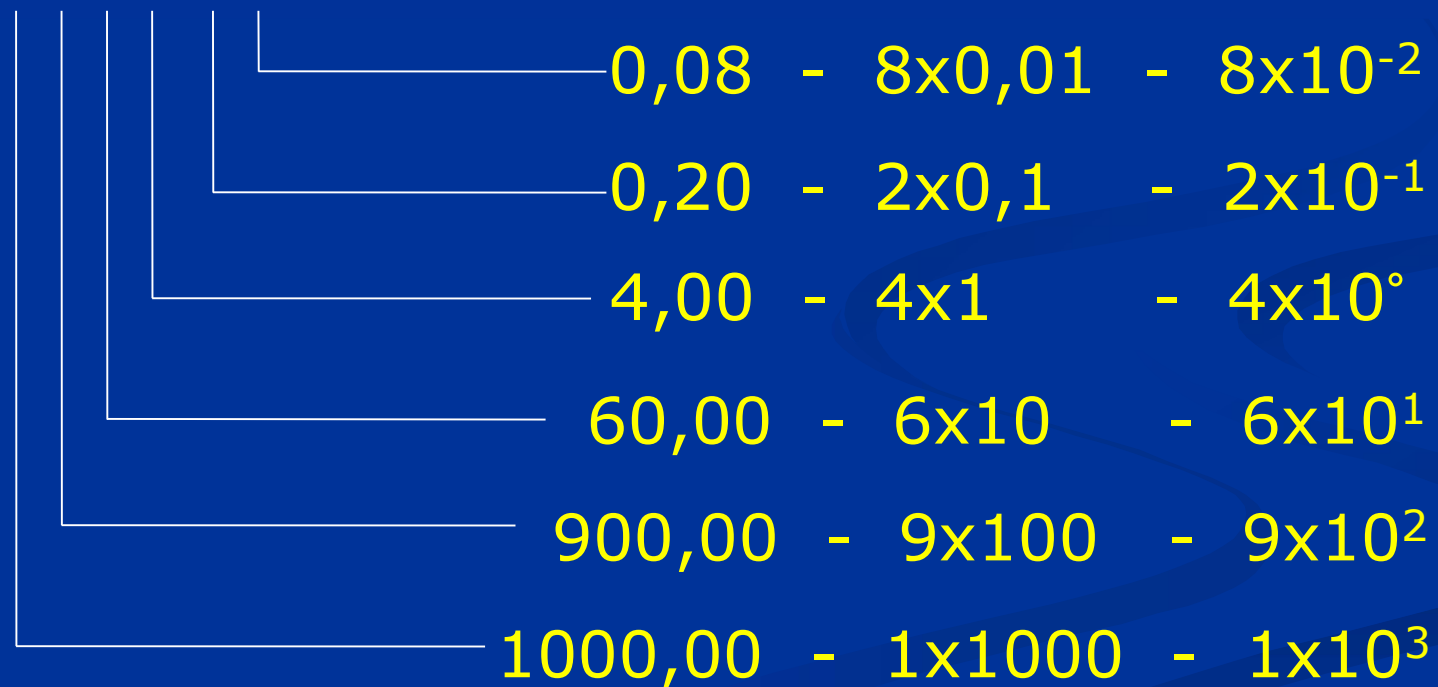


# Números binários não Inteiros

No sistema decimal utiliza-se um ponto (ou vírgula) para separar a parte inteira da parte fracionária do número.

Exemplo:

Ex.  $1964,28_{10}$       Fator de posição (Sistema Decimal)



1964,28

# Números binários não inteiros

O sistema binário segue o mesmo padrão, mas usando a base 2. A virgula é chamada Radix Point.

1011 . 101<sub>2</sub>      Fator de posição (Sistema Binário)

0,001	-	1 x 2 <sup>-3</sup>	-	1 x 1/8	-	1/8
0,00	-	0 x 2 <sup>-2</sup>	-	0 x 1/4	-	0
0,1	-	1 x 2 <sup>-1</sup>	-	1 x 1/2	-	1/2
1	-	1 x 2 <sup>0</sup>	-	1 x 1	-	1
10	-	1 x 2 <sup>1</sup>	-	1 x 2	-	2
000	-	0 x 2 <sup>2</sup>	-	0 x 4	-	0
1000	-	1 x 2 <sup>3</sup>	-	1 x 8	-	8

11 5/8

# Números Binários Inteiros (Positivos e Negativos)

Para representar números positivos e negativos binários é necessário também adotar um sistema para codificação, ou seja, haverá uma tabela na qual serão apresentadas seqüências de bits que representarão valores negativos e positivos. Existem dois métodos:

- Notação de Excesso; e
- Notação de Complemento de dois

# Notação de Excesso

1º Escolher o comprimento em bit a ser usado

2º Representar todas as combinações possíveis, iniciando pelo maior valor e decrescendo até "0"

1ª Coluna - bit de sinal

Padrão de 3 bits			
1	1	1	3
1	1	0	2
1	0	1	1
1	0	0	0
0	1	1	-1
0	1	0	-2
0	0	1	-3
0	0	0	-4

O BMS indica o sinal:

1 – não negativo

0 - negativo

Obs. Notação excesso de  $n$ , sendo  $n$  o valor decimal do padrão que representa o numero zero. No caso  $100_2$  (equivalente a 4), portanto é uma notação Excesso de 4.

A notação é conhecida por excesso, pois o valor binário normal do número é sempre o valor do excesso adicionado ao valor que ele representa. Por exemplo, no padrão de 3 bits o valor que ele representa o numeral  $011_2$ , considerando o sistema binário normal, é 3, mas como o excesso é 4, ele representa o valor -1 (ou seja  $3-4$ ).

# Conversão da notação de excesso para decimal

Converter  $001_{(2)}$  para base 10

1º Descobrir qual o excesso do padrão escolhido

$$100_{(2)} = 4_{(10)} \text{ portanto o excesso é } 4$$

2º Converter o valor para binário usando sistema binário normal

$$001_{(2)} = 1_{(10)}$$

3º Subtrair o excesso desse valor

$$1 - 4 = -3$$

# Conversão de decimal para notação de excesso

Converter  $2_{(10)}$  para base 2 notação em excesso

1º Adotar um excesso  $\rightarrow 4$

2º Descobrir o comprimento do padrão de bits escolhido.  
Basta representar o excesso em binário

$$4_{(10)} = 100_{(2)} \rightarrow \text{padrão de 3 bits}$$

3º Adicionar o excesso ao número a ser convertido

$$2 + 4 = 6$$

4º Representar esse resultado em binário usando o sistema normal

$$6_{(10)} = 110_{(2)}$$

# Notação de Complemento de Dois

1º Iniciar com um conjunto de zeros no comprimento escolhido

Padrão de 3 bits	
0 0 0	0

2º Acima desse conjunto (valores positivos) completa a seqüência até que o BMS seja igual a "0" e o restante seja "1".

Padrão de 3 bits	
0 1 1	3
0 1 0	2
0 0 1	1
0 0 0	0



# Notação de Complemento de Dois

3º Abaixo do conjunto de zeros (valores negativos) colocar um conjunto de dígitos "1" no mesmo comprimento e completa-se a seqüência decrescente até que o BMS seja "1" e o restante "0".

O BMS indica o sinal:

0 – positivo ou zero

1 - negativo

Padrão de 3 bits	
0 1 1	3
0 1 0	2
0 0 1	1
0 0 0	0
1 1 1	-1
1 1 0	-2
1 0 1	-3
1 0 0	-4

# Representação de Imagens

# Imagens

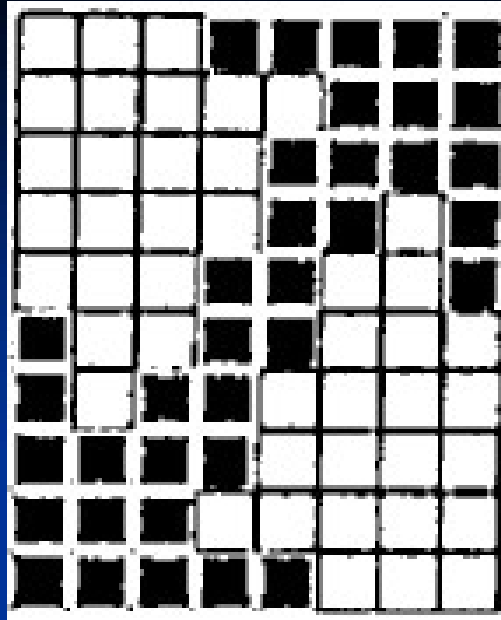
Para que uma imagem possa ser trabalhada no computador é necessário adequá-la ao modelo de numeração binária.

Assim, associa-se

- ao bit 0
- ao bit 1



Podemos imaginar a imagem preto e branco acima como sendo formada por uma série de pontos pretos e pontos brancos.  
A figura a seguir demonstra isso.



Para que esta imagem possa ser trabalhada usando um computador, é necessário adequá-la ao modelo de numeração binária, como é feito com os caracteres.

Os quadrados pretos são associados ao bit 0 e os brancos ao bit 1, como mostra a figura a seguir.

1	1	1	1	0	0	0	0
1	1	1	1	1	0	0	0
1	1	1	1	0	0	0	0
1	1	1	1	0	0	1	0
1	1	1	0	0	1	1	0
0	1	1	0	0	1	1	1
0	1	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	1	1	1	1	1
0	0	0	0	0	1	1	1

Como cada grupo de 8 bits é 1 byte e a figura tem 10 linhas de 1 byte, o tamanho total da figura é 10 bytes.

No caso de uma imagem colorida, deve-se levar em consideração que a imagem continuará sendo representadas por pontos, mas eles assumirão cores diferentes.

# Imagens Coloridas

A tela do computador é formada por vários pontos (pixels) e são estes que produzem as imagens (representadas pelos bits "0" e "1") e podem assumir cores diferentes. Para representar as cores que vão aparecer, monta-se uma tabela, na qual uma combinação de bits representa uma determinada cor.

O número de bits utilizados para representar as cores de cada pixel é conhecido como Profundidade de Cor.

Quanto mais bits por pixel, mais cores serão representadas e maiores definição e qualidade.

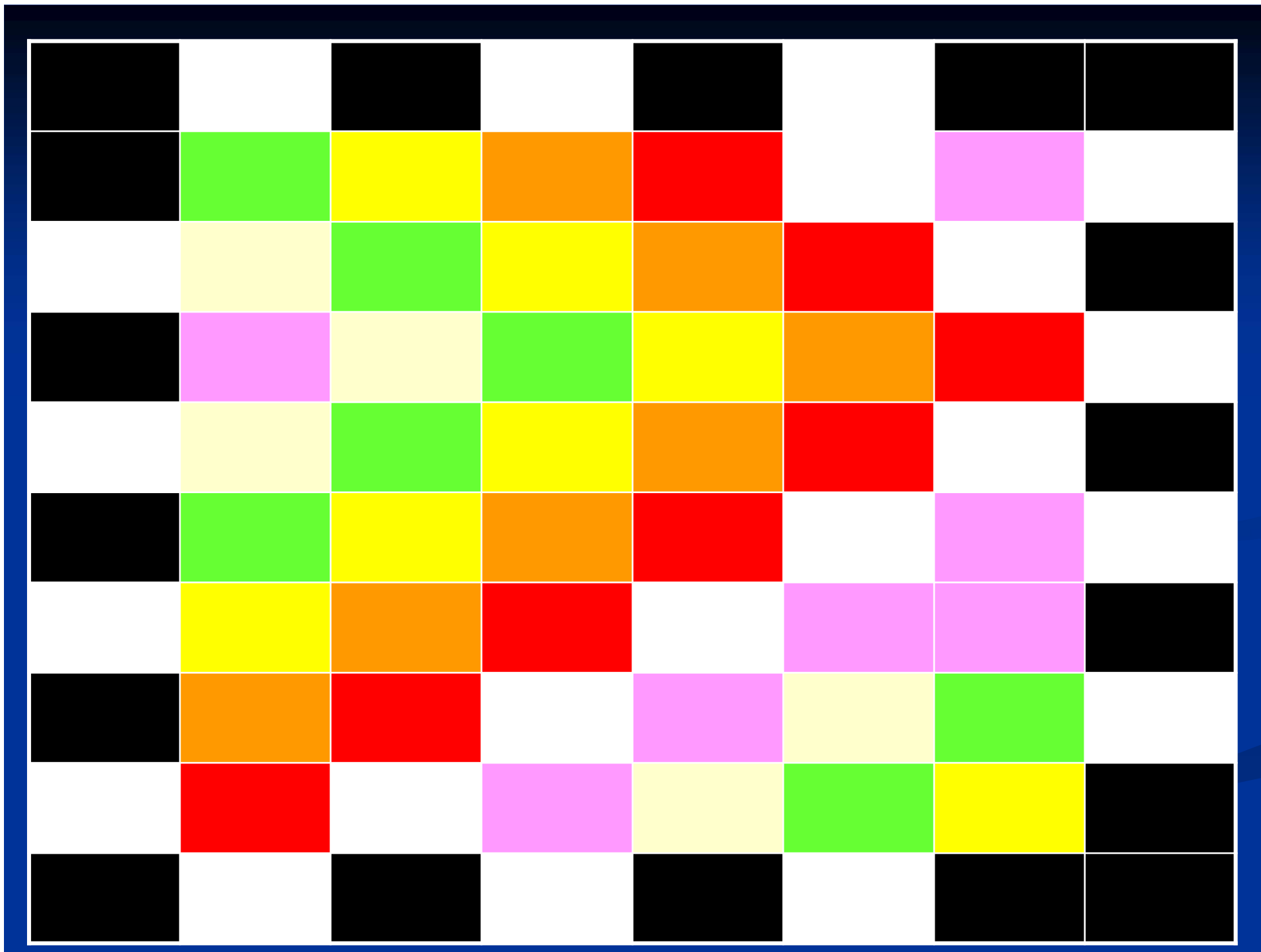


# Tabela de decodificação de cores com 3 bits

preto	000	
vermelho	001	
laranja	010	
amarelo	011	
verde	100	
azul	101	
violeta	110	
branco	111	

111	111	111	001	001	001	001	001
111	111	111	111	111	001	001	001
111	111	111	111	001	001	001	001
111	111	111	111	001	001	111	001
111	111	111	001	001	111	111	001
001	111	111	001	001	111	111	111
001	111	001	001	111	111	111	111
001	001	001	001	111	111	111	111
001	001	001	111	111	111	111	111
001	001	001	001	001	111	111	111

000	111	000	111	000	111	000	000
000	100	011	010	001	111	110	111
111	101	100	011	010	001	111	000
000	110	101	100	011	010	001	111
111	101	100	011	010	001	111	000
000	100	011	010	001	111	110	111
111	011	010	001	111	110	101	000
000	010	001	111	110	101	100	111
111	001	111	110	101	100	011	000
000	111	000	111	000	111	000	000



# Profundidade de cor

- É o número de bits utilizados para representar as cores de cada ponto (pixel).
- Neste caso temos 3 bits de profundidade de cor.
- 1 ponto – 3 bits
- 1 linha – 8 pontos – 24 bits
- Figura – 10 linhas – 240 bits – 30 bytes

# Tipos de Arquivos de Imagens

## **BMP (Windows Bitmap)**

O tamanho da imagem é diretamente proporcional ao seu tamanho e a profundidade de cor.

Ex. Imagem com 800x600 pontos com profundidade de cor de 24 bits:

Tamanho =  $800 \times 600 \times 24 = 1,44$  MBytes

# Tipos de Arquivos de Imagens

## **GIF (Graphics Interchange Format)**

Formato que utiliza um algoritmo de compressão de dados LZM, consegue apresentar a mesma imagem com tamanho menor (lossless). Mantém a mesma qualidade porém para obter a compressão reduz o número de cores possíveis por imagem (até 256), causando distorção em relação a imagem original. Este formato é ideal para imagens que não possuem grandes variações de cores.

# Tipos de Arquivos de Imagens

## **JPEG (Joint Photographic Experts Group)**

Formato que também utiliza um algoritmo de compressão de dados, porém produz perda de qualidade (lossy). Em compensação, o formato JPEG utiliza até 16 milhões de cores por imagem, que pode compensar a perda de qualidade. Este formato é ideal para imagens de cenas reais com variações sutis na tonalidade e na cor.



# Vídeo

Um vídeo nada mais é do que várias imagens apresentadas sucessivamente em uma determinada velocidade.

É uma sucessão de imagens estáticas apresentadas em uma determinada velocidade.

É dessa forma que o sistema de computação deve trabalhar para que seja possível apresentar um vídeo ou um filme.

# Imagens em Movimento

Várias imagens são apresentadas sucessivamente em uma determinada velocidade, com pequenas variações de movimento entre uma e outra, os nossos olhos são enganados e entendem que a imagem está em movimento (como no cinema). Portanto, um vídeo é uma sucessão de imagens estáticas, apresentadas em seqüência numa determinada velocidade. Desta forma, para o sistema de computação um vídeo nada mais é que uma coleção de imagens.

A partir desta constatação é possível realizar o seguinte cálculo:

Cada quadro do filme > 800x600 pontos (tela inteira)

Profundidade de cor > 24 bits (16 milhões de cores)

Tamanho quadro =  $800 \times 600 \times 24 = 11.520.000$   
bits =  $1.440.000 \text{ B} \sim 1,4 \text{ MB}$

Velocidade = 30 quadros por segundo

$1,4 \times 30 \rightarrow$  1 segundo de filme tem 42 MB de tamanho

Filme de 2 horas  $\rightarrow 7.200 \times 42 \text{ MB} = 302.400 \sim 302 \text{ GB}$

# Imagens em Movimento (Vídeos)

Vê-se que, desse modo, o tamanho do filme é exageradamente grande e é difícil arranjar meios de armazenamento para tanta informação.

Para amenizar este problema usa-se o Codificador/Decodificador (CODEC), dispositivo que codifica os dados referentes a uma imagem em movimento para que ela seja transmitida ou armazenada. O arquivo é comprimido para facilitar o armazenamento e a manipulação dessas imagens em movimento.

Os Codecs de vídeo mais conhecidos são os: MPEG (Moving Picture Experts Group), QuickTime, AVI (Audio Video Interleave) e DivX.

# CODEC

CoDec é o acrônimo de Codificador/Decodificador, dispositivo de *hardware* ou *software* que codifica/decodifica sinais.

Os codecs são utilizados para codificar os dados referentes a um vídeo, uma imagem em movimento, um som para que ela seja comprimida e facilite sua transmissão ou armazenamento.

Existem dois tipos de codecs:

- Sem perdas (lossless)

- Com perdas (lossy)

# CODECs sem perdas

Os codecs sem perdas são codecs que codificam som ou imagem com a finalidade de comprimir o arquivo (diminuir seu tamanho) sem alterar o som ou imagem originais. Se o arquivo for descomprimido, o novo arquivo será idêntico ao original.

Esse tipo de codec normalmente gera arquivos codificados que são entre 2 a 3 vezes menores que os arquivos originais. São muito utilizados em rádios e emissoras de televisão para manter a qualidade do som ou imagem.

Exemplos desse tipo de codec para som são o *flac*, *shorten*, *wavpack* e monkey's audio. Para vídeo o HuffYUV, MSU, *MJPEG*, H.264 e FFmpeg Video 1. Para imagens o PNG e *TIFF*.

# CODECs com perda

Codecs com perdas são codecs que codificam som ou imagem, gerando uma perda de qualidade com a finalidade de alcançar maiores taxas de compressão. Essa perda de qualidade é balanceada com a taxa de compressão para que não sejam perceptíveis.

Por exemplo, se um instrumento muito baixo toca ao mesmo tempo que outro instrumento mais alto, o primeiro é suprimido, já que dificilmente será ouvido. Nesse caso, somente um ouvido bem treinado pode identificar que o instrumento foi suprimido.

Esses Codecs foram criados para comprimir os arquivos de som ou imagem a taxas de compressão muito altas. Por exemplo, o Vorbis e o MP3 são codecs para som que facilmente comprimem o arquivo de som em 10 a 12 vezes o tamanho original, sem gerar ruídos significativos.

Exemplos de codecs com perdas para som são o Ogg Vorbis, MP3, AC3 e WMA. Para vídeo, temos o Xvid, DivX, RMVB, WMV, Theora e Sorenson. E para imagens temos o JPEG, JPEG 2000 e GIF.



# Alguns Codecs

MPEG - Utilizam a técnica de armazenar somente os quadros principais da imagem em movimento e prevêem como seriam os quadros intermediários analisando as alterações ocorridas.

QuikTime – apresenta imagens compactadas de alta qualidade.

AVI (Audio Video Interleave) – armazena os dados de um quadro e, a partir dele, armazena apenas o que é diferente nos quadros que vêm em seguida.

DivX – Codec de vídeo criado pela DivX Inc que permite que um filme gravado em DVD, com aproximadamente 4 GB, passe a ter em torno de 600 MB. É semelhante ao MP3 de audio.

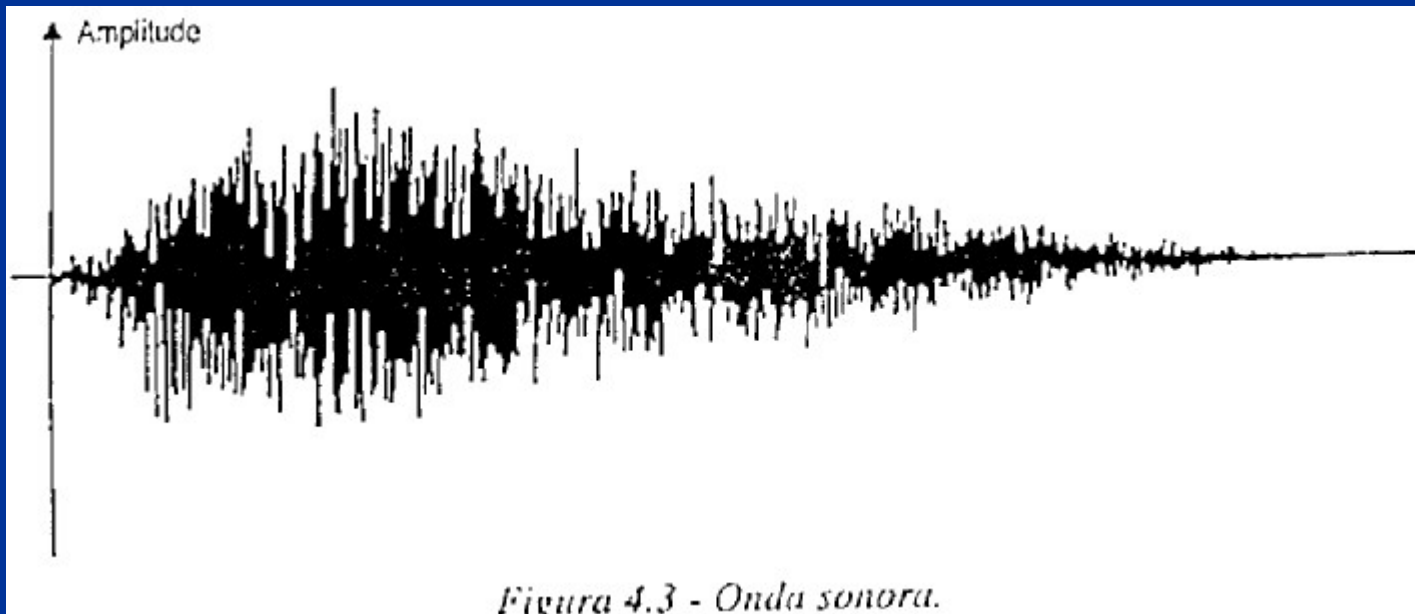
# Representação de Som

# Representação dos Sons

Os sons são ondas que trafegam pelo ar e chegam aos ouvidos ou ao microfone do sistema de computação. Essa onda não apresenta números binários, mas valores analógicos com amplitude variável, ou seja, a intensidade do som é maior ou menor ao longo da sua duração.

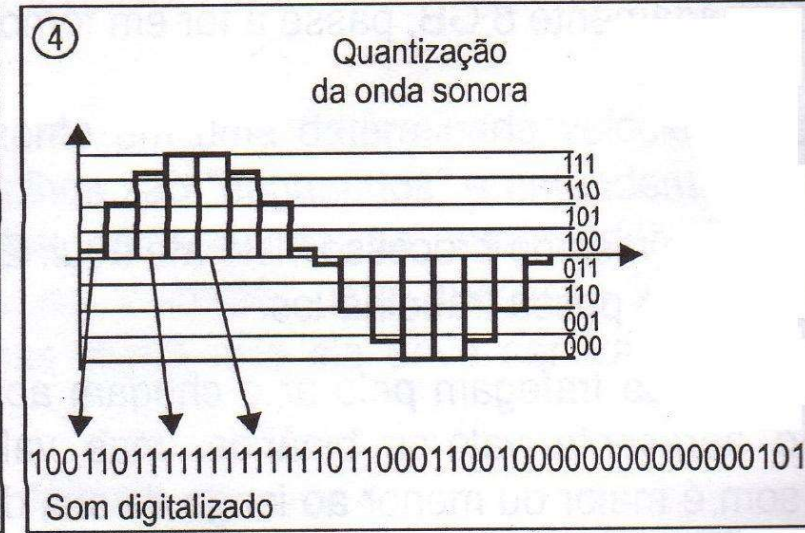
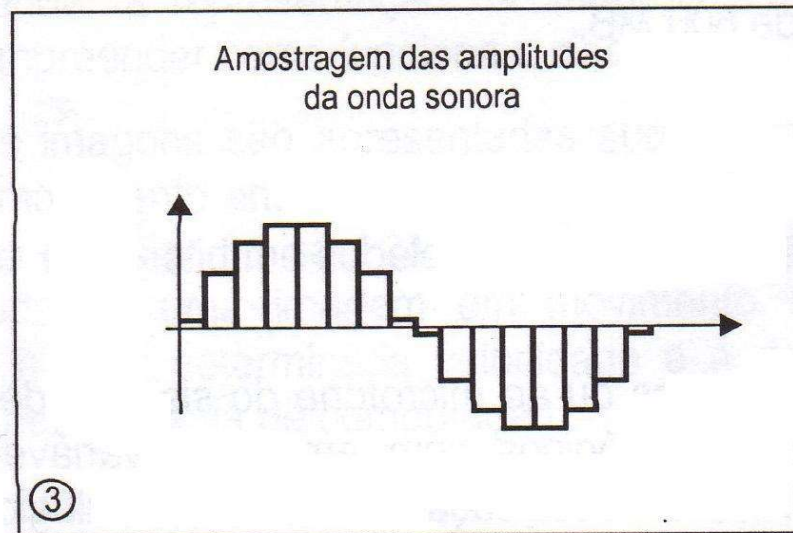
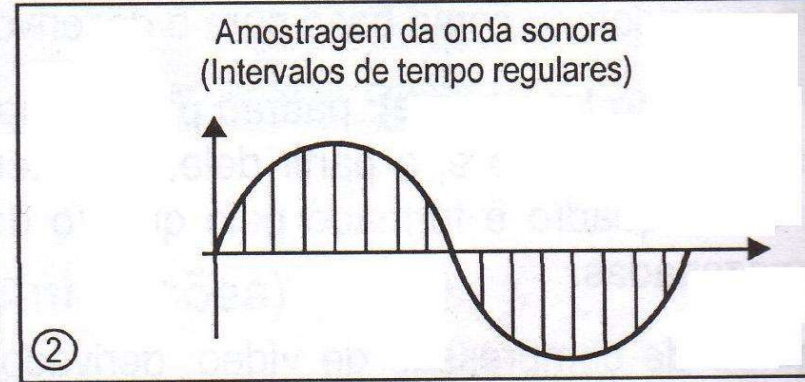
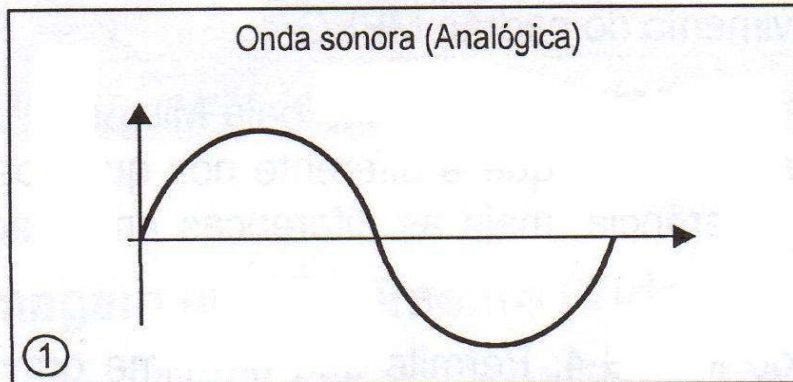
A Digitalização do Som é a codificação desse sinal analógico em digital. Para fazer isto, aplica-se um conceito muito parecido com o usado para caractere. A faixa de amplitude – desde a mais alta até a mais baixa – é dividida em pequenas faixas, cada uma equivale a uma combinação de caractere.

# Onda sonora



# **Digitalização de som**

Na digitalização de som, como visto na representação de caracteres, quanto maior o número de bits maior o número de combinações possíveis e portanto maior o número de faixas de amplitudes.



O número de bits vai determinar o nº de faixas, portanto quanto maior o nº de bits, mais fiel o modelo será em relação ao original. Neste exemplo utilizamos 8 faixas (3 bits).

# Considerações sobre digitalização de sons

Taxa de Amostragem (frequência do sinal):

- 10 KHz – equivalente a voz telefone ou rádio AM;
- 20KHz - som e musica equivalente a radio FM:
- 40 KHz - som equivalente a CDs.

Nº de Bits

- 08 bits - equivalente a voz telefone ou emissora de rádio AM (haverá uma pequena distorção sonora);
- 16 bits – para qualquer outro tipo elimina distorções;

Obs. Na digitalização, dobrar o tamanho da taxa de amostragem ou dobrar o nº de bits, equivale também ao dobro do tamanho no resultado final.



# Representação dos Sons

Esta tabela mostra a quantidade de KB necessários para armazenar 01 segundo de som de acordo com as diferentes taxas de amostragens e quantidade de bits:

Formato	KB por segundo	Tempo/MB
44 KHZ, 16 bits, estéreo	176 KB/s	5,8 s
44 KHZ, 16 bits, mono	88 KB/s	11,6 s
22 KHz, 16 bits, estéreo	88 kb/S	11,6 s
22 KHz, 16 bits, mono	44 KB/s	23,3 s
22 KHz, 8 bits, estéreo	44 KB/s	23,3 s
22 KHz, 8 bits, mono	22 KB/s	46,5 s
11 KHz, 8 bits, estéreo	22 KB/s	46,5 s
11 khZ, 8 BITS, mono	11 KB/s	93 s

# Representação dos Sons

Utilizando a tabela e considerando uma digitalização realizada com o formato 44 KHz, 16 bits e estéreo (melhor qualidade), é possível calcular:

1 MB > 5,8 segundos

1 música = 4 minutos (240 segundos)

Tamanho final =  $240 \div 5,8 = 41$  MB

Obs. A digitalização neste nível de qualidade resulta em tamanhos da ordem de 40 MB, o que, em alguns casos é excessivo. Então, são utilizados CODECs para compressão digital de áudio que reduz para aproximadamente 10% do valor original, como MP3, WMA, Ogg Vorbis, e AAC.

# Organização de arquivos

# Nomes de arquivo

Os Sistemas Operacionais atuais permitem que um determinado arquivo seja referenciado por um nome seguindo regras para sua formação.

Nos ambientes gráficos como o Windows ou MAC OS X, por exemplo, os nomes de arquivos podem conter até 256 caracteres seguidos de uma extensão de 3 caracteres. Nestes ambientes, além do nome, os arquivos têm uma representação gráfica chamada ícone.

# Nomes de Arquivos e Pastas

**trabalho.doc**  
nome      extensão

- O nome pode possuir até 256 caracteres
- Extensões geralmente apresentam 3 caracteres:
  - documento do Word: **.doc**
  - figura bitmap: **.bmp**
  - planilha do Excel: **.xls**
  - apresentação do PowerPoint: **.ppt**

# Organização das Informações

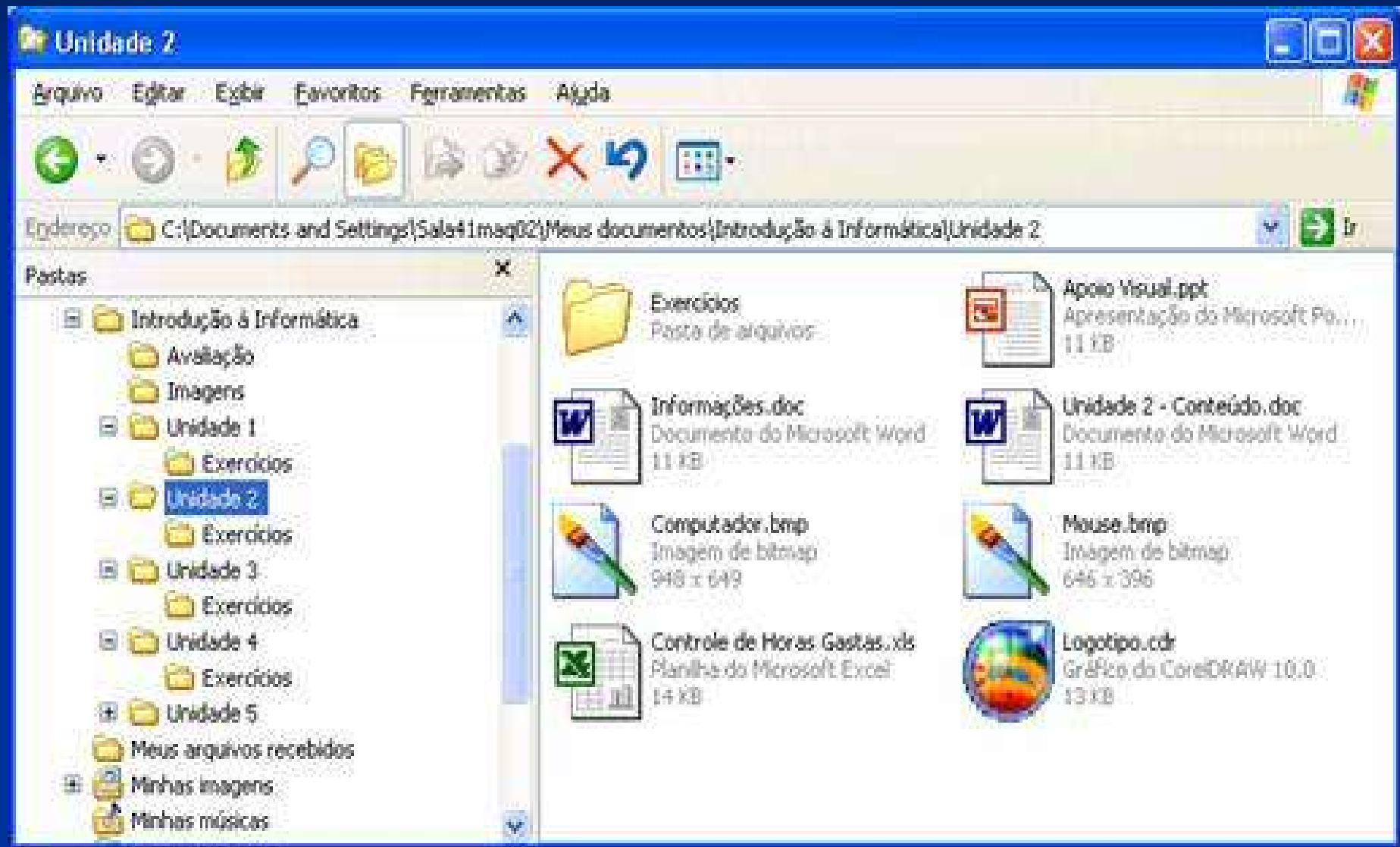
## Tipos de Arquivos

- Arquivos executáveis (.EXE .COM)
- Arquivos de banco dados (.MDB .DBF)
- Arquivos de Som (.WAV .MID .MP3)
- Arquivos de Imagens (.TIF .GIF .BMP)
- Arquivos de Vídeo (AVI, MPEG, etc)
- Arquivos Gerais (.PPT .XLS .DOC)

# Como identificar arquivos

- Para identificar um arquivo é preciso saber a sua origem, ou seja, qual o programa que o gerou e em que sistema operacional ou computador ele está armazenado.
- Uma carta produzida com o processador de textos Word da Microsoft gera arquivos contendo no nome a extensão .DOC. Planilhas financeiros produzidos com o Excel da Microsoft geram arquivos .XLS e assim por diante.

# Nome de arquivo





# Arquivos de programas

- Arquivos de programas (executáveis) equivalem às aplicações de um computador que por sua vez são programas escritos para serem executados por um computador.
- Normalmente, a extensão de arquivos executáveis é .EXE e .COM.

# Arquivos de dados

- Um fichário eletrônico, um conjunto de páginas de um livro e um cadastro de CDs podem ser considerados arquivos de dados.
- As extensões: .DAT, .MDB e .DBF, são as mais comuns para nomes de arquivos de dados.

# Arquivos de som

- Existem diversos tipos de arquivos de som. Alguns comportam apenas voz, outros música de qualidade de CD, outros músicas geradas por instrumentos musicais eletrônicos, etc.
- As extensões: .WAV, .MID e .MP3 são as mais comuns para arquivos de som.

# Arquivos de imagens

- Arquivos de imagens são formados por imagens estáticas (fotografias, desenhos, gráficos, etc.) ou imagens dinâmicas (vídeo, animação).
- As extensões .BMP, .TIF, .GIF e .JPG são as mais comuns para nomes de arquivos de imagens estáticas.
- As extensões .AVI, .MPG e .MOV são as mais comuns para nomes de arquivos para imagens dinâmicas (videos).

# ORGANIZAÇÃO DAS INFORMAÇÕES

Dispositivo	Capacidade	Velocidade	Custo aproximado
CD-ROM	700 MB	média	R\$ 1,00
DVD-R	4,7 GB	média-alta	R\$ 1,50
Pen Drive	16 GB	alta	R\$ 50,00
disco rígido	1 TB	alta	R\$ 500,00

# Estruturas de Diretórios de SO Windows

