



Mestrado Integrado Eng^a. Informática

1º ano

2019/20

A.J.Proença

Tema

Introdução aos Sistemas de Computação



Estrutura do tema ISC

1. Representação de informação num computador
2. Organização e estrutura interna dum computador
3. Execução de programas num computador
4. Análise das instruções de um processador
5. Evolução da tecnologia e da eficiência



Um computador é um sistema físico que:

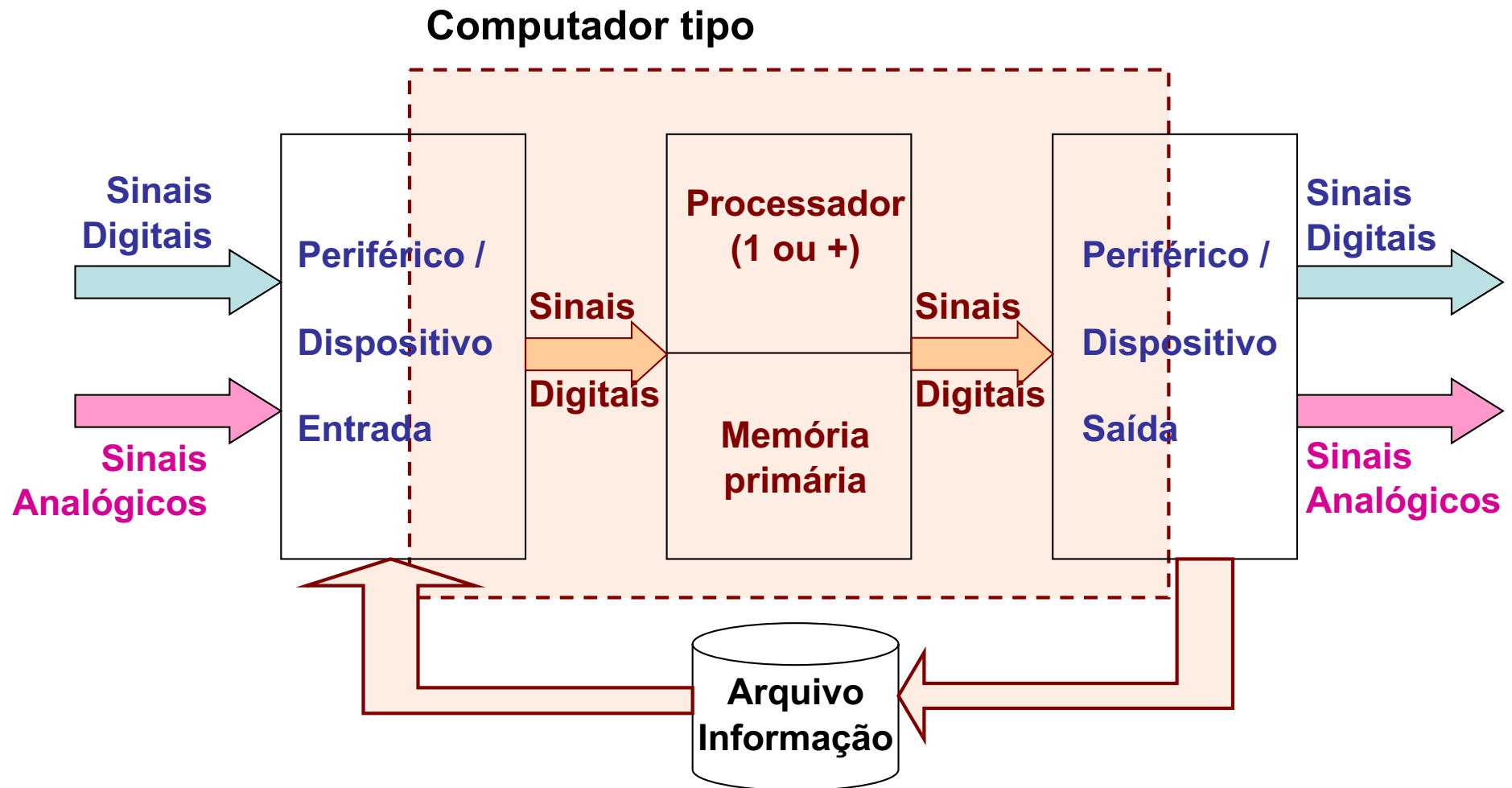
- recebe informação,
processa / arquiva informação,
transmite informação, e ...
- é programável
i.e., a funcionalidade do sistema pode ser modificada,
sem alterar fisicamente o sistema

Quando a funcionalidade é fixada no fabrico do sistema onde o computador se integra, diz-se que o computador existente nesse sistema está “embebido”: ex. *smart phone*, máq. fotográfica, automóvel, ...

Como se representa a informação num computador ?

Como se processa a informação num computador ?

Noção de computador (2)





- **Como se representa a informação num computador ?**
 - representação da informação num computador ->
- **Como se processa a informação num computador ?**
 - organização e funcionamento de um computador ->



Como se representa a informação?

– com **binary digits**!



Um **algarismo** ou **dígito**, é um tipo de representação (um símbolo numérico, como "2" ou "5") usado em combinações (como "25") para representar **números** (como o número 25) em **sistemas de numeração posicionais**. O nome "dígito" vem do facto de os 9 dígitos (do **latim** *digitem*, "dedo") das mãos corresponderem aos 10 símbolos do sistema de numeração comum de **base 10**, isto é, o decimal (digestivo do latim antigo *decoração* . que significa nove) dígitos.

A palavra "algarismo" tem sua origem no nome do famoso matemático **Al-Khwarizmi**.

Mais:

- Cada um dos elementos de um numeral é um algarismo ou dígito:
 - Numeral com 3 dígitos: 426.
 - Numeral com 10 algarismos: 1.234.567.890

→ • Dígitos **Binários**: podem ser apenas dois, o 0 (zero) e o 1 (um)



Como se representa a informação?

- com *binary digits*!

Tipos de informação a representar:

- números (para cálculo)
 - » bases de numeração, inteiros (positivos e negativos)
 - » reais (*fp*), norma IEEE 754
- textos (caracteres alfanuméricos)
- conteúdos multimédia
- código para execução no computador

Sistemas de numeração :
quanto vale na base 10 um n° representado numa outra base



1532.54₁₀ (base 10) ; quanto vale cada algarismo?

$$1*10^3 + 5*10^2 + 3*10^1 + 2*10^0 + 5*10^{-1} + 4*10^{-2} = 1532.54_{10}$$

Nota: a potência de 10 dá-nos a ordem do algarismo no número...

1532₆ (base 6) ; quanto vale cada algarismo na base 10?

$$1*6^3 + 5*6^2 + 3*6^1 + 2*6^0 = 416_{10}$$

1532₁₃ (base 13) ; quanto vale cada algarismo na base 10?

$$1*13^3 + 5*13^2 + 3*13^1 + 2*13^0 = 3083_{10}$$

110110.011₂ (base 2) ; quanto vale cada algarismo na base 10?

$$1*2^5 + 1*2^4 + 0*2^3 + 1*2^2 + 1*2^1 + 0*2^0 + 0*2^{-1} + 1*2^{-2} + 1*2^{-3} = 54.375_{10}$$

Sistemas de numeração :
como se passa um n° na base 10 para uma outra base



1532.54₁₀ (base 10) ; algoritmo para extrair os algarismos?

- parte inteira: divisão sucessiva pela base e...
- parte decimal: multiplicação sucessiva pela base e...

416₁₀ ; quanto vale cada algarismo na base 6?

- parte inteira ... parte decimal ...

3083₁₀ ; quanto vale cada algarismo na base 13?

- parte inteira ... parte decimal ...

154.375₁₀; quanto vale cada algarismo na base 2?

- parte inteira ... parte decimal ...

Sistemas de numeração : caso particular da base 2



110110.011₂ (base 2) ; quanto vale cada algarismo na base 10?

$$1*2^5 + 1*2^4 + 0*2^3 + 1*2^2 + 1*2^1 + 0*2^0 + 0*2^{-1} + 1*2^{-2} + 1*2^{-3} = \dots$$

Para simplificar:

- eliminar os produtos, ignorar parcelas com produtos por 0

$$\bullet \text{ } 1*2^5 + 1*2^4 + 0*2^3 + 1*2^2 + 1*2^1 + 0*2^0 + 0*2^{-1} + 1*2^{-2} + 1*2^{-3} = \dots$$

$$\Rightarrow 2^5 + 2^4 + 2^2 + 2^1 + 1/2^2 + 1/2^3 = \dots$$

Recomendação:

- decorar a tabuada das potências de 2 (**2⁰ + 2¹⁰**)
- compreender as potências de 2 múltiplas de 10

Numeração de base 2 : dicas para uma rápida conversão de potências de 2 para a base 10



$2^0 =$	1
$2^1 =$	2
$2^2 =$	4
$2^3 =$	8
$2^4 =$	16
$2^5 =$	32
$2^6 =$	64
$2^7 =$	128
$2^8 =$	256
$2^9 =$	512
$2^{10} =$	1024

$$2^{10} = 1024 = 1 \text{ Ki}(\text{bi}) \approx 1\,000 = 10^3 = 1 \text{ K}(\text{ilo})$$

...

$$2^{12} = 2^2 * 2^{10} = 4 \text{ Ki}(\text{bi}) \approx 4\,000 = 4 * 10^3 = 4 \text{ K}$$

...

$$2^{16} = 2^6 * 2^{10} = 64 \text{ Ki}(\text{bi}) \approx 64 * 10^3 = 64 \text{ K}$$

$$2^{20} = 1 \text{ Me}(\text{bi}) \approx 1\,000\,000 = 10^6 = 1 \text{ M}(\text{ega})$$

$$2^{30} = 1 \text{ Gi}(\text{bi}) \approx 1\,000\,000\,000 = 10^9 = 1 \text{ G}(\text{iga})$$

$$2^{40} = 1 \text{ Te}(\text{bi}) \approx 10^{12} = 1 \text{ T}(\text{era})$$

$$2^{50} = 1 \text{ Pe}(\text{bi}) \approx 10^{15} = 1 \text{ P}(\text{eta})$$

Sistemas de numeração : caso particular da base 16 (hexadecimal)



- Dígitos na base 16: $0, 1, 2, \dots, 9, \overset{10}{a}, \overset{11}{b}, \overset{12}{c}, \overset{13}{d}, \overset{14}{e}, \overset{15}{f}$
- Vantagens sobre um valor de 32 bits:
 $10100110100001110110010111010100_2$ VS. $a68765d4_{16}$
- Facilidade de conversão:

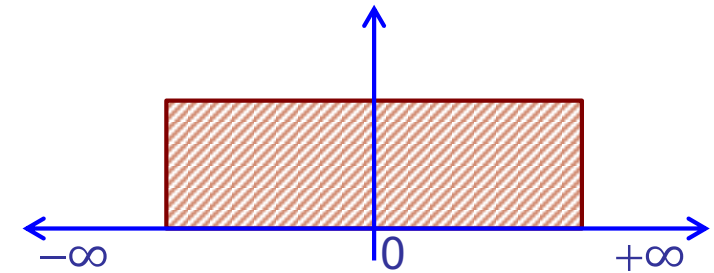
1010	0110	1000	0111	0110	0101	1101	0100 ₂	←
a	6	8	7	6	5	d	4 ₁₆	
- Mesmo com ponto decimal:

							←	→	
1010	0110	1000	0111	0110	0101	1101	.01 ₂		
1010	0110	1000	0111	0110	0101	1101	.0100 ₂		
a	6	8	7	6	5	d	.4 ₁₆	← →	



Gama de valores representáveis

- ideal: todos os valores e simetria em relação ao 0
- mas ...
- e quantos bits para representar um inteiro?



Representação de positivos & negativos

- estratégias
- análise dum exemplo com todos os valores possíveis
 - S+M: Sinal + Magnitude/amplitude≠
 - Complemento para 1
 - Complemento para 2
 - Notação por excesso

Inteiros positivos e negativos: o universo com 3 bits



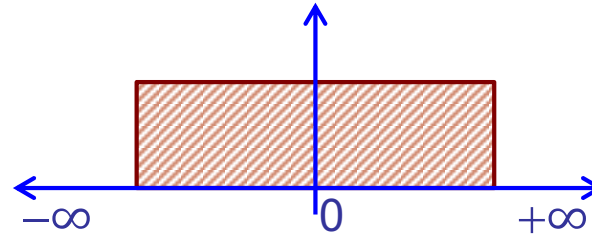
Base 10	Base 2	S+M	Comp p/ 1	Comp p/ 2	Excesso 2^{n-1}	Excesso $2^{n-1}-1$
0	000	+0	+0	+0	0-4 > -4	0-3 > -3
1	001	+1	+1	+1	1-4 > -3	1-3 > -2
2	010	+2	+2	+2	2-4 > -2	-3 > -1
3	011	+3	+3	+3	3-4 > -1	3-3 > 0
4	100	-0	$-11_2 > -3$	$-(11+1)_2 > -4$	4-4 > 0	4-3 > +1
5	101	-1	$-10_2 > -2$	$-(10+1)_2 > -3$	5-4 > +1	5-3 > +2
6	110	-2	$-01_2 > -1$	$-(01+1)_2 > -2$	6-4 > +2	6-3 > +3
7	111	-3	$-00_2 > -0$	$-(00+1)_2 > -1$	7-4 > +3	7-3 > +4

Nota: $n = \text{\#bits}$, $2^{n-1} = 2^{3-1} = 2^2 = 4$, $2^{n-1}-1 = 2^{3-1}-1 = 2^2-1 = 3$

Representação de reais em vírgula flutuante



- **Gama de valores**
 - esta gama é viável?



- **Notação científica**

$$\text{Valor} = (-1)^s * \text{Mantissa} * \text{Radix}^{\text{Exp}}$$

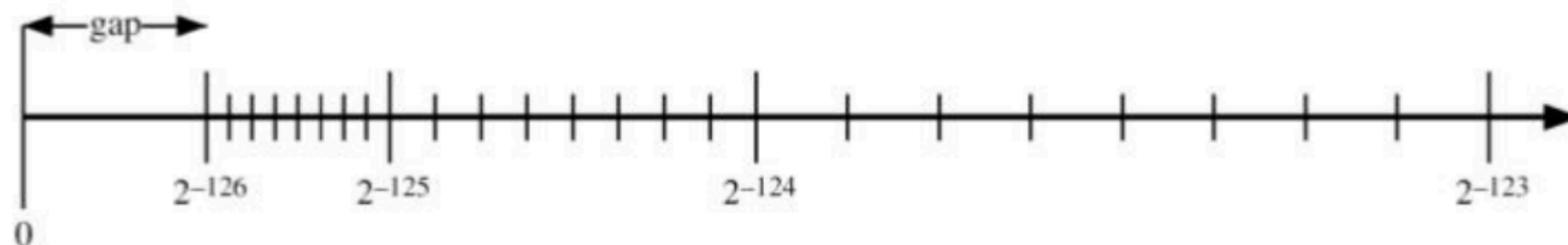
- **Normalização na representação**
 - valores normalizados e subnormais
- **Intervalo e precisão de valores representáveis**
- **Formato binário dum valor em fp**
 - Sinal, Mantissa ou parte Fracionária, Expoente
- **O bit escondido**
- **A norma IEEE 754-2008 para valores em fp**

A norma IEEE 754-2008 para valores em fp

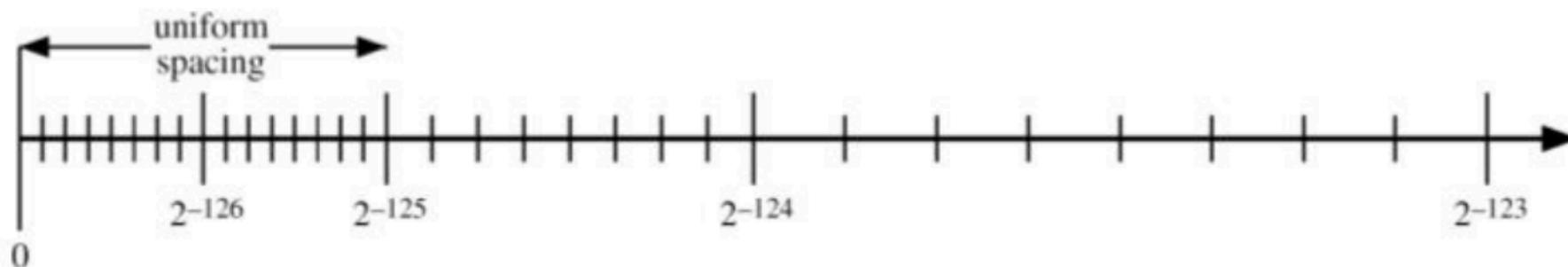


- **Representação do sinal e parte fracionária**
 - S+M
- **Representação do expoente**
 - Notação por excesso $2^{n-1}-1$
- **Valor decimal de um fp em binário (normalizado)**
 - Precisão simples: $V = (-1)^S * (1.F) * 2^{E-127}$
- **Valor decimal de um fp em binário (subnormal)**
 - Precisão simples: $V = (-1)^S * (0.F) * 2^{-126}$
- **Representação do zero:** $E=0$ e $F=0$
- **Representação de $\pm\infty$:** $E = 1111\ 1111_2$ e $F = 0$
- **Representação de n.º não real:** $E = 1111\ 1111_2$ e $F \neq 0$

O papel dos subnormais na norma IEEE 754



(a) 32-bit format without denormalized numbers



(b) 32-bit format with denormalized numbers



Como se representa a informação?

- com *binary digits*!

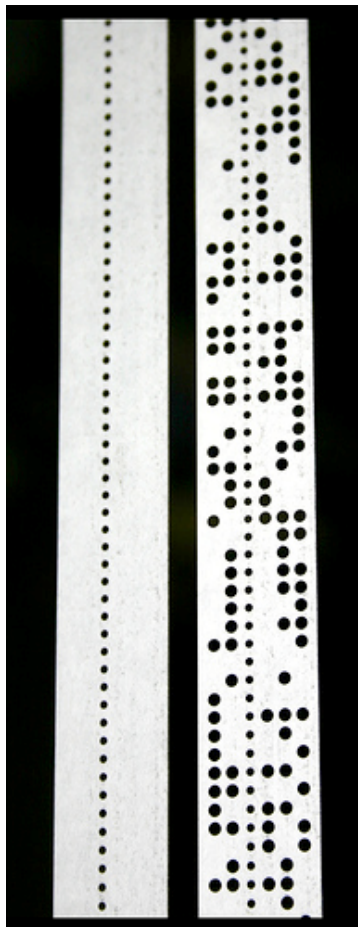
Tipos de informação a representar:

- números (para cálculo)
 - » bases de numeração, inteiros (positivos e negativos)
 - » reais (*fp*), norma IEEE 754
- textos (caracteres alfanuméricos)
 - » Baudot, Braille, ASCII, Unicode, ...
- conteúdos multimédia
- código para execução no computador

Ex.: codificação telegráfica de texto, código de Baudot, 5-bits



- Baudot,



V	IV		I	II	III	V	IV		I	II	III
		A /	•			•	•	P. %	•	•	•
	•	B 8			•	•	•	Q /	•		•
	•	C 9	•		•	•	•	R -			•
	•	D 0	•	•	•	•		S ;			•
		E 2				•		T !	•		•
		E' &	•	•				U 4	•		•
	•	F 5			•	•		V '	•	•	•
	•	G 7			•	•		W ?		•	•
	•	H 4	•	•		•		X ,		•	
		I 3			•			Y 3			•
	•	J 6	•			•		Z :	•	•	
•	•	K (•			•		£ .	•		
•	•	L =	•	•		•	•	⌘ ⌘ Erasure			
•	•	M)			•	•		Figure Blank			
•	•	N N°			•	•		Letter Blank			
		O 5	•	•	•						

Letters	Figures	V	IV	I	II	III	Letters	Figures	V	IV	I	II	III
A	1			•			-	.	•		•		
E	2				•		X	9/	•			•	
Y	3					•	S	7/	•				•
/	1/			•	•		Z	:	•		•	•	
1	3/				•	•	W	?	•			•	•
U	4			•		•	T	2	•		•	•	•
O	5			•	•	•	V	!	•		•	•	•
							Letter Blank						
J	6	•	•				K	(•	•	•		
G	7	•		•			M)	•	•	•	•	
B	8												
H	1												
F	5/												
C	9												
D	0												
Figure Blank													

Fig 1. The Baudot code

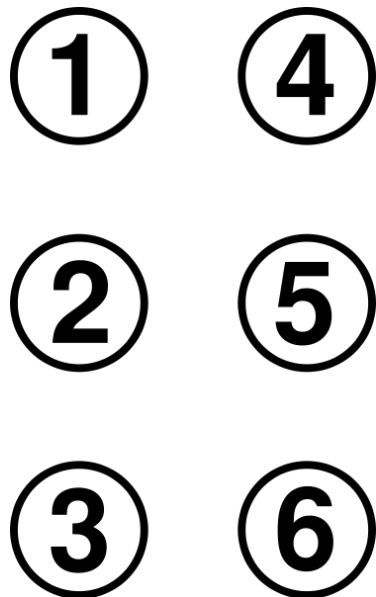


*Ex.: codificação de texto em relevo,
código Braille com 6-bits*



- Baudot, Braille,

Alfabeto Braille



a	b	c	d	e	f	g	h	i	j
k	l	m	n	o	p	q	r	s	t
u	v	x	y	z	ç	é	á	è	ú
â	ê	ì	ô	ù	à	ï	ü	õ	w
í	ó	ã	sinal numérico	-	'	—	...	grifo maiúscula	caixa alta
,	;	:	.	\$?	!	()	"	*
1	2	3	4	5	6	7	8	9	0

*Ex.: representação de texto
com ASCII (7 bits)*



Tabela ASCII 7 bits

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

H	e	l	l	o		w	o	r	l	d	!
48	65	6c	6c	6f	20	77	6f	72	6c	64	21

*Ex.: codificação universal de texto,
UTF-8 no Unicode*



- Baudot, Braille, ASCII, Unicode, (UTF-8)

binary	hex	decimal	notes
00000000-01111111	00-7F	0-127	US-ASCII (single byte)
10000000-10111111	80-BF	128-191	Second, third, or fourth byte of a multi-byte sequence
11000000-11000001	C0-C1	192-193	Overlong encoding: start of a 2-byte sequence, but code point ≤ 127
11000010-11011111	C2-DF	194-223	Start of 2-byte sequence
11100000-11101111	E0-EF	224-239	Start of 3-byte sequence
11110000-11110100	F0-F4	240-244	Start of 4-byte sequence
11110101-11110111	F5-F7	245-247	Restricted by RFC 3629: start of 4-byte sequence for codepoint above 10FFFF
11111000-11111011	F8-FB	248-251	Restricted by RFC 3629: start of 5-byte sequence
11111100-11111101	FC-FD	252-253	Restricted by RFC 3629: start of 6-byte sequence
11111110-11111111	FE-FF	254-255	Invalid: not defined by original UTF-8 specification



Como se representa a informação?

- com ***binary digits***!

Tipos de informação a representar:

- números (para cálculo)
 - » inteiros: S+M, Compl. p/ 1, Compl. p/ 2, Excesso
 - » reais (*fp*): norma IEEE 754
- textos (caracteres alfanuméricos)
 - » Baudot, Braille, ASCII, Unicode, ...
- conteúdos multimédia
 - » imagens fixas: BMP, JPEG, GIF, PNG, ...
 - » audio-visuais: AVI, MPEG/MP3, ...
- código para execução no computador

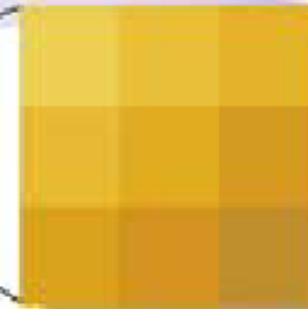
Ex.: representação de uma imagem em bitmap



You can create a 24-bit image in a graphics program such as Paint.



A graphics program saves the image line by line, from the bottom to the top.



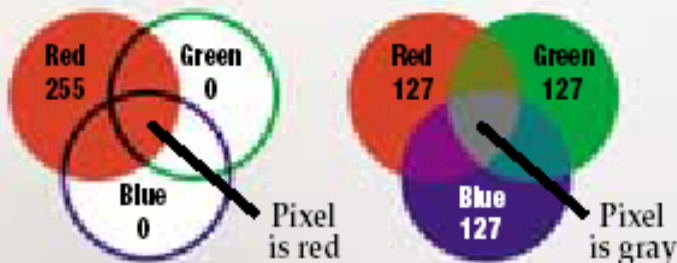
Each of the pixel's three-color values, RGB (red-green-blue), are read from left to right.

R 250 G 210 B 94	R 244 G 195 B 69	R 238 G 182 B 51
R 242 G 190 B 60	R 235 G 176 B 42	R 222 G 160 B 26
R 228 G 167 B 27	R 218 G 153 B 17	R 201 G 148 B 53

A graphics program translates the RGB values into palette values. The palette values are a software-specific decision; each program's values are different.

Forming A Pixel

A pixel is the smallest part of an image that a computer's monitor can control. Each pixel consists of three colors: red, green, and blue. Each of the three colors is assigned a value that shows its intensity; the values are from 0 to 255. You can think of each value as a percentage. For example, 127 has a 50% intensity. These are known as the RGB values.



Each palette value, a hexadecimal value in this case, is stored in the same order as displayed in the image.

FAD25E	F4C345	EEB633
F2BE3C	EBB02A	DEA01A
E4A71B	DA9911	C99435

The pixel values are stored in the bit-mapped file in the same width and depth as the original image.





Como se representa a informação?

- com *binary digits*!

Tipos de informação a representar:

- números (para cálculo)
 - » inteiros: S+M, Compl. p/ 1, Compl. p/ 2, Excesso
 - » reais (*fp*): norma IEEE 754
- textos (caracteres alfanuméricos)
 - » Baudot, Braille, ASCII, Unicode, ...
- conteúdos multimédia
 - » imagens fixas: BMP, JPEG, GIF, PNG, ...
 - » audio-visuais: AVI, MPEG/MP3, ...
- código para execução no computador
 - » noção de *instruction set*

Ex.: representação de código para execução num PC



```
int x = x+y;
```

```
addl 8(%ebp), %eax
```

Idêntico à expressão
x = x + y

```
0x401046:    03 45 08
```

- Código numa linguagem de programação
 - somar 2 inteiros
- Código numa linguagem mais próxima do processador
 - somar 2 inteiros (de 4-bytes)
 - operandos:
 - x: no registo `eax`
 - y: na memória em `[(ebp)+8]`
- Código “objecto” (em hexadecimal)
 - instrução com 3-bytes
 - na memória em `0x401046`