

EAE1106 - Métodos Computacionais para Economia

Fundamentos de Computação

Prof. Arthur Viaro

FEA-USP

Fevereiro, 2026

Plano da Aula

① Programa do Curso

② Sistema Computacional

③ Como um computador funciona?

Programa do Curso

- ▶ O programa e o material da disciplina podem ser consultados no link:
 - <https://arthurviaro.github.io/teaching/eae1106>

Plano da Aula

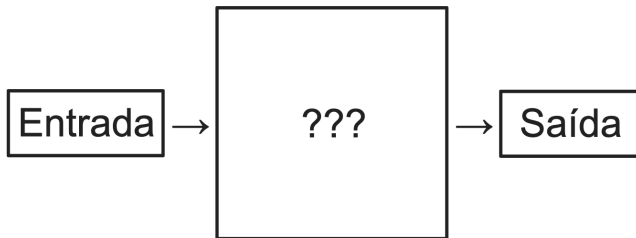
① Programa do Curso

② Sistema Computacional

③ Como um computador funciona?

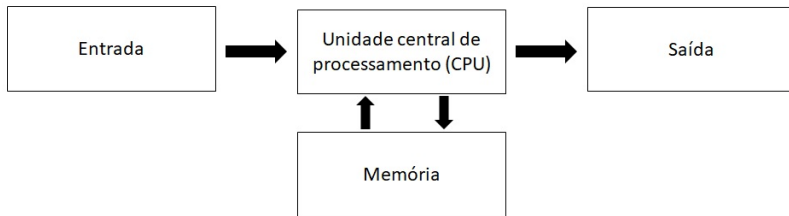
Sistema Computacional

- ▶ Podemos entender o funcionamento de um computador como um fluxo contínuo de leitura, processamento, armazenamento e saída de dados.
- ▶ O que acontece entre a entrada e a saída é o foco da ciência da computação.



Sistema Computacional

- ▶ Um sistema computacional é composto por dois elementos principais:
 1. **Hardware:** Componente físico do computador responsável por entrada, processamento, armazenamento e saída.
 2. **Software:** Componente lógico do sistema, ou seja, o conjunto de instruções que dizem ao hardware o que fazer.



- ▶ A Unidade Central de Processamento (CPU) é o principal componente de hardware:
 - Executa cálculos, processa dados e interpreta as instruções dos programas.
- ▶ A memória RAM (*Random Access Memory*) armazena temporariamente os dados e programas em execução.
 - Permite acesso rápido pelo processador.
 - Seu conteúdo é volátil, ou seja, é apagado quando o computador é desligado.
- ▶ Dispositivos como o disco rígido (HDD) ou a unidade de estado sólido (SSD) armazenam os dados de forma permanente.
 - Preserva as informações mesmo após o desligamento do computador.
 - Funciona como a “memória de longo prazo” do sistema.

Plano da Aula

① Programa do Curso

② Sistema Computacional

③ Como um computador funciona?

Como um computador funciona?

- ▶ Como representamos informações de forma padronizada em um computador?
- ▶ Por exemplo, para contar a presença em uma turma, poderíamos usar um sistema chamado **unário** (ou base 1), contando um dedo de cada vez.
- ▶ Já os computadores utilizam um sistema chamado **binário** (ou base 2), que usa apenas dois valores.
- ▶ Do termo *binary digit* surge a palavra **bit**.
 - Um bit pode assumir apenas dois estados: 0 ou 1 (“desligado” ou “ligado”).

Como um computador funciona?

- Dentro dos computadores existem milhões de transistores que tornam possíveis as atividades que realizamos diariamente em nossos dispositivos.



Como um computador funciona?

- Dentro dos computadores existem milhões de transistores que tornam possíveis as atividades que realizamos diariamente em nossos dispositivos.



Como um computador funciona?

- Dentro dos computadores existem milhões de transistores que tornam possíveis as atividades que realizamos diariamente em nossos dispositivos.



Como um computador funciona?

- Dentro dos computadores existem milhões de transistores que tornam possíveis as atividades que realizamos diariamente em nossos dispositivos.



Como um computador funciona?

- Dentro dos computadores existem milhões de transistores que tornam possíveis as atividades que realizamos diariamente em nossos dispositivos.



Como um computador funciona?

- Dentro dos computadores existem milhões de transistores que tornam possíveis as atividades que realizamos diariamente em nossos dispositivos.



Como um computador funciona?

- Dentro dos computadores existem milhões de transistores que tornam possíveis as atividades que realizamos diariamente em nossos dispositivos.



Como um computador funciona?

- Dentro dos computadores existem milhões de transistores que tornam possíveis as atividades que realizamos diariamente em nossos dispositivos.



- ▶ Mas qual é a lógica por trás do sistema binário?
- ▶ Para entender isso, vamos começar pelo que já conhecemos: o sistema decimal (base 10).
- ▶ No sistema decimal utilizamos 10 símbolos (0 a 9) e cada posição representa uma potência de 10.

123

100 10 1

123

100 10 1

123

$$100 \times 1 + 10 \times 2 + 1 \times 3$$

100 10 1

123

100 + 20 + 3

100 10 1

#

10^2 10^1 10^0

#

Sistema Binário

- O sistema binário funciona com potências de 2 e pode ser representado da seguinte maneira:

$$2^2 \quad 2^1 \quad 2^0$$



Sistema Binário

- O sistema binário funciona com potências de 2 e pode ser representado da seguinte maneira:

4	2	1
0	0	0

Sistema Binário

- O sistema binário funciona com potências de 2 e pode ser representado da seguinte maneira:

4	2	1
0	0	1

Sistema Binário

- O sistema binário funciona com potências de 2 e pode ser representado da seguinte maneira:

4	2	1
0	1	0

Sistema Binário

- O sistema binário funciona com potências de 2 e pode ser representado da seguinte maneira:

4	2	1
0	1	1

Sistema Binário

- O sistema binário funciona com potências de 2 e pode ser representado da seguinte maneira:

4	2	1
1	0	0

Sistema Binário

- O sistema binário funciona com potências de 2 e pode ser representado da seguinte maneira:

4	2	1
1	0	1

Sistema Binário

- O sistema binário funciona com potências de 2 e pode ser representado da seguinte maneira:

4	2	1
1	1	0

Sistema Binário

- O sistema binário funciona com potências de 2 e pode ser representado da seguinte maneira:

4	2	1
1	1	1

Sistema Binário

- O sistema binário funciona com potências de 2 e pode ser representado da seguinte maneira:

8	4	2	1
1	0	0	0

Sistema Binário

- ▶ Os computadores normalmente utilizam oito bits para representar informações.
- ▶ Um conjunto de 8 bits é chamado de **byte**.

128 64 32 16 8 4 2 1

00000000

- ▶ Com 8 bits, é possível representar valores de 0 a 255.
- ▶ Para representar números maiores, precisamos de mais bits!
 - Por exemplo, com 32 bits temos \approx 4 bilhões de bytes.

- ▶ Assim como os números são representados por padrões de 0 e 1, as letras também são codificadas em binário.
- ▶ O padrão chamado ASCII (*American Standard Code for Information Interchange*) foi criado para associar cada caractere a um número específico.
- ▶ Por exemplo, a letra "A" foi associada ao número "65".
 - Em binário, o número 65 é representado por "01000001".

ASCII

0	NUL	16	DLE	32	SP	48	0	64	@	80	P	96	`	112	p
1	SOH	17	DC1	33	!	49	1	65	A	81	Q	97	a	113	q
2	STX	18	DC2	34	"	50	2	66	B	82	R	98	b	114	r
3	ETX	19	DC3	35	#	51	3	67	C	83	S	99	c	115	s
4	EOT	20	DC4	36	\$	52	4	68	D	84	T	100	d	116	t
5	ENQ	21	NAK	37	%	53	5	69	E	85	U	101	e	117	u
6	ACK	22	SYN	38	&	54	6	70	F	86	V	102	f	118	v
7	BEL	23	ETB	39	'	55	7	71	G	87	W	103	g	119	w
8	BS	24	CAN	40	(56	8	72	H	88	X	104	h	120	x
9	HT	25	EM	41)	57	9	73	I	89	Y	105	i	121	y
10	LF	26	SUB	42	*	58	:	74	J	90	Z	106	j	122	z
11	VT	27	ESC	43	+	59	;	75	K	91	[107	k	123	{
12	FF	28	FS	44	,	60	<	76	L	92	\	108	l	124	
13	CR	29	GS	45	-	61	=	77	M	93]	109	m	125	}
14	SO	30	RS	46	.	62	>	78	N	94	^	110	n	126	~
15	SI	31	US	47	/	63	?	79	O	95	_	111	o	127	DEL

01001000

01001001

00100001

72

73

33

H

72

I

73

!

33

Unicode

- ▶ O padrão ASCII é limitado, pois não possui combinações suficientes para representar todos os caracteres utilizados no mundo.
- ▶ O padrão Unicode expandiu a quantidade de bits utilizados, permitindo que os computadores representem e interpretem uma enorme variedade de caracteres.



- ▶ Sequências de 0s e 1s também podem ser usadas para representar cores.
- ▶ O sistema **RGB** (Red, Green, Blue) representa cada cor como a combinação de três valores numéricos: vermelho, verde e azul.
 - Esses três valores (bytes) formam um **pixel** (ou ponto) da imagem.
- ▶ Por exemplo, os valores 72, 73 e 33 – que em ASCII poderiam representar “HI!” – quando interpretados como RGB formam uma cor.



- ▶ Assim, imagens digitais são simplesmente grandes coleções de valores RGB.
- ▶ Vídeos são sequências rápidas de imagens exibidas em ordem, como um *flipbook*.
- ▶ De forma semelhante, músicas também podem ser representadas por combinações de bytes que codificam sons.

- ▶ A partir dos bytes, surgem unidades maiores, amplamente utilizadas para medir o tamanho de arquivos, a quantidade de memória e a capacidade de armazenamento:
 - Kilobyte (KB): 1,024 (ou 2^{10}) bytes.
 - Megabyte (MB): 1,048,576 (ou 2^{20}) bytes.
 - Gigabyte (GB): 1,073,741,824 (ou 2^{30}) bytes.
 - Terabyte (TB): 1,099,511,627,776 (ou 2^{40}) bytes.
 - Petabyte (PB): 1,125,899,906,842,624 (ou 2^{50}) bytes.
 - Exabyte, Zettabyte, Yottabyte ...

O que vem pela frente?

- ▶ Arquivos e diretórios
- ▶ Terminal
- ▶ Linguagens de programação
- ▶ Algoritmos
- ▶ Pseudocódigo