

# Arquitectura de Sistemas e Computadores I

## Semana 1

Miguel Barão

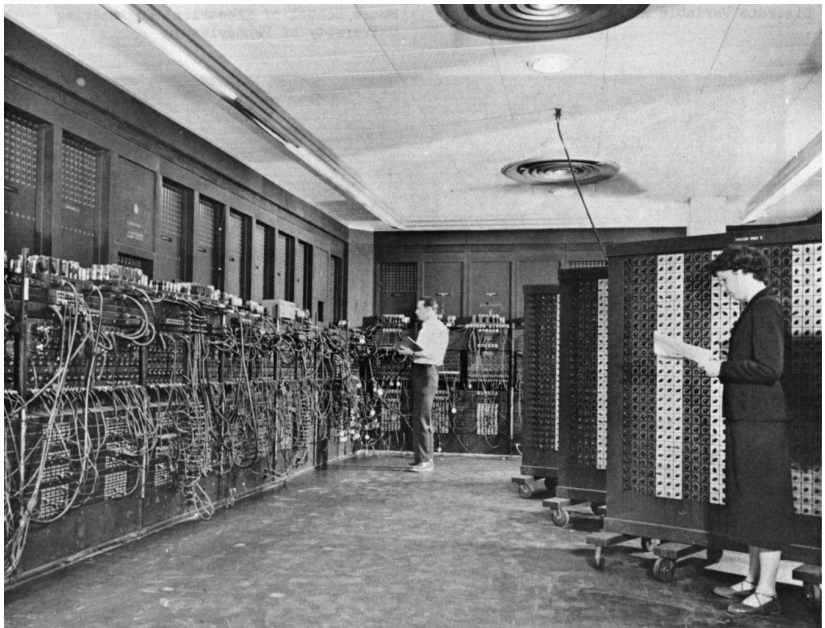
mjsb@di.uevora.pt

## Resumo

- Introdução à arquitectura de Von Neumann (stored program computer)
- Organização da memória RAM. Conceito de endereço e byte.

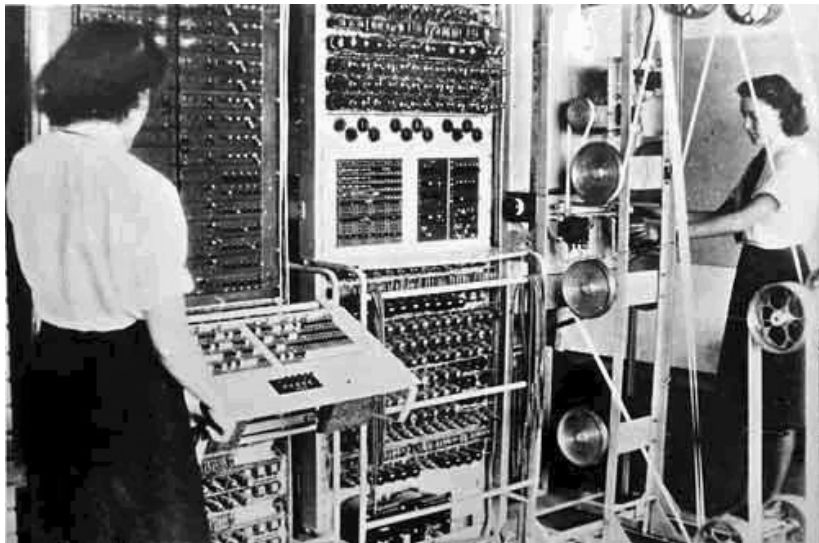
# ENIAC (Electronic Numerical Integrator And Computer)

Usado para cálculos de balística, 1946–1955.



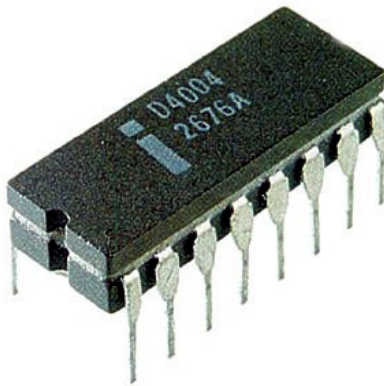
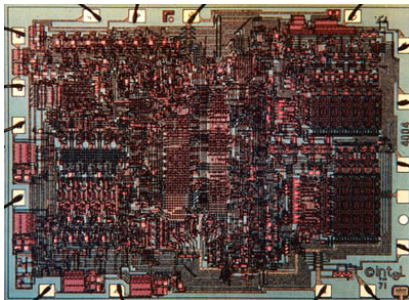
## Colossus Mark 2

Usado pelos Britânicos para ler mensagens cifradas pelos Alemães durante a 2ª Grande Guerra



## Intel 4004 (92000 instr./s)

CPU de 4 bits desenvolvido em 1971. Foi o primeiro microprocessador



# MOS 6502

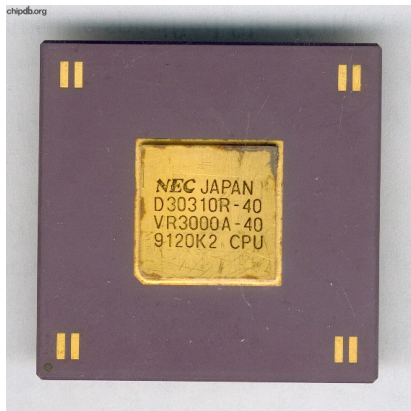
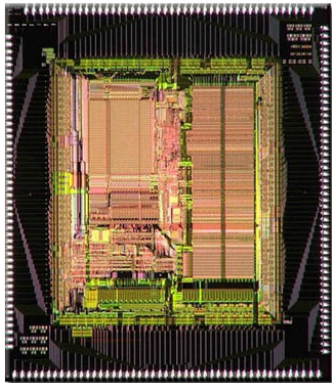
CPU de 8 bits desenvolvido em 1975. Custava \$25 na altura.



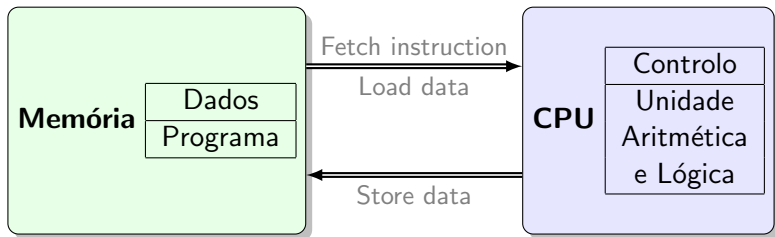
Equipava os microcomputadores: BBC, Apple IIe e Commodore 64

# MIPS R3000

CPU de 32bits desenvolvido em 1988. Pioneiro na execução em pipeline.



Processadores desta arquitectura são actualmente usados em dispositivos de rede, consolas de jogos, impressoras, set-top boxes, televisões digitais e modems de cabo/ADSL.



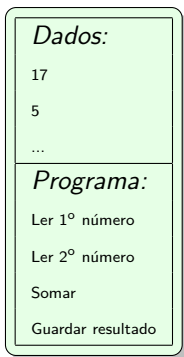
O programa e os dados são guardados conjuntamente em memória.

Um programa é um conjunto de números em memória que codificam as instruções a executar → código máquina.

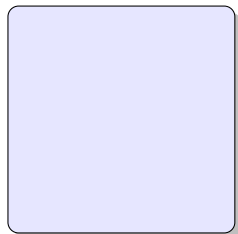


## Exemplo de execução de um programa

Pretende-se somar os números 17 e 5...

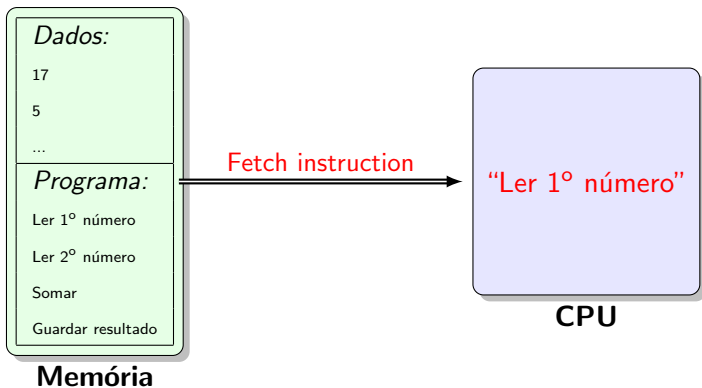


**Memória**



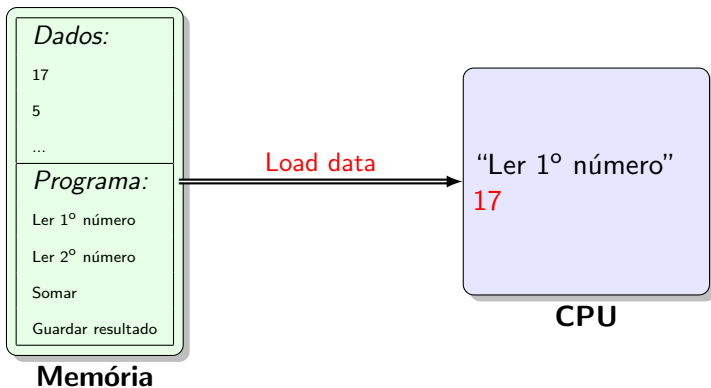
**CPU**

Pretende-se somar os números 17 e 5...

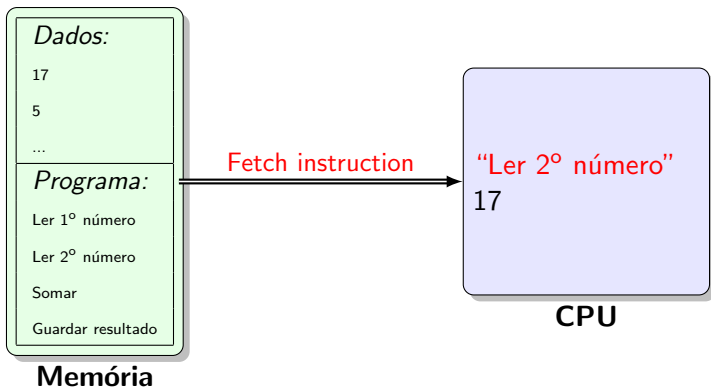


## Exemplo de execução de um programa

Pretende-se somar os números 17 e 5...

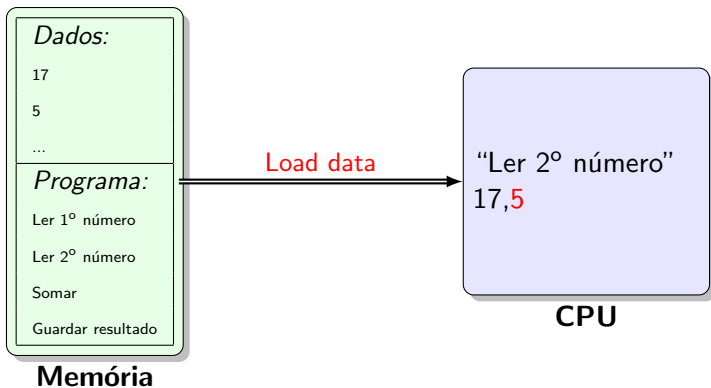


Pretende-se somar os números 17 e 5...



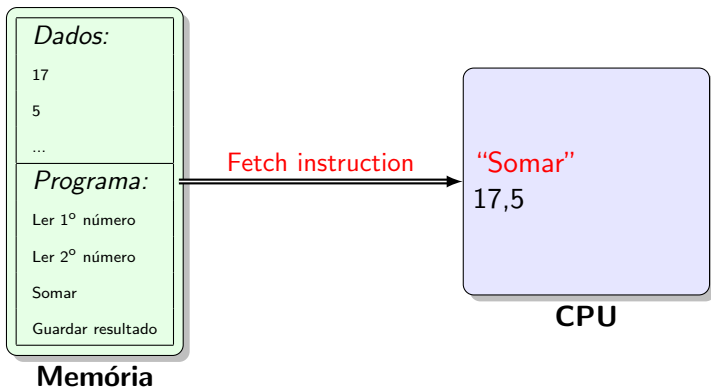
## Exemplo de execução de um programa

Pretende-se somar os números 17 e 5...

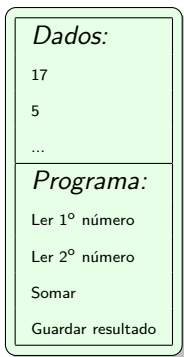


## Exemplo de execução de um programa

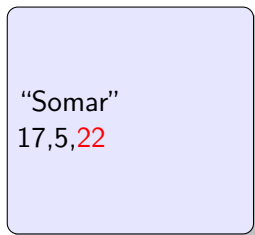
Pretende-se somar os números 17 e 5...



Pretende-se somar os números 17 e 5...



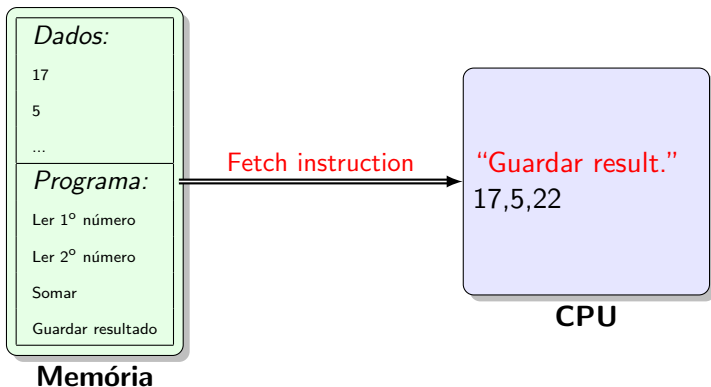
**Memória**



**CPU**

## Exemplo de execução de um programa

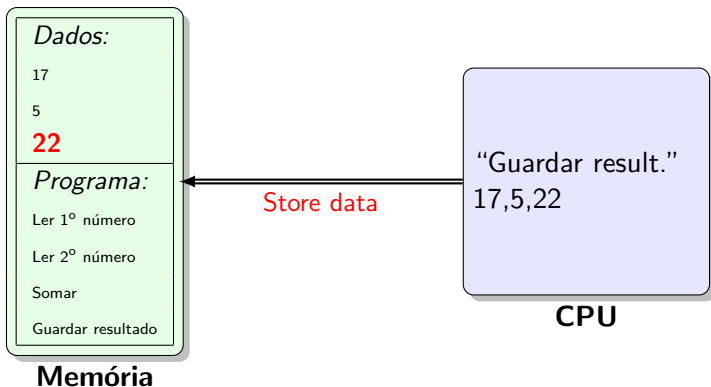
Pretende-se somar os números 17 e 5...



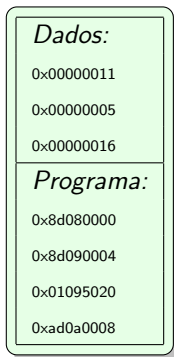


## Exemplo de execução de um programa

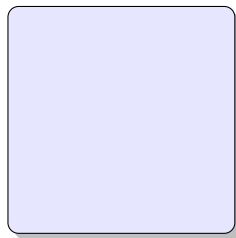
Pretende-se somar os números 17 e 5...



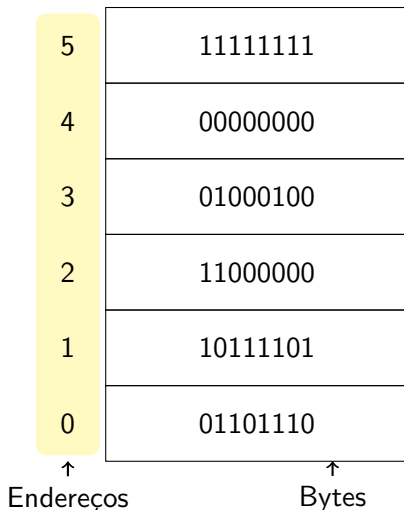
Na realidade o que está em memória é:



**Memória**



**CPU**



Funciona como uma estante onde as prateleiras são numeradas e cada prateleira armazena exactamente 8 bits de informação (1 byte).

- 1 bit é abreviatura de “binary unit”. Pode tomar os valores 0 ou 1. Um sistema digital funciona nesta base (base binária).
- 1 byte tipicamente são 8 bits. É a menor unidade de informação endereçável em memória.
- 1 word é a unidade de informação natural para uma dada arquitectura. No caso da arquitectura MIPS estudada nesta disciplina 1 word = 32 bits, mas pode ter outros tamanhos noutras arquitecturas (16 bits, 64 bits, etc).

## Atenção

Não é possível endereçar de forma directa um bit em memória.  
Um endereço apenas permite aceder a um bloco de 8 bits (1 byte).

As operações permitidas sobre a memória são:

- Ler um byte da memória (*load byte*)
- Escrever um byte na memória (*store byte*)

Também é possível escrever vários bytes de uma vez.

Por exemplo:

- Ler uma word da memória (*load word*)
- Escrever uma word na memória (*store word*)

## Exemplo

Suponha que a memória de um computador contém a seguinte sequência de bits:

00000111000000001111111100000000111111...

- 1 Qual o conteúdo da memória nos endereços 0, 1, 2, 3, 4 e 5?
- 2 Represente em decimal e hexadecimal o conteúdo desses endereços.
- 3 Se os bytes representam números inteiros em complemento para 2, que números são?

Nota: Os números hexadecimais são representados com prefixo 0x.  
Por exemplo: 10 = 0xA.

## Problema

Quando se pretende escrever um número com mais de um byte para a memória surge um problema:

- Qual a ordenação dos bytes a usar? *i.e.*, deve colocar-se primeiro os bytes mais significativos ou menos significativos?

## Problema

Quando se pretende escrever um número com mais de um byte para a memória surge um problema:

- Qual a ordenação dos bytes a usar? *i.e.*, deve colocar-se primeiro os bytes mais significativos ou menos significativos?

## Definição (Endianness)

Existem duas convenções em uso:

**Little endian** o byte menos significativo primeiro. É usado na arquitectura Intel.

**Big endian** o byte mais significativo primeiro. É usado na arquitectura PowerPC.



## Problema

Quando se pretende escrever um número com mais de um byte para a memória surge um problema:

- Qual a ordenação dos bytes a usar? *i.e.*, deve colocar-se primeiro os bytes mais significativos ou menos significativos?

## Definição (Endianness)

Existem duas convenções em uso:

**Little endian** o byte menos significativo primeiro. É usado na arquitectura Intel.

**Big endian** o byte mais significativo primeiro. É usado na arquitectura PowerPC.

Algumas arquitecturas suportam ambos os modos. É o caso das arquitecturas MIPS e ARM.

São usados os termos MIPSEL e ARMEL para realçar que a ordenação de bytes usada é little endian.

Como escrever o número **0x12345678** em memória?

Endereço base →

0x78	LSB
0x56	
0x34	
0x12	MSB

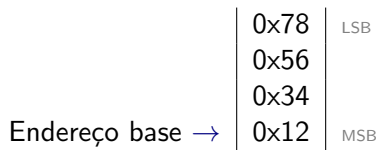
?

Endereço base →

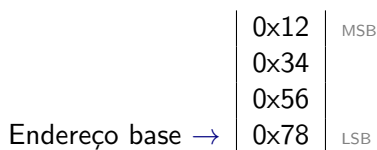
0x12	MSB
0x34	
0x56	
0x78	LSB

?

Como escrever o número **0x12345678** em memória?



Big Endian



Little Endian

A definição e o tratamento correcto da ordenação de bytes é especialmente importante quando é necessário transmitir informação entre máquinas com ordenações diferentes.

É o que acontece quando:

- escrevemos um ficheiro (e.g. música, imagem, etc) para ser lido noutra máquina.
- enviamos informação pela rede
- no interface com dispositivos da própria máquina (placas gráficas, rede, etc)

Quando uma grandeza é muito grande podem usar-se prefixos:

SI			IEC <sup>1</sup>		
$10^0 = 1$	(sem prefixo)	Unidade	$2^0$		
$10^3 = 1000$	k	kilo	$2^{10} = 1024$	Ki	kibi
$10^6$	M	mega	$2^{20}$	Mi	mebi
$10^9$	G	giga	$2^{30}$	Gi	gibi
$10^{12}$	T	tera	$2^{40}$	Ti	tebi
$10^{15}$	P	peta	$2^{50}$	Pi	pebi
$10^{18}$	E	exa	$2^{60}$	Ei	exbi
$10^{21}$	Z	zetta	$2^{70}$	Zi	zebi
$10^{24}$	Y	yotta	$2^{80}$	Yi	yobi

Pergunta: Uma memória com 2 GiB tem quantos bits?

---

<sup>1</sup>International Electrotechnical Commission

Quando uma grandeza é muito grande podem usar-se prefixos:

SI			IEC <sup>1</sup>		
$10^0 = 1$	(sem prefixo)	Unidade	$2^0$		
$10^3 = 1000$	k	kilo	$2^{10} = 1024$	Ki	kibi
$10^6$	M	mega	$2^{20}$	Mi	mebi
$10^9$	G	giga	$2^{30}$	Gi	gibi
$10^{12}$	T	tera	$2^{40}$	Ti	tebi
$10^{15}$	P	peta	$2^{50}$	Pi	pebi
$10^{18}$	E	exa	$2^{60}$	Ei	exbi
$10^{21}$	Z	zetta	$2^{70}$	Zi	zebi
$10^{24}$	Y	yotta	$2^{80}$	Yi	yobi

Pergunta: Uma memória com 2 GiB tem quantos bits?

Resposta:  $2\text{GiB} = 2 \times 2^{30} \times 8 \text{ bits} =$

---

<sup>1</sup>International Electrotechnical Commission

Quando uma grandeza é muito grande podem usar-se prefixos:

SI			IEC <sup>1</sup>		
$10^0 = 1$	(sem prefixo)	Unidade	$2^0$		
$10^3 = 1000$	k	kilo	$2^{10} = 1024$	Ki	kibi
$10^6$	M	mega	$2^{20}$	Mi	mebi
$10^9$	G	giga	$2^{30}$	Gi	gibi
$10^{12}$	T	tera	$2^{40}$	Ti	tebi
$10^{15}$	P	peta	$2^{50}$	Pi	pebi
$10^{18}$	E	exa	$2^{60}$	Ei	exbi
$10^{21}$	Z	zetta	$2^{70}$	Zi	zebi
$10^{24}$	Y	yotta	$2^{80}$	Yi	yobi

Pergunta: Uma memória com 2 GiB tem quantos bits?

Resposta:  $2\text{GiB} = 2 \times 2^{30} \times 8 \text{ bits} = 2^1 \times 2^{30} \times 2^3 \text{ bits} =$

---

<sup>1</sup>International Electrotechnical Commission

Quando uma grandeza é muito grande podem usar-se prefixos:

SI			IEC <sup>1</sup>		
$10^0 = 1$	(sem prefixo)	Unidade	$2^0$		
$10^3 = 1000$	k	kilo	$2^{10} = 1024$	Ki	kibi
$10^6$	M	mega	$2^{20}$	Mi	mebi
$10^9$	G	giga	$2^{30}$	Gi	gibi
$10^{12}$	T	tera	$2^{40}$	Ti	tebi
$10^{15}$	P	peta	$2^{50}$	Pi	pebi
$10^{18}$	E	exa	$2^{60}$	Ei	exbi
$10^{21}$	Z	zetta	$2^{70}$	Zi	zebi
$10^{24}$	Y	yotta	$2^{80}$	Yi	yobi

Pergunta: Uma memória com 2 GiB tem quantos bits?

Resposta:  $2\text{GiB} = 2 \times 2^{30} \times 8 \text{ bits} = 2^1 \times 2^{30} \times 2^3 \text{ bits} = 2^{34} \text{ bits}$ .

---

<sup>1</sup>International Electrotechnical Commission



A utilização não é consensual sendo necessário alguma cautela.  
A União Europeia introduziu legislação em 2007 para regular a utilização dos prefixos binários.

Alguns exemplos:

**Tamanhos dos ficheiros** Linux e OS X usam IEC, Windows usa SI.

A utilização não é consensual sendo necessário alguma cautela.  
A União Europeia introduziu legislação em 2007 para regular a utilização dos prefixos binários.

Alguns exemplos:

**Tamanhos dos ficheiros** Linux e OS X usam IEC, Windows usa SI.

**Memória** usam-se prefixos binários IEC, e.g. 4 GiB.

A utilização não é consensual sendo necessário alguma cautela.  
A União Europeia introduziu legislação em 2007 para regular a utilização dos prefixos binários.

Alguns exemplos:

**Tamanhos dos ficheiros** Linux e OS X usam IEC, Windows usa SI.

**Memória** usam-se prefixos binários IEC, e.g. 4 GiB.

**Discos rígidos** fabricantes usam sempre prefixos decimais, e.g.  
disco de 320 GB.  
(mas programas diferentes apresentam o tamanho do disco usando o prefixos diferentes...)

A utilização não é consensual sendo necessário alguma cautela.  
A União Europeia introduziu legislação em 2007 para regular a utilização dos prefixos binários.

Alguns exemplos:

**Tamanhos dos ficheiros** Linux e OS X usam IEC, Windows usa SI.

**Memória** usam-se prefixos binários IEC, e.g. 4 GiB.

**Discos rígidos** fabricantes usam sempre prefixos decimais, e.g.  
disco de 320 GB.  
(mas programas diferentes apresentam o tamanho do  
disco usando o prefixos diferentes...)

**Frequência de relógio** usa prefixo decimal, e.g. 2.66 GHz.

A utilização não é consensual sendo necessário alguma cautela.  
A União Europeia introduziu legislação em 2007 para regular a utilização dos prefixos binários.

Alguns exemplos:

**Tamanhos dos ficheiros** Linux e OS X usam IEC, Windows usa SI.

**Memória** usam-se prefixos binários IEC, e.g. 4 GiB.

**Discos rígidos** fabricantes usam sempre prefixos decimais, e.g.  
disco de 320 GB.  
(mas programas diferentes apresentam o tamanho do  
disco usando o prefixos diferentes...)

**Frequência de relógio** usa prefixo decimal, e.g. 2.66 GHz.

**CD** usa prefixo binário, e.g. 700 MiB.

A utilização não é consensual sendo necessário alguma cautela.  
A União Europeia introduziu legislação em 2007 para regular a utilização dos prefixos binários.

Alguns exemplos:

**Tamanhos dos ficheiros** Linux e OS X usam IEC, Windows usa SI.

**Memória** usam-se prefixos binários IEC, e.g. 4 GiB.

**Discos rígidos** fabricantes usam sempre prefixos decimais, e.g.  
disco de 320 GB.  
(mas programas diferentes apresentam o tamanho do  
disco usando o prefixos diferentes...)

**Frequência de relógio** usa prefixo decimal, e.g. 2.66 GHz.

**CD** usa prefixo binário, e.g. 700 MiB.

**DVD, Blu-ray** usa prefixo decimal, e.g. 4.7 GB.

- 1 Considere os números 127, 77, 1024, 0.
  - 1.1 Escreva-os em binário e em hexadecimal.
  - 1.2 Represente-os de modo a ocuparem quatro endereços de memória consecutivos.
  - 1.3 Repita para o simétrico de cada um dos números.
- 2 Quatro endereços de memória consecutivos contêm 0x01, 0x7f, 0xfc, 0x10, respectivamente. Sabendo que a ordenação de bytes é Little Endian e que os quatro bytes representam um número inteiro de 32 bits, escreva o número correspondente em hexadecimal.
- 3 Quantos bits contém um ficheiro de 4 MiB?