

# Sistemas de Computação

Mestrado Integrado em Engenharia e Gestão de  
Sistemas de Informação

2020/2021

## Dados (I)

- **Computadores**
  - Processam e armazenam todos os tipos de dados em binário;
- **Comunicação entre pessoas**
  - Linguagem, imagens, sons;
- **Formatos dos dados**
  - Existem especificações para converter os dados em algo usável pelo computador;
  - Define as diferentes formas em que os dados usados pelas pessoas pode ser representados, guardados e processados num computador.
  - Existem variados dispositivos de entrada (*input*) que podem ser usados. A escolha dos dispositivos depende do formato original dos dados e da representação dos dados que se quer ter no computador.
    - Exemplo: teclado é algo simples e directo – cada tecla gera um código binário que é identificado como uma representação de um dado carácter
    - Um som já é mais complicado de representar, dado que é algo que muda de forma contínua ao longo do tempo.

## Dados (II)

- Binário
  - É de natureza discreta;
  - Exemplo: teclado, em que o utilizador carregar no "A"
  - O teclado gera um código binário para cada tecla
- Analógico
  - Dados em contínuo, como o som e as imagens
  - É preciso hardware próprio para converter os dados em números binários;

## Dados (III)

- Metadados
  - Informação que descreve ou permite interpretar o significado dos dados
  - Por exemplo, para representar uma imagem é preciso saber:
    - Número de cores representado em cada ponto da imagem;
    - O método usado para representar a cor;
    - O número de pontos que constituem a imagem em termos de largura e altura;
- Cada programa pode guardar e processar os dados no formato que desejar
  - Exemplo: Word e WordPerfect são dois processadores de texto que guardam os dados de forma diferente;
  - Os formatos usados por cada programa são chamados de **formatos proprietários**;
- Representações normalizadas (*standard*) de dados
  - Permite a interacção:
    - Diferentes programas;
    - Entre os programas e os dispositivos de I/O;
    - Entre diferentes peças de hardware;
    - Entre sistemas que partilham dados através das redes ou de dispositivos transportáveis como CD-ROMs, memórias USB, etc.
  - Os dados tem de ser reconhecidos por uma grande variedade de hardware e de software para permitir que possam ser usados por utilizadores com diferentes sistemas computacionais

# Dados (IV)

- Representação de dados usuais

Tipo de dados	Standards
Alfa numéricos	Unicode, ASCII, EBDCDIC

Tipo de dados	Standards
Som	WAV – Waveform Audio File Format AVI – Audio Video Interleave MP3 – MPEG-1 Audio Layer-3 MIDI – Musical Instrument Digital Interface WMA – Windows Media Audio
Descrição de páginas	PDF – Adobe Portable Document Format HTML – HyperText Markup Language XML – Extensible Markup Language
Video	Quicktime MPEG-2 ou MPEG-4 RealVideo WMV - Windows Media Video DivX - Digital Video Express

## Representação interna de dados

- Reflecte:
  - Complexidade da fonte
  - Tipos de processamento necessário
- Compromissos:
  - Precisão e resolução
    - Exemplo: simples foto vs pintura num livro de arte
  - Densidade (arquivo e transmissão)
    - É necessária uma maior quantidade de dados para aumentar a precisão e resolução
    - Compressão permite representar os dados de forma mais compacta
    - Metadados: dados que descrevem ou interpretam o significado dos dados
  - Facilidade da manipulação
    - Exemplo: processar um áudio simples vs um som de alta fidelidade
  - Normalização
    - Formatos proprietários para arquivar e processar os dados (exemplo: word)
    - Normas / standards de facto: normas proprietárias baseadas em sistemas aceitáveis de forma generalizada (exemplo: *PostScript*)

# Tipos de dados: alfa-numéricos

- Alfa-numéricos
  - Caracteres b H p
  - Dígitos numéricos: 7 9 3
  - Sinais de pontuação: ! , ?
  - Caracteres com significado especial: \$ £ &
- Caracteres numéricos vs números
  - Ambos são introduzidos como caracteres normais
  - O computador “converte” para números para efeitos de cálculos
  - Tratados como caracteres se forem processados como texto
    - Exemplo: número de telefone, código-postal

## Códigos alfa-numéricos

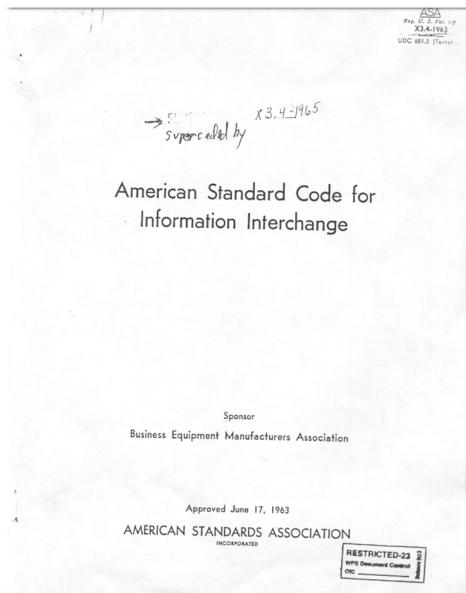
- Conjunto de bits representam um carácter;
- Valor do número binário que representa o carácter respeita a ordem no alfabeto
  - Facilita a ordenação e a pesquisa

# Representação de caracteres

- ASCII – Esquema de codificação mais utilizado
  - EBCDIC – Sistemas IBM (legado)
  - Unicode – Desenvolvido para utilização universal

## ASCII (I)

- ASCII - *American Standard Code for Information Interchange*
  - Desenvolvido pela ANSI – *American National Standards Institute*
  - Década 1960
  - Representar textos em computadores e equipamentos de comunicação
  - Representa
    - Alfabeto latino
    - Numeração árabe
    - Caracteres de pontuação *standard*
    - Pequeno conjunto de acentos e outros caracteres usados em línguas europeias
  - Código de 7 bits: 128 caracteres
  - Sistemas de codificação atuais (modernos) baseiam-se no ASCII



## ASCII (II)

MSD LSD	0	1	2	3	4	5	6	7
0	NUL	DLE	SP	0	@	P		p
1	SOH	DC1	!	1	A	Q	a	w
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	\$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACJ	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	-	o	DEL

## ASCII (III)

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	Ø	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	\$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	Ø	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	Ø	127	7F	□

## ASCII (III)

Dec	Hex	Char									
128	80	Ҫ	160	A0	ܾ	192	C0	ܼ	224	E0	ܾ
129	81	ܻ	161	A1	ܵ	193	C1	ܼ	225	E1	ܼ
130	82	ܵ	162	A2	ܶ	194	C2	ܼ	226	E2	ܼ
131	83	ܵ	163	A3	ܻ	195	C3	ܼ	227	E3	ܼ
132	84	ܵ	164	A4	ܻ	196	C4	ܼ	228	E4	ܼ
133	85	ܵ	165	A5	ܻ	197	C5	ܼ	229	E5	ܼ
134	86	ܵ	166	A6	ܼ	198	C6	ܼ	230	E6	ܼ
135	87	ܼ	167	A7	ܼ	199	C7	ܼ	231	E7	ܼ
136	88	ܼ	168	A8	ܼ	200	C8	ܼ	232	E8	ܼ
137	89	ܼ	169	A9	ܼ	201	C9	ܼ	233	E9	ܼ
138	8A	ܼ	170	AA	ܼ	202	CA	ܼ	234	EA	ܼ
139	8B	ܼ	171	AB	ܼ	203	CB	ܼ	235	EB	ܼ
140	8C	ܼ	172	AC	ܼ	204	CC	ܼ	236	EC	ܼ
141	8D	ܼ	173	AD	ܼ	205	CD	ܼ	237	ED	ܼ
142	8E	ܼ	174	AE	ܼ	206	CE	ܼ	238	EE	ܼ
143	8F	ܼ	175	AF	ܼ	207	CF	ܼ	239	EF	ܼ
144	90	ܼ	176	BO	ܼ	208	DO	ܼ	240	FO	ܼ
145	91	ܼ	177	B1	ܼ	209	D1	ܼ	241	F1	ܼ
146	92	ܼ	178	B2	ܼ	210	D2	ܼ	242	F2	ܼ
147	93	ܼ	179	B3	ܼ	211	D3	ܼ	243	F3	ܼ
148	94	ܼ	180	B4	ܼ	212	D4	ܼ	244	F4	ܼ
149	95	ܼ	181	B5	ܼ	213	D5	ܼ	245	F5	ܼ
150	96	ܼ	182	B6	ܼ	214	D6	ܼ	246	F6	ܼ
151	97	ܼ	183	B7	ܼ	215	D7	ܼ	247	F7	ܼ
152	98	ܼ	184	B8	ܼ	216	D8	ܼ	248	F8	ܼ
153	99	ܼ	185	B9	ܼ	217	D9	ܼ	249	F9	ܼ
154	9A	ܼ	186	BA	ܼ	218	DA	ܼ	250	FA	ܼ
155	9B	ܼ	187	BB	ܼ	219	DB	ܼ	251	FB	ܼ
156	9C	ܼ	188	BC	ܼ	220	DC	ܼ	252	FC	ܼ
157	9D	ܼ	189	BD	ܼ	221	DD	ܼ	253	FD	ܼ
158	9E	ܼ	190	BE	ܼ	222	DE	ܼ	254	FE	ܼ
159	9F	ܼ	191	BF	ܼ	223	DF	ܼ	255	FF	ܼ

Fonte: <http://www.cdrummond.qc.ca/cegep/informat/professeurs/alain/files/ascii.htm>

- ISO-8859-N
    - ISO-8859-1 ... ISSO-8859-16
    - Ano: 1987-1998
  - Windows-1252
    - Microsoft
    - Confunde-se (erradamente) com o ISSO-8859-1
    - Versão 1: Microsoft Windows 1.0

# EBCDIC (I)

- EBCDIC - *Extended Binary Coded Decimal Interchange Code*
  - Desenvolvido pela IBM
  - Usado essencialmente pela IBM e por alguns sistemas “compatíveis com IBM”;
  - É possível a conversão de/para ASCII
  - Códigos dos caracteres são diferentes no ASCII e no EBCDIC

# EBCDIC (II)

EBCDIC character codes															
		1st hex digit													
		2nd hex digit													
0	NUL	DLE	DS		SP	&	-								0
1	SOH	DC1	SOS			/		a	j			A	J		1
2	STX	DC2	FS	SYN				b	k	s		B	K	S	2
3	ETX	TM						c	l	t		C	L	T	3
4	PF	RES	BYP	PN				d	m	u		D	M	U	4
5	HT	NL	LF	RS				e	n	v		E	N	V	5
6	LC	BS	ETB	UC				f	o	w		F	O	W	6
7	DEL	IL	ESC	EOT				g	p	x		G	P	X	7
8		CAN						h	q	y		H	Q	Y	8
9		EM						i	r	z	'	I	R	Z	9
A	SMM	CC	SM		CENT	!	:								
B	VT	CUI	CU2	CU3		\$	,	#							
C	FF	IFS		DC4	<	*	%	@							
D	CR	IGS	ENQ	NAK	(	)	-	'							
E	SO	IRS	ACK		+	;	>	=							
F	SI	IUS	BEL	SUB		--	?	"							

# Unicode (I)

- A norma internacional ISO/IEC 10646 define o **Conjunto Universal de Caracteres (UCS)**
  - Unicode Consortium trabalha em conjunto com a ISO na definição dos caracteres
- Universal Coded Character Set (UCS)
- Unificação dos vários sistemas existentes
- ASCII Latin-1 é um subconjunto do Unicode
  - Corresponde aos caracteres 0-255 da tabela Unicode
- Suporta múltiplas línguas
  - Suporta quase os caracteres de todos os alfabetos
  - Suporta os símbolos usados nas línguas chinesas, japonês e coreano
  - Suporta caracteres “compostos” usados em algumas línguas
- Versão 12.1
  - Maio 2019
  - 137994 caracteres
- Versão 13.0
  - Março 2020
  - 143859 caracteres



# Unicode (II)

- Implementado por diferentes esquemas de codificação
  - UTF-8
  - UTF-16 / UCS-2
    - 65536 caracteres
  - UTF-32
    - UCS-4
    - 4 bytes
    - Menos usado que o UTF-8 e UTF-16 porque necessita de espaço adicional

# Unicode (III)

- Alfabeto cirílico

- As letras do cirílico, incluindo alfabetos nacionais e históricos, são representados pelos seguintes blocos:
- Cirílico U+0400–U+04FF

	040	041	042	043	044	045	046	047	048	049	04A	04B	04C	04D	04E	04F
0	Ѐ	Ӑ	Ӗ	Ҫ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ
1	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ
2	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ
3	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ
4	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ
5	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ
6	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ
7	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ
8	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ
9	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ
A	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ

Fonte: <http://www.unicode.org/>

# Unicode (IV)

- Alfabeto árabe

- Árabe base: U+0600–U+06FF
- Suplemento
- Extensão-A
- Apresentação

	060	061	062	063	064	065	066	067	068	069	06A	06B	06C	06D	06E	06F
0	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ
1	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ
2	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ
3	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ
4	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ
5	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ
6	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ
7	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ
8	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ
9	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ
A	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ	ؠ

Fonte: <http://www.unicode.org/>

# Unicode (V)

- Matemática (operadores)
  - Código U+2200–U+22FF

	220	221	222	223	224	225	226	227	228	229	22A	22B	22C	22D	22E	22F
0	∀	∏	∠	∤	϶	÷	≠	≯	≮	⊓	⊓	⊓	⊓	⊓	⊓	⊓
1	⅀	⅀	⅀	⅀	⅀	⅀	⅀	⅀	⅀	⅀	⅀	⅀	⅀	⅀	⅀	⅀
2	∅	–	△	∅	϶	϶	϶	϶	϶	⊓	⊓	⊓	⊓	⊓	⊓	⊓
3	Ǝ	Ǝ	Ǝ	Ǝ	Ǝ	Ǝ	Ǝ	Ǝ	Ǝ	⊓	⊓	⊓	⊓	⊓	⊓	⊓
4	∅	+	+	∅	϶	϶	϶	϶	϶	⊓	⊓	⊓	⊓	⊓	⊓	⊓
5	∅	/		∅	϶	϶	϶	϶	϶	⊓	⊓	⊓	⊓	⊓	⊓	⊓
6	Δ	＼	≠	⋮	϶	϶	϶	϶	϶	⊓	⊓	⊓	⊓	⊓	⊓	⊓
7	∇	*	Δ	⋮	϶	϶	϶	϶	϶	⊓	⊓	⊓	⊓	⊓	⊓	⊓
8	϶	◦	∨	–	϶	϶	϶	϶	϶	⊓	⊓	⊓	⊓	⊓	⊓	⊓
9	϶	•	∩	–	϶	϶	϶	϶	϶	⊓	⊓	⊓	⊓	⊓	⊓	⊓
A	϶	√	∪	⊓	϶	϶	϶	϶	϶	⊓	⊓	⊓	⊓	⊓	⊓	⊓
B	϶	³√	ʃ	϶	϶	϶	϶	϶	϶	⊓	⊓	⊓	⊓	⊓	⊓	⊓

Fonte. <http://www.unicode.org/>

# Unicode (VI)

- *Emoticons*
  - Código U+1F600–U+1F64F

	1F60	1F61	1F62	1F63	1F64
0	😊	😐	☹	😱	😢
1	😁	😑	☹	😱	😢
2	😂	😑	☹	😳	😊
3	😊	😑	☹	😴	😊
4	😊	😑	☹	😴	😴
5	😊	😑	☹	😴	😴
6	😊	😑	☹	😊	😊
7	😊	😑	☹	😊	😊

Fonte. <http://www.unicode.org/>

## Unicode (VII)

- Unicode para braile
    - Código U+2800–U+28FF

Fonte. [http://commons.wikimedia.org/wiki/File:Unicode\\_Braille\\_table.svg](http://commons.wikimedia.org/wiki/File:Unicode_Braille_table.svg)

## Códigos de caracteres

- A ordem dos caracteres na tabela ASCII é importante
    - Permite ordenar com facilidade
  - Classes de códigos:
    - Caracteres imprimíveis
      - Aparecem visíveis no ecrã e na impressora
    - Caracteres de controlo
      - Controlo da posição no ecrã ou na impressora
        - VT: *Vertical Tab*
        - LF: *Line Feed*
        - FF: *Form Feed*
      - Causam uma acção
        - BEL: campainha
        - DEL: apaga o carácter anterior
      - Comunicação de estado entre o computador e o dispositivo de I/O
        - ESC: altera o significado dos caracteres seguintes

# Teclado

- Scan code
  - 2 *scan code* no teclado
    - Um é gerado quando a tecla é pressionada e o outro quando se solta a tecla
  - Convertido para Unicode, ASCII, EBBCID por software incluído no PC
- Vantagem
  - Fácil de adaptar a diferentes linguagens e diferentes *layout* de teclados;
  - Os *scan codes* separados permitem múltiplas combinações de teclas
    - Exemplo: teclas de *shift* e *control*

## Outras formas de leitura de alfa-numéricos

- OCR – *Optical Character Reader/Recognition*
  - Leitura (*scan*) do texto e introduz o texto (caracteres)
- Leitores de códigos de barras
  - Usados em aplicações que requerem a introdução rápida, sem erros ou repetitiva de dados, com um treino mínimo dos funcionários
  - Exemplos: supermercados, controlo de inventário
- Leitores de banda magnética
  - Leitura de dados alfanuméricos de cartões (de crédito/débito)
- RFID – *Radio Frequency Identification*
  - Leitura e transmissão de dados entre etiquetas RFID e computadores
- Voz
  - É fácil gravar a voz num formato digital mas difícil de converter para alfanumérico
  - Requer conhecimentos de padrões de som em linguagem, regras de pronúnciação, gramática e sintaxe
  - Grau de dificuldade variável em função da língua ... e do sotaque!

# Imagens

- Fotografias, figuras, ícones, desenhos, gráficos
- Duas opções:
  - *Bitmap* ou *raster images*
  - *Object* ou *Vector images*
    - Composto de objectos gráficos como linhas e curvas definidas geometricamente
- Diferenças:
  - Qualidade da imagem;
  - Espaço de armazenamento necessário;
  - Tempo de transmissão;
  - Facilidade de modificação

# Imagens bitmap

- Usado para imagens realísticas, com uma variação contínua de sombra, cor, formato e textura
  - Exemplos:
    - Fotografias digitalizadas
    - Imagens geradas pelo *paint* do *Windows*
  - Formatos populares: JPEG, BMP
- Preferível quando a imagem contém grande quantidade de detalhe e os requisitos de processamento são simples
- *Input*.
  - *Scanners*;
  - Máquinas fotográficas digitais;
  - Criado no PC
- Gestão feita por software de edição de fotografias e similares

# Imagens bitmap

- Cada pixel individual é um elemento gráfico guardado com um número em binário;
  - Pixel: pequeno ponto associado a uma dada coordenada
- Monocromática
  - 1 bit por pixel
- Escala de cinzento: preto, branco ou 254 níveis de cinzento
  - 1 byte por pixel
- Cores: 16 cores, 256 cores, 24-bit *true color* (16.7 milhões de cores)

# Imagenes bitmap



1 bit p&b



8 bits  
cinzentoo

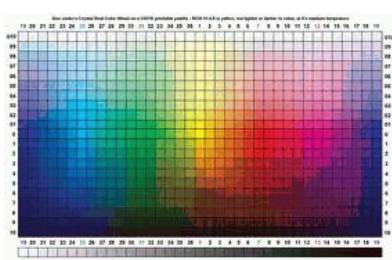


07.29.2010 8 bits

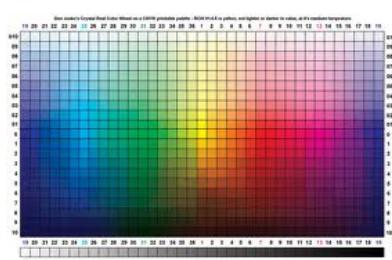


07.29.2010 24 bits

# Imagenes bitmap



8 bits



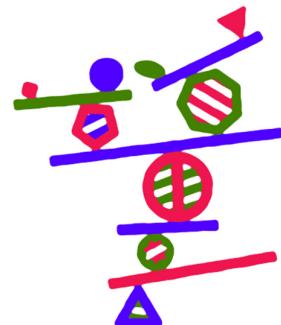
24 bits

# Imagens bitmap

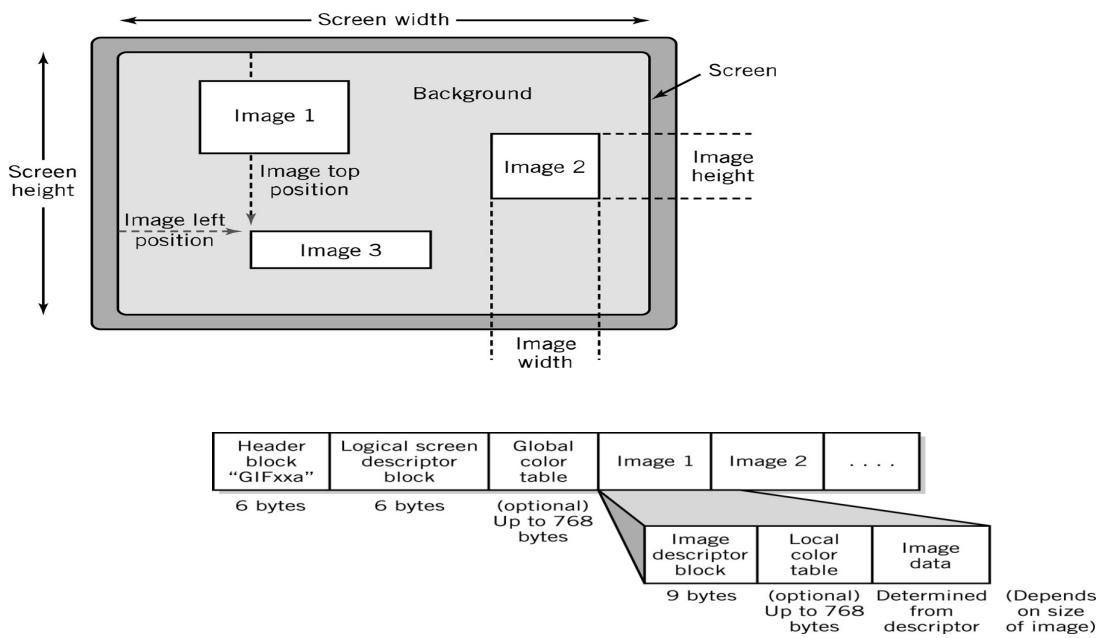
- Ficheiros de grande dimensão
  - Exemplo: 600 linhas por 800 pixel com 1 byte por cada uma das três cores → ~1,5 Mb
    - $600 \times 800 = 480.000$  pixels  $\times 3$  bytes = 1.440.000
- Tamanho do ficheiro determinado por:
  - Resolução (número de pixels por polegada)
    - Nível de detalhe que afecta a clareza o *sharpness* de uma imagem
  - *Levels*: número de bits usados para definir um nível de cinzento ou número de cores
    - Paleta: tabela de cores onde se usa um código do valor da cor para cada pixel
  - Compressão dos dados

## GIF - PNG - JPEG

- GIF – Graphics Interchange Format
  - Desenvolvido pela CompuServe em 1987
  - GIF89a permite construir imagens animadas
    - Permite que as imagens sejam mostradas em sequência, com intervalo de tempo definido
  - Limite no número de cores: 256
  - Compressão da imagem usando o algoritmo LZW – Lempel-Zif-Welch
  - Usado em imagens que sejam desenhos de linhas, *clip art* e outras onde surgem grandes blocos da mesma cor sólida
  - Compressão sem perca de dados (*lossless compression*)



# GIF - PNG - JPEG



Fonte: Englander, Irv; "The architecture of Computer Hardware, system software & networking"

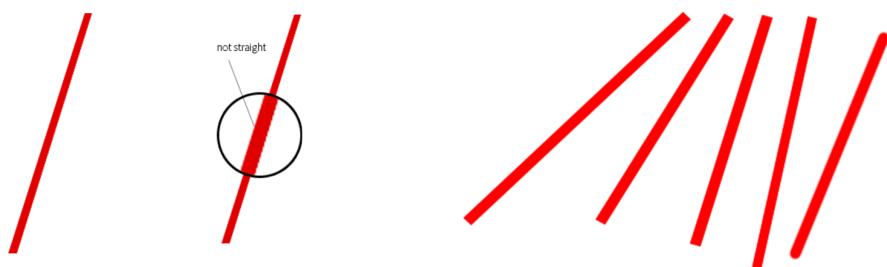
# GIF - PNG - JPEG

- **PNG - Portable Network Graphics**
  - PNG's Not GIF
  - Data: 1995
  - Surge em resposta aos problemas identificados com o formato GIF
- **JPEG – *Joint Photographers Expert Group***
  - Permite mais de 16 milhões de cores
  - Popular na área da fotografia
  - Usa um algoritmo de compressão que perde informação (*lossy compression*)
    - Perde alguns dados de forma a diminuir o tamanho do ficheiro e permitir a sua transmissão de forma mais célere
    - Pode levar à redução da qualidade das imagens, tendendo a distorcer as linhas rectas (*sharp lines*)

# GIF - PNG - JPEG

- Escolha determinada por vários fatores:
  - Velocidade
    - Menos problemático nos computadores atuais;
  - Nitidez (*sharpness*)
  - Cor sólida
  - Transparência
  - Tamanho
  - Paleta de cor
  - Tamanho dos ficheiros

# GIF - PNG - JPEG



Fonte: <http://graphicdesign.stackexchange.com/questions/20394/how-to-remove-jagged-edge-aliasing-on-a-straight-diagonal-line>



Fonte: <http://www.scantips.com/basics09b.html>

# GIF - PNG - JPEG

- JPEG – Questões
  - Ficheiros JPEG perdem qualidade quando são abertos e fechados?
  - Ficheiros JPEG perdem qualidade quando são abertos, editados e fechados?
  - Se comprimir uma imagem a 70% e posteriormente abrir e gravar com 90%, a qualidade da imagem fica reestabelecida nos 90%?
  - Gravar com a mesma qualidade numérica em dois programas distintos dá o mesmo resultado?
  - Escolher 100% significa que não vai ocorrer qualquer degradação?
  - JPEG progressivos são mais rápidos a descarregar do que os normais (não progressivos)?
  - JPEG necessita de maior poder computacional do que, por exemplo, o formato GIF?
  - JPEG suporta transparências?

## Imagen como objectos (*object images*)

- Imagens criadas por ferramentas de desenho (Exemplo: Illustrator) ou como resultado de aplicações específicas (ex.: gráficos criados numa folha de cálculo)
- Compostas por linhas e formas de várias cores
- O computador traduz formulas geométricas para criar gráficos
- O espaço ocupado pelo ficheiro depende da complexidade da imagem
  - Normalmente gera ficheiros mais pequenos do que os bitmaps
- Baseada em formulas matemáticas
  - Fácil de mover, de dimensionar e rodar sem perder qualidade/forma (ao contrário do que acontece com os bitmap)
- Não pode ser usado em fotografias ou pinturas
- Não pode ser mostrado ou impresso de forma directa
  - Tem de se converter para bitmap, dado que os dispositivos de *output* mostram imagens bitmap (com exceção das *plotters*)
- Há filmes de cinema criados com este tipo de imagens

# Bitmap vs Object images

Bitmap (raster)	Object (vector)
Mapa de pixels	Formas definidas geometricamente
Qualidade fotográfica	Desenhos complexos
Software de tratamento de imagens	Software de desenho vectorial
Maiores necessidades em termos de espaço de armazenamento	Maiores necessidades em termos de processamento
Aumentar as imagens causa perda de qualidade	Objectos aumentam de forma contínua
Resolução de <i>output</i> limitada pela resolução da imagem	Resolução de <i>output</i> limitada pela resolução do dispositivo de saída

# PostScript

- *Page Description Language*

```
%!PS-Adobe-3.0 %APL_DSC_Encoding: UTF8 %%Title: (Pages from
HEID_issue2_9.2.editPH) %%Creator: (Acrobat: pictwpstops filter
1.0) %%CreationDate: (Wednesday, October 05 2005 10:37:45 EDT) %%
Henry) %%Pages: 1 %%DocumentFonts: %%DocumentNeededFonts:
%%DocumentSuppliedFonts: %%DocumentData: Clean7Bit %%PageOrder:
Ascend %%Orientation: Portrait %%DocumentMedia: (Default) 612 792
() %%RBINupNess: 1 1 %%ADO_ImageableArea: 18 40 594 774 %%RBIPCFileN
(HP4100_5.PPD)
%%RBIPPDFFileVersion: (1.1.1 X)
%%RBIDocumentSuppliedFonts: %%RBINumCopies: 1 %%LanguageLevel: 3
%%DocumentProcessColors: Cyan Magenta Yellow Black
%%EndComments %%BeginDefaults %%ViewingOrientation: 1 0 0
1 %%EndDefaults userdict/dscInfo 5 dict dup begin /Title(Pages fr
HEID_issue2_9.2.editPH)def /Creator(Acrobat: pictwpstops filter
1.0)def /CreationDate(Wednesday, October 05 2005 10:37:45 EDT)def
Henry)def /Pages 1 def end put %%BeginProlog %%BeginFeature: *Job
1 /limit interpolate { dup type /dicttype eq { dup /Interpolate
/Interpolate get { 1 1 idtransform 2 index /ImageMatrix get dtra
1 exch div exch abs 1 exch div 2 copy lt { exch }if pop 4 lt
/Interpolate false put } if } if } if } bind def /image { li
systemdict /image get exec } bind def /imagemask { limitinterpol
/imagemask get exec } bind def %%EndFeature %%BeginFeature:
```

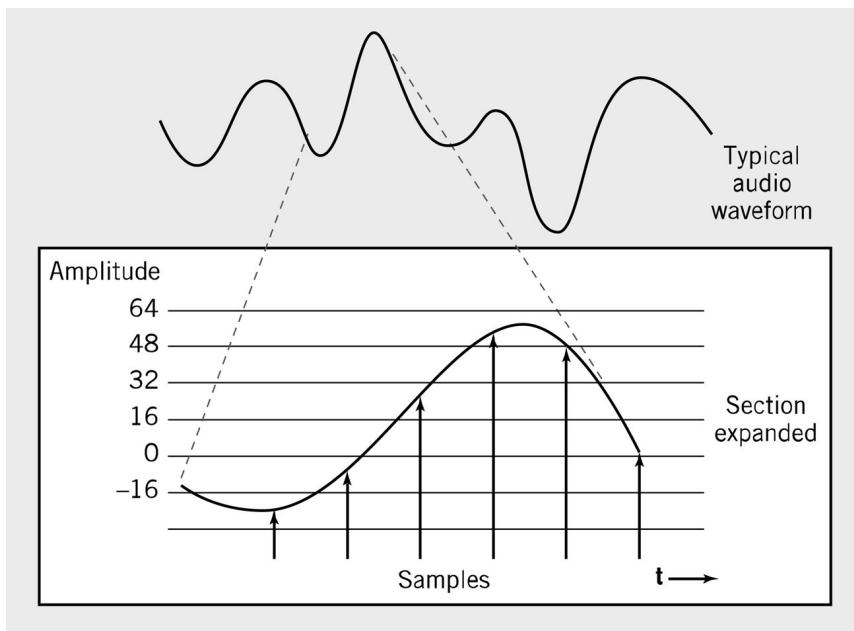
# Vídeo

- Quantidades massivas de dados
  - Exemplo 1
    - Câmara de vídeo produz um vídeo com 640 x 480 pixels, em *true color*, gravando 30 *frames/segundo*  $> 27,65\text{Mb}/\text{Segundo}$
    - Filme com 1 minuto: 1,6Gb
  - Exemplo 2
    - $1024 \times 768 \text{ pixel} \times 3 \text{ bytes/cor} \times 30 \text{ frames/segundo} = 70,8 \text{ Mb/segundo}$
    - Filme de 1 minuto: 4,25 Gb
- Opções para reduzir o tamanho do ficheiro
  - Diminuir a resolução
  - Limitar o número de cores
  - Reduzir o *frame rate*
  - Comprimir o vídeo
- A opção depende do destino do vídeo
  - *Streaming* de vídeo: é enviado pela Internet e visualizado ao mesmo tempo que é recebido
  - Dados locais: como um DVD ou um ficheiro armazenado no computador

# Áudio

- Requisitos para transmissão e processamento são inferiores aos que existiam no vídeo
- O som é analógico por natureza
  - Para o digitalizar recolhe-se amostras em intervalos regulares de tempo
  - Usa-se um conversor A-D
- A velocidade de recolha de amostras deve ser suficientemente elevada para captar todas as nuances do som
  - Número de vezes, por segundo, que o som é medido durante o processo de gravação
    - 1000 amostras por segundo = 1Khz
    - CD de música: 44,1 khz
    - Cada amostra pode ser gravada com 8 bits, 16 bits ou 2x16 bits no caso do som stereo

# Áudio



Fonte: Englander, Irv; "The architecture of Computer Hardware, system software & networking"

## Exemplos de formatos de áudio

- **MP3 - MPEG 1 Layer-3**
  - Deriva do **MPEG-2** (ISO Moving Picture Experts Group)
    - 1993: **MPEG-1 Audio** (MPEG-1 Part 3)
    - 1995: **MPEG-2 Audio** (MPEG-2 Part 3)
  - Remove os sons que o ouvinte não consegue ouvir ou não vai notar que foram removidos
  - Usa várias técnicas para criar ficheiros “pequenos”
  - Comparado com um CD áudio: 75% a 90% de redução de tamanho
    - Importante para a transmissão e armazenamento
  - Bit rate: velocidade de amostragem x número de bits por amostra
    - Valor típico: 128kbps
  - Combinação de ficheiros com tamanho reduzido e com aceitável qualidade
    - => Popular no final da década de 90!
- **WAV**
  - Desenvolvido pela Microsoft como parte da sua especificação multimédia
  - Formato genérico, usado para armazenar e reproduzir pequenos sons

# Compressão de dados (I)

- Compressão
  - Guardar os dados de forma a ocupar menos espaço
- Rácio de compressão
- *Lossless* – sem perdas
  - O algoritmo inverso permite obter os dados originais de forma exacta.
    - Exemplo: TIFF, GIF, PCX
- *Lossy* – Com perdas
  - Balanço entre os dados que se perde e o tamanho do ficheiro
  - Rácios de compressão mais elevados: chegando a 10:1
  - Usado habitualmente em multimédia
    - Exemplo: JPEG

# Compressão de dados (II)

05573200001473291000006682732732...

0155732041473291056682732732...

732 -> Z

0155Z0414Z91056682ZZ...