

# *Tecnologia em Análise e Desenvolvimento de Sistemas - TADS*

## ***Fundamentos da Computação***

*TADS – IFRS – 2023/1*

***Prof. Luciano Vargas Gonçalves***



# Aula 2

## ***Aula 2 – Representação da Informação***

# Computadores Digitais

- Processamento das Informações e o Sistema Binário



**Informações de Entradas**

**Saída da Informação**

# Representação da Informação

Existem diversas fontes de informação (dados).

- Como representar imagens, sons, textos e números decimais nos circuitos do computador?
- Computador é um equipamento eletrônico, composto por inúmeros circuitos digitais.



FOTOS



VÍDEOS



FRASES



GRÁFICOS



CHECKLISTS



APRESENTAÇÕES



INFOGRÁFICOS

# Representação da Informação

- Textos
  - Os textos são formados por vários símbolos chamados *caracteres*;
  - Cada carácter é representado por um número inteiro, de acordo com tabelas de codificação
    - Uma das mais conhecidas é a tabela ASCII
    - Link:
      - <https://pt.wikipedia.org/wiki/ASCII>

# Parte da Tabela ASCII

Cada carácter (CHAR) possui um código numérico:  
Representado em por um valor:

***Dec = Decimal***

***Hex = Hexadecimal***

Os códigos variam de 0 a 127 na versão ASC  
padrão ([Link 1](#), [Link 2](#))

Os códigos variam de 0 a 255 na versão ASCII  
estendida (8bits);  
Codificação com acentuação.

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	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	0	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	_	127	7F	□

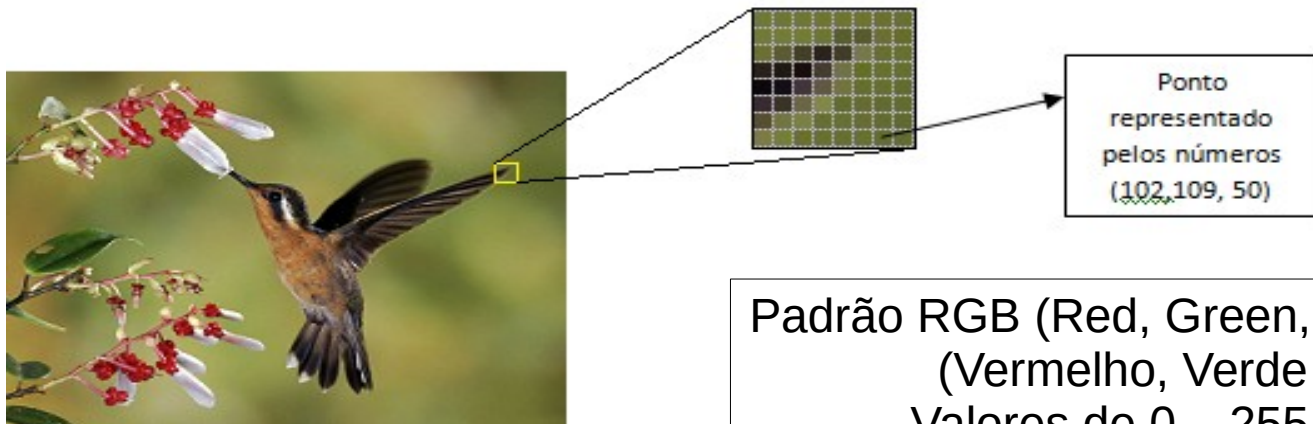
# Exemplo de Texto

- ABba

Carácter		Código ASC II Decimal	
–	A	=	65
–	B	=	66
–	b	=	98
–	a	=	97

# Imagens

- Cada imagem é dividida em pontos chamados ***Pixel***
- Cada pixel contém uma cor, representada por um ou três números inteiros

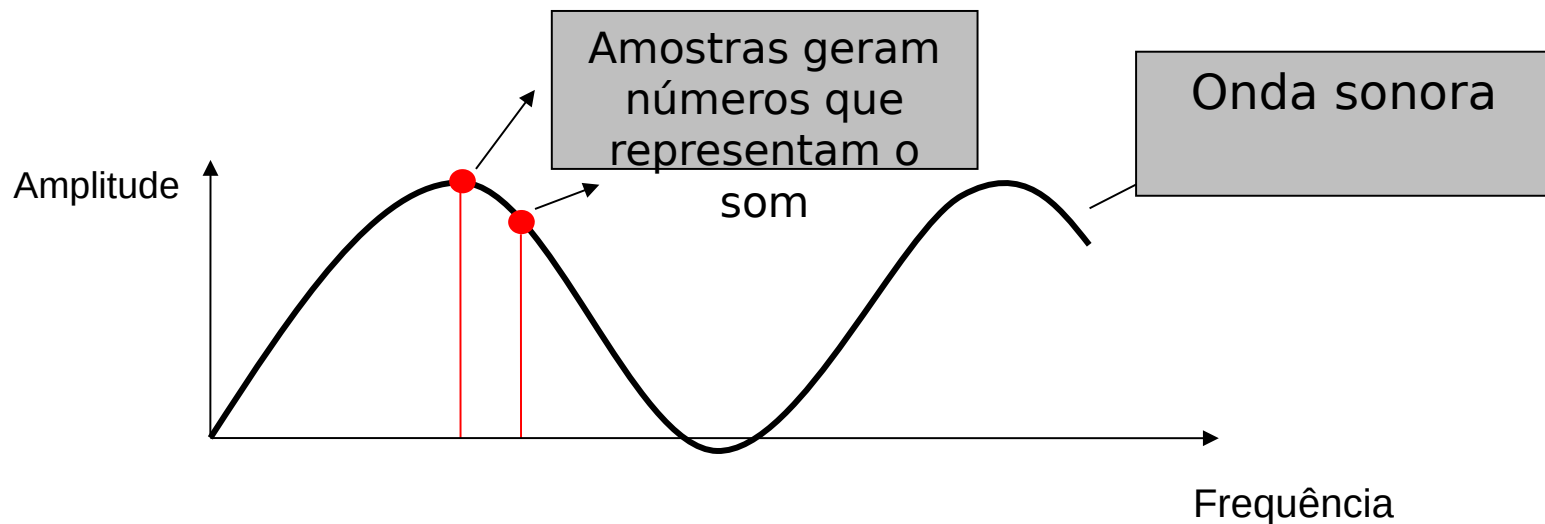


Padrão RGB (Red, Green, Blue)  
(Vermelho, Verde e Azul)  
Valores de 0 – 255, intensidade da cor  
Branco (255,255,255)  
Preto (0,0,0)



# Sons

Números representam amostras da onda sonora (amplitude, frequência, etc)



# A Língua do Computador

- ***Linguagem de Máquina ou linguagem do processador***
  - Tudo é representado na forma de números inteiros
- Cada instrução diferente é representada por um número diferente
  - Por exemplo
    - Instrução de soma (ADD) = código 1
    - Instrução de entrada (Load) = código 5
    - Etc.
- Cada tipo de processador possui sua própria linguagem de máquina
  - Exemplos:
    - Família X86, AMD64

# Representação da Informação

- **Por que toda a informação é representada por números inteiros?**
  - Números inteiros podem ser facilmente representados na memória RAM do computador;
    - Isso se deve ao fato de como a memória é organizada “em **Células de 8bits ou 1Byte**”;
    - Cada Byte pode representar valores de 0 à 255;
      - Semelhante a tabela ASCII

# Organização da Memória

Célula {

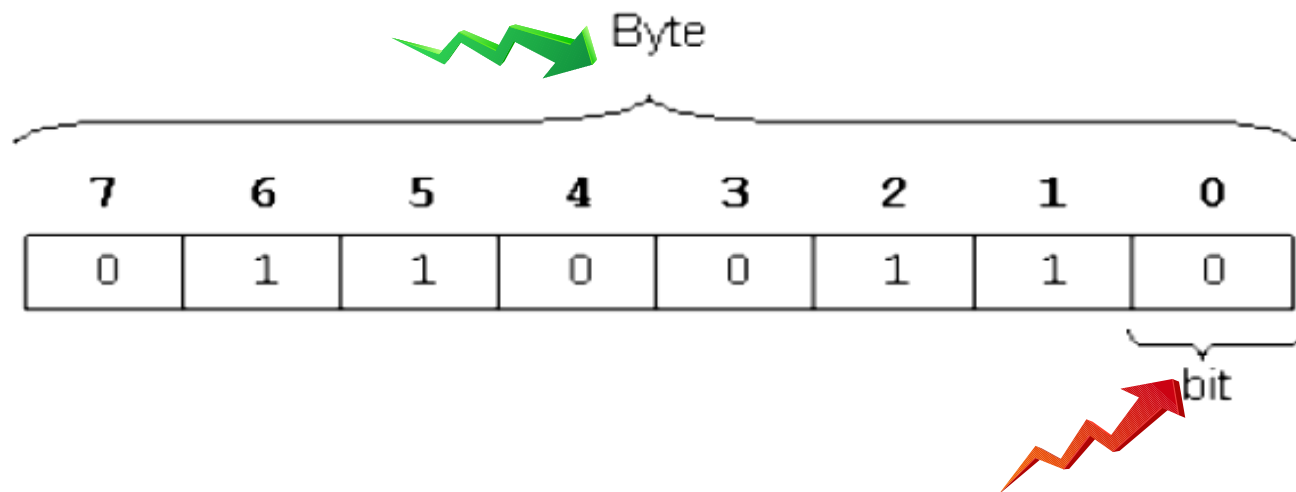
Endereço	Conteúdo da Célula
0	XXXXXX ← Informação
1	XXXXXX
2	XXXXXX
3	XXXXXX
4	XXXXXX
5	XXXXXX
⋮	⋮

8bits = 1Byte

- Memória é organizada em células;
- Cada célula armazena 1Byte (8 bits) que corresponde a informação armazenada;
- Cada célula tem um endereço sua para localização;

# Células da memória

- Cada célula é conhecida como *byte*
- Cada byte é formado por 8 *bits*



Bit (menor unidade de informação)  
0 ou 1

- Exemplo: Pente de memória 2GB (dois GigaBytes ou ~2 bilhões de Bytes ou Células de memória)

# Bytes na memória

- Cada célula armazena oito bits, cada bit pode assumir um dos valores (Zero (0 - Desligado) ou UM(1 - Ligado))

Endereço	Conteúdo da Célula							
0	0	0	0	0	1	1	0	0
1	1	1	1	1	1	1	0	0
2	0	1	1	0	0	1	1	0
3	0	1	0	1	0	1	0	1
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	1
⋮	⋮							

Memória com valores armazenados

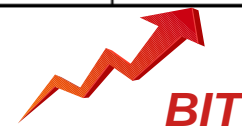


# Bytes na memória

- Todo valor binário pode ser convertido para decimal e vice-versa.

Endereço	Conteúdo da Célula							
0	0	0	0	0	1	1	0	0
1	1	1	1	1	1	1	0	0
2	0	1	1	0	0	1	1	0
3	0	1	0	1	0	1	0	1
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	1
⋮	⋮							

Memória com valores armazenados



# Informação e sua Representação

Vídeo para exemplificar a informação e suas representações

- <https://www.youtube.com/watch?v=uIG1SIOqn9s>



# Sistema Binário

## John von Neumann: 1945

Sugeri que o sistema binário fosse adotado em todos os computadores, e que as ***instruções e dados*** fossem compilados e armazenados internamente no computador, na sequência correta de utilização.

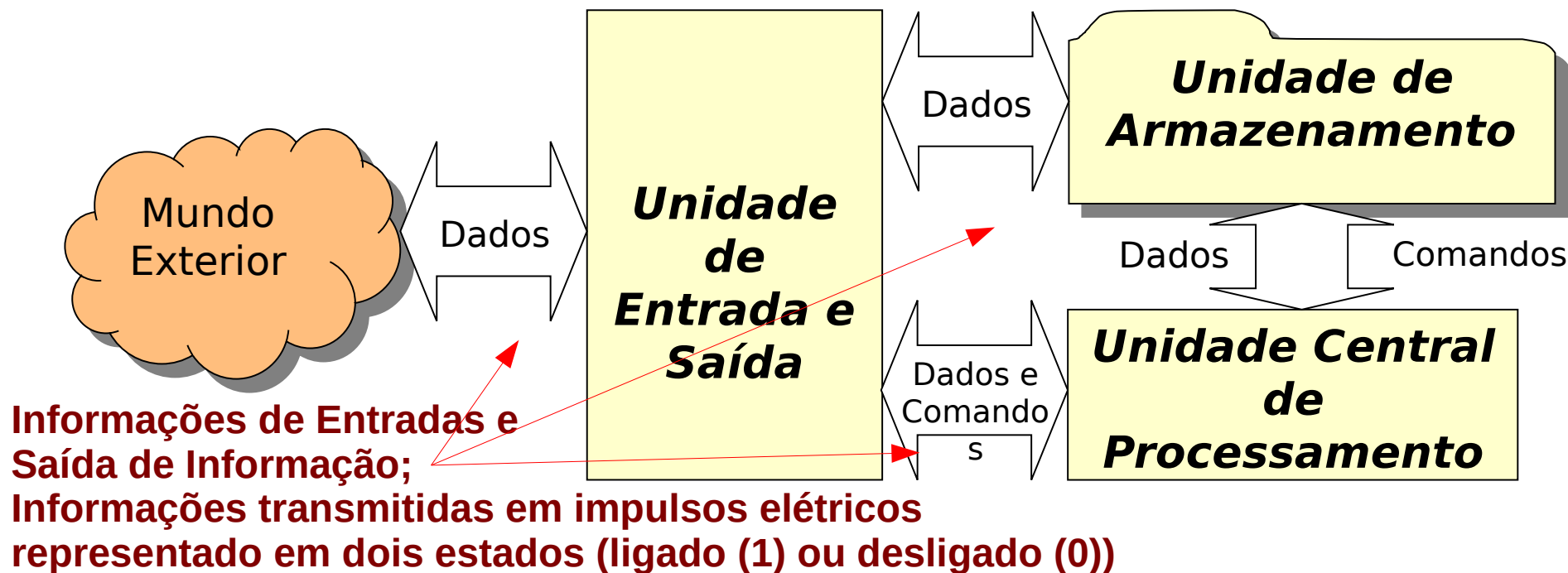
Mais tarde esse computador foi construído recebendo o nome de EDVAC.



John von Neumann

# Computadores Digitais













- Processamento das Informações (Dados Binários)



# Sistema Binário

- **Sistema binário**

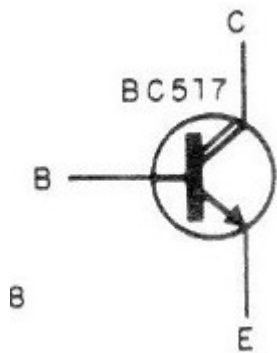
- Um bit pode assumir dois valores
  - Zero (desligado)
  - Um (ligado)
- Por quê?
  - Representam a variação da grandeza Tensão Elétrica.
  - Exemplo:
    - 0 – 2 volts = Zero (0)
    - 4 a 5 volts = Um (1)
- ***Fácil leitura e interpretação pelos circuitos eletrônicos.***

				=	0	1	0	1	=	5
				=	1	0	0	1	=	9
<hr/>										
				=	1	1	1	0	=	14

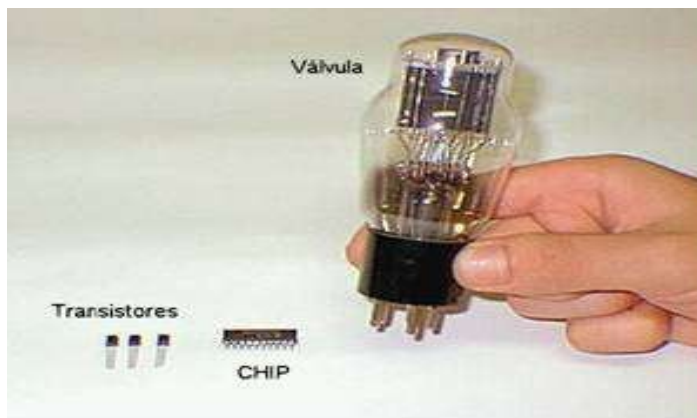
Operação de Soma Binária

# Sistema Binário

- ***Componentes principais – primeiros computadores***
  - Chave automática foi implementada através de:
    - Relés >> Válvulas >> transistores >> Circuitos integrados
    - Elementos da Eletrônica para representar **bits**.



Espécie de Interruptor (liga e desliga)



a



b

# Sistema Binário

- Utiliza apenas dois símbolos para representar números:
  - 0 (zero - desligado) e o 1 (um - Ligado)

**0 1**



- Que coincidência, é exatamente o que cabe em um bit!!!

# Sistema Binário

**Relação entre a quantidade de bits e as representações das informações.**

***o total de representações =  $2^n$***

onde n = número de bits

com 1 bit =  $2^1 = 2$  valores

N(nºbits)	Valores
1	0
	1

# Sistema Binário

**Relação entre a quantidade de bits e as representações das informações.**

***o total de representações =  $2^n$***

onde n = número de bits

com 1 bit =  $2^1 = 2$  valores

com 2 bits =  $2^2 = 4$  valores

N(n°bits)	Valores
2	00
	01
	10
	11

# Sistema Binário

**Relação entre a quantidade de bits e as representações das informações.**

***o total de representações =  $2^n$***

onde n = número de bits

com 1 bit =  $2^1$  = 2 valores

com 2 bits =  $2^2$  = 4 valores

com 3 bits =  $2^3$  = 8 valores

com 4 bits =  $2^4$  = 16 valores

N(n°bits)	Valores
2	00
	01
	10
	11



# Sistema Binário

**Relação entre a quantidade de bits e as representações das informações.**

***O total de representações =  $2^n$***

onde n = número de bits

com 1 bit =  $2^1 = 2$  valores

com 2 bits =  $2^2 = 4$  valores

com 3 bits =  $2^3 = 8$  valores

com 4 bits =  $2^4 = 16$  valores

Representações	1 Bit	2 Bits	3 Bits	Com 4 Bits
	0	0 0	0 0 0	0 0 0 0
$2^1 = 2$	1	0 1	0 0 1	0 0 0 1
		1 0	0 1 0	0 0 1 0
$2^2 = 4$		1 1	0 1 1	0 0 1 1
			1 0 0	0 1 0 0
			1 0 1	0 1 0 1
			1 1 0	0 1 1 0
$2^3 = 8$			1 1 1	0 1 1 1
				1 0 0 0
				1 0 0 1
				1 0 1 0
				1 0 1 1
				1 1 0 0
				1 1 0 1
				1 1 1 0
$2^4 = 16$				1 1 1 1

# Sistema Binário

**Relação entre a quantidade de bits e as representações das informações.**

***O total de representações =  $2^n$***

onde n = número de bits

com 1 bit =  $2^1$  = 2 valores

com 2 bits =  $2^2$  = 4 valores

com 3 bits =  $2^3$  = 8 valores

com 4 bits =  $2^4$  = 16 valores

com 5 bits =  $2^5$  = \_\_\_\_\_

Representações	Com 5 Bits
	0 0 0 0 0
	0 0 0 0 1
	0 0 0 1 0
	0 0 0 1 1
	0 0 1 0 0
	0 0 1 0 1
	0 0 1 1 0
$2^5 = 32$	0 0 1 1 1
	0 1 0 0 0
	0 1 0 0 1
	0 1 0 1 0
	0 1 0 1 1
	0 1 1 0 0
	0 1 1 0 1
	0 1 1 1 0
	0 1 1 1 1

# Sistema Binário

**Relação entre a quantidade de bits e as representações das informações.**

***O total de representações =  $2^n$***

onde n = número de bits

com 1 bit =  $2^1$  = 2 valores

com 2 bits =  $2^2$  = 4 valores

com 3 bits =  $2^3$  = 8 valores

com 4 bits =  $2^4$  = 16 valores

com 5 bits =  $2^5$  = \_\_\_\_\_

Representações	Com 5 Bits
	1 0 0 0 0
	1 0 0 0 1
	1 0 0 1 0
	1 0 0 1 1
	1 0 1 0 0
	1 0 1 0 1
	1 0 1 1 0
$2^5 = 32$	1 0 1 1 1
	1 1 0 0 0
	1 1 0 0 1
	1 1 0 1 0
	1 1 0 1 1
	1 1 1 0 0
	1 1 1 0 1
	1 1 1 1 0
	1 1 1 1 1

# Representação dos Números

- Existem formas diferentes de dizer a mesma coisa;
- Existem formas diferentes de representar grandezas;
  - Exemplo (10):
    - 10(decimal)
    - X (romano)
    - Dez (extenso)
- Nós comumente utilizamos o *sistema decimal* para representar os valores (grandezas).
  - Dinheiro, objetos, medidas e etc.

# Sistema Decimal = 10 símbolos

- Utiliza dez algarismos (base 10) para representar qualquer número.
  - Algarismos: 0,1,2,3,4,5,6,7,8,9
- As posições dos algarismos nos números indica seu valor
  - Sistema Posicional

$$\bullet \quad 132_{10} = 1^2 \ 3^1 \ 2^0 = 1 \times 10^2 + 3 \times 10^1 + 2 \times 10^0$$

*Posição = Expoente*

$$\bullet \quad 5078_{10} = 5^3 \ 0^2 \ 7^1 \ 8^0 = 5 \times 10^3 + 0 \times 10^2 + 7 \times 10^1 + 8 \times 10^0$$
$$5000 + 0 + 70 + 8 = 5078$$

# Sistema de Numeração

- Podemos utilizar outras quantidades de algarismos (Bases) para representar grandezas(valores).
- A isso se dá o nome de ***Sistemas de Numeração***
  - Cada sistema de numeração é identificado por uma **base**, que indica a quantidade de algarismos:
    - Sistema Decimal (**base<sub>10</sub> = 10**)  $\leftrightarrow$  (0,1,2,3,4,5,6,7,8,9)
    - Sistema Binário (**base<sub>2</sub> = 2**)  $\leftrightarrow$  (0,1)
    - Sistema OctaDecimal (**base<sub>8</sub> = 8**)  $\leftrightarrow$  (0,1,2,3,4,5,6,7)
    - Sistema HexaDecimal (**base<sub>16</sub> = 16**)  $\leftrightarrow$  (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F)



# **Sistemas de Numerações**

# Sistema Decimal – Base<sub>10</sub>

- Sistema Decimal (Dez algarismo):
  - Nosso Sistema de Contagem:
  - Possui 10 símbolos para representar grandezas
    - **0 – 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9**
  - Base (10)
- Sistema Posicional ou polinomial



# Sistema Decimal – Base <sub>10</sub>

## Notação Polinomial ou Posicional

- Válida para qualquer base numérica.
- LEI DE FORMAÇÃO
- (Notação ou Representação Polinomial):

$$\text{Número} = a_n b^n + a_{n-1} b^{n-1} + a_{n-2} b^{n-2} + \dots + a_0 b^0$$

- $a_n$  = algarismo,  $b$  = base do número,  $n$  = (qtd de algarismo - 1) ou posição

Exemplo:

- Valor 5437 tem 4 elementos ( $n = 3$ )

# Sistema Decimal – Base <sub>10</sub>

## Notação Polinomial ou Posicional

- Exemplos:

$$\text{Número} = a_n b^n + a_{n-1} b^{n-1} + a_{n-2} b^{n-2} + \dots + a_0 b^0$$

- $123_{10} \Leftrightarrow (n = 2)$

$$123_{10} = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 \quad \leftarrow \text{Posição}$$

$$123_{10} = 1 \times 100 + 2 \times 10 + 3 \times 1$$

$$123_{10} = 100 + 20 + 3$$

$$123_{10} = 123_{10}$$

# Sistema Decimal – Base <sub>10</sub>

- Próximo valor será o anterior + 1

- $0 = 0$

- $0 + 1 = 1$

- $1 + 1 = 2$

- $2 + 1 = 3$

- $9 + 1 = 10$

- Houve estouro Buffer para valor 10, se considerar apenas um dígito, conhecido com “Overflow”



Vai um

0	1	2	9
<u>+1</u>	<u>+1</u>	<u>+1</u>	<u>+1</u>
1	2	3	10

Adição Decimal

Decimal	Decimal
0	0
1	0 + 1 = 1
2	1 + 1 = 2
3	2 + 1 = 3
4	3 + 1 = 4
5	4 + 1 = 5
...	
9	8 + 1 = 9
10	9 + 1 = 10

# Sistema Binário – Base <sub>2</sub>

- Sistema Binário (Dois algarismo):
  - Sistema dos equipamentos digitais
  - Possui 2 símbolos para representar grandezas
    - **0 – 1**
  - Base (2)
- Sistema Posicional ou polinomial

# Sistema Binário – Base <sub>2</sub>

- Sistema Binário (Dois algarismo):
  - Sistema dos equipamentos digitais
  - Possui 2 símbolos para representar grandezas

• **0 , 1**

- **Sistema de contagem:**


• **Anterior + 1 = próximo**

• **0 = 0**

• **0+1 = 1**

• **1+1 = 10 (Overflow) para um bit**

Vai um



$$\begin{array}{r}
 0 \quad 0 \quad 1 \quad 1 \\
 +0 \quad +1 \quad +0 \quad +1 \\
 \hline
 0 \quad 1 \quad 1 \quad 10
 \end{array}$$

Adição binária

Decimal	Binário
0	0
1	0+1 = 1
2	1+1 = 10
3	10+1 = 11
4	11+1 = 100
5	100+1 = 101
6	101+1 = 110
7	110+1 = 111
8	111+1 = 1000
9	1000+1 = 1001
10	1001+1 = 1010
11	1010+1 = 1011
12	1011+1 = 1100
13	1100+1 = 1101
14	1101+1 = 1110
15	1110+1 = 1111
16	1111+1 = 10000

# Sistema Binário – Base <sub>2</sub>

## Notação Polinomial ou Posicional

- Exemplos:

$$1101_2 =$$

$$\text{Número} = a_n b^n + a_{n-1} b^{n-1} + a_{n-2} b^{n-2} + \dots + a_0 b^0$$

- $1101_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

- $1101_2 = 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1$

$$1101_2 = 8 + 4 + 0 + 1$$

$$1101_2 = 13_{10}$$


Base dos valores;

# Sistema Octal – Base <sub>8</sub>

- Sistema Octal (Oito algarismo):
  - Sistema dos equipamentos digitais
  - Possui 8 símbolos para representar grandezas
    - **0 – 1 – 2 – 3 – 4 – 5 – 6 – 7**
  - Base (8)
- Sistema Posicional ou polinomial

# Sistema Octal – Base <sub>8</sub>

- Sistema Octal (Oito algarismo):
  - Possui 8 símbolos para representar grandezas
    - **0,1,2,3,4,5,6,7**


$$\begin{array}{cccc} 0 & 1 & 2 & 7 \\ +1 & +1 & +1 & +1 \\ \hline 1 & 2 & 3 & 10 \end{array}$$

Adição binária

Decimal	Octal	Decimal
0	0	
1	1	
2	2	
3	3	
4	4	
5	5	
6	6	
7	7	
8	10	
9	11	
10	12	
11	13	
12	14	
13	15	
14	16	
15	17	
16	20	



# Sistema Octal – Base <sub>8</sub>

## Notação Polinomial ou Posicional

- Exemplos:

$$257_8 =$$

$$\text{Número} = a_n b^n + a_{n-1} b^{n-1} + a_{n-2} b^{n-2} + \dots + a_0 b^0$$

$$257_8 = 2 \times 8^2 + 5 \times 8^1 + 7 \times 8^0$$

$$257_8 = 2 \times 64 + 5 \times 8 + 7 \times 1$$

$$257_8 = 128 + 40 + 7$$

$$257_8 = 175_{10}$$

# Sistema Hexadecimal – Base<sub>16</sub>

- Sistema Hexadecimal (Dezesseis algarismo):
  - Sistema dos equipamentos digitais
  - Possui 16 símbolos para representar grandezas
    - **0 – 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – A – B – C – D – E – F**
  - Base (16)
- Sistema Posicional ou polinomial

# Sistema Hexadecimal – Base <sub>16</sub>

- Sistema Hexadecimal (Dezesseis algarismo):
  - Sistema dos equipamentos digitais
  - Possui 16 símbolos para representar grandezas

- **0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F**

$$\begin{array}{r}
 0 \quad 1 \quad 9 \quad F \\
 +1 \quad +1 \quad +1 \quad +1 \\
 \hline
 1 \quad 2 \quad A \quad 10
 \end{array}$$

Adição HexaDecinal



Decimal	Hexadecimal
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	A
11	B
12	C
13	D
14	E
15	F
16	10

# Sistema Hexadecimal – Base <sub>16</sub>

## Notação Polinomial ou Posicional

- Exemplos:

$$2A2_{16} =$$

$$\text{Número} = a_n b^n + a_{n-1} b^{n-1} + a_{n-2} b^{n-2} + \dots + a_0 b^0$$

$$2A2_{16} = 2 \times 16^2 + A \times 16^1 + 2 \times 16^0$$

$$2A2_{16} = 2 \times 256 + A \times 16 + 2 \times 1$$

$$2A2_{16} = 512 + 10 \times 16 + 2$$

$$2A2_{16} = 512 + 160 + 2$$

$$2A2_{16} = 674_{10}$$

	Decimal <sub>10</sub>	Binário <sub>2</sub>	Octal <sub>8</sub>	Hexadecimal <sub>16</sub>	
	0	0	0	0	
	1	1	1	1	
	2	10	2	2	
	3	11	3	3	
	4	100	4	4	
	5	101	5	5	
	6	110	6	6	
	7	111	7	7	
	8	1000	10	8	
	9	1001	11	9	
	10	1010	12	A	
	11	1011	13	B	
	12	1100	14	C	
	13	1101	15	D	
	14	1110	16	E	
	15	1111	17	F	
	16	10000	20	10	

# Converter de Qualquer Base >> Decimal

## Notação Polinomial ou Posicional

- Válida para qualquer base numérica.
- LEI DE FORMAÇÃO
- (Notação ou Representação Polinomial):

$$\text{Número} = a_n b^n + a_{n-1} b^{n-1} + a_{n-2} b^{n-2} + \dots + a_0 b^0$$

- $a_n$  = algarismo,  $b$  = base do número
- $n$  = quantidade de algarismo - 1

# Sistema Binário >> Decimal

$$\text{Número} = a_n b^n + a_{n-1} b^{n-1} + a_{n-2} b^{n-2} + \dots + a_0 b^0$$

- Exemplo:

- Número em binário =  $101011_2$  (n = 6 termos)

$$\mathbf{101011_2 = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 43_{10}}$$

- Número:  $101011_2 = 43_{10}$  **Decimal**

# Sistema Octal >> Decimal

$$\text{Número} = a_n b^n + a_{n-1} b^{n-1} + a_{n-2} b^{n-2} + \dots + a_0 b^0$$

- Exemplo:

- Número  $327_8$  (n = 3 termos)

$$327_8 = 3 \times 8^2 + 2 \times 8^1 + 7 \times 8^0 = 215_{10}$$

- Número:  $327_8 = 215_{10}$  ***Decimal***



# Sistema Hexadecimal >> Decimal

$$\text{Número} = a_n b^n + a_{n-1} b^{n-1} + a_{n-2} b^{n-2} + \dots + a_0 b^0$$

- Exemplo:

- Número  $16_{16}$  (n = 2 termos)

$$\mathbf{16_{16}} = 1.16^{2-1} + 6.16^{2-2} = 16 + 6 = 22_{10}$$

- Número:  $16_8 = 22_{10}$  ***Decimal***

# Sistema Hexadecimal >> Decimal

$$\text{Número} = a_n b^n + a_{n-1} b^{n-1} + a_{n-2} b^{n-2} + \dots + a_0 b^0$$

Exemplo:

- Número  $A2F_{16}$  (n = 3 termos)

$$A2F_{16} = A \times 16^2 + 2 \times 16^1 + F \times 16^0 = 2607_{10}$$

$$A2F_{16} = 10 \times 16^2 + 2 \times 16^1 + 15 \times 16^0 = 2607_{10}$$

- Número:  $A2F_{16} = 2607_{10}$  *Decimal*

# Exercícios

- Converta para decimal

1)  $101001_2 =$

2)  $11010111_2 =$

3)  $146_8 =$

4)  $852_8 =$

5)  $123_{16} =$

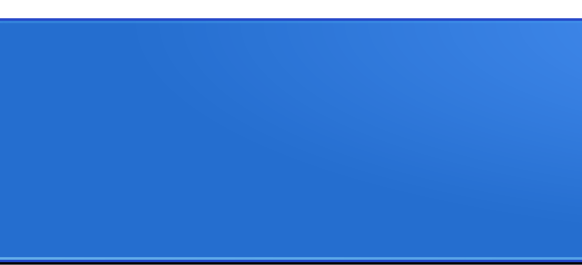
6)  $ABC_{16} =$

7)  $DEF_{16} =$

8)  $2C3_{16} =$

Complete a tabela

Decimal <sub>10</sub>	Binário <sub>2</sub>	Octal <sub>8</sub>	Hexadecimal <sub>16</sub>
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			
27			
28			
29			
30			
31			
32			
33			



Complete a tabela

Decimal <sub>10</sub>	Binário <sub>2</sub>	Octal <sub>8</sub>	Hexadecimal <sub>16</sub>
34			
35			
36			
37			
38			
39			
40			
41			
42			
43			
44			
45			
46			
47			
48			
49			
50			



Complete a tabela

Decimal <sub>10</sub>	Binário <sub>2</sub>	Octal <sub>8</sub>	Hexadecimal <sub>16</sub>
100			
101			
102			
103			
104			
105			
106			
150			
151			
152			
153			
154			
155			
200			
201			
202			
203			



## Conversão de Bases

Próxima aula!!!

Perguntas??