

Sistemas Numéricos



Universidade Federal do Ceará - Campus Quixadá

Pedro Botelho
pedro.botelho@ufc.br

18 de Setembro de 2025

Arquitetura de Computadores

Seção 1

Representação de Informações

A informação e sua representação

- O computador, sendo um equipamento eletrônico, **armazena e movimenta as informações** internamente sob forma eletrônica.
- Tudo o que um computador faz é reconhecer **dois estados físicos distintos**
 - Produzidos pela eletricidade (e.g. SSD), pela polaridade magnética (e.g. HDD) ou pela luz refletida (e.g. Blu-ray).
 - Em essência, eles sabem dizer se um “interruptor” está **ligado** ou **desligado**.

A informação e sua representação

- Por ser uma máquina eletrônica, ele só consegue processar duas informações: a **presença** ou **ausência** de energia.
- Para que uma máquina pudesse representar eletricamente todos os símbolos utilizados na linguagem humana, seriam necessários mais de 100 diferentes valores de tensão (ou de corrente).

Tipos de grandezas

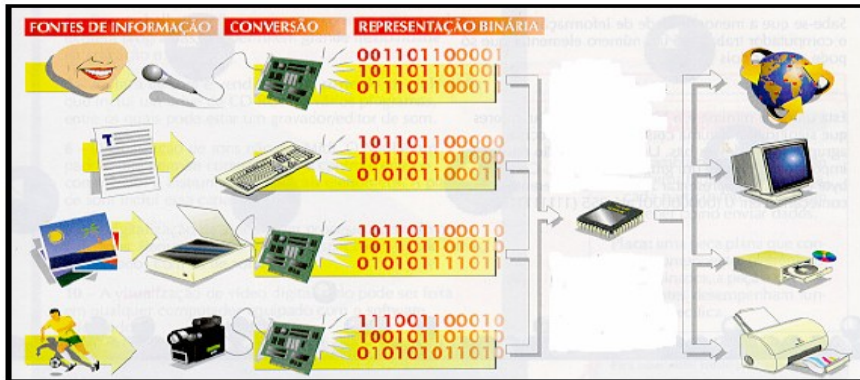
- **Digital:** são aquelas que não variam continuamente no tempo, mas sim em saltos entre valores bem definidos (discreta - passo a passo).
 - Exemplo: Relógios digitais, apesar do tempo em si variar continuamente, o visor do relógio mostra o tempo em saltos de um em um segundo.
- **Analógica:** variam continuamente dentro de uma faixa de valores (contínua).
 - Exemplo: O velocímetro de um carro, pois o ponteiro gira continuamente conforme o automóvel acelera ou freia. Se o ponteiro girasse em saltos, o velocímetro seria considerado digital.

Tipos de grandezas

- **Computadores analógicos** – Trabalham com sinais elétricos de infinitos valores de tensão e corrente (modelo continuamente variável, ou analogia, do que quer que estejam medindo).
 - Intrinsecamente mais preciso, porém mais complexo de se trabalhar e menos confiável
 - Exemplo: Sinal analógico de TV precisa de equipamento mais complexo e está sujeito a ruído
- **Computadores digitais** – Trabalham com dois níveis de sinais elétricos: **alto** e **baixo**. Representam dados por meio de um símbolo facilmente identificado (dígito).
 - Intrínsecamente menos preciso, porém menos complexo de se trabalhar e mais confiável
 - Exemplo: Sinal digital de TV é mais robusto contra ruído

Representação da informação

Como os computadores modernos representam as informações?



Representação da informação

Para o computador, tudo são **números**.

- No computador digital, normalmente a informação a ser processada é de forma numérica ou texto.
- Essa informação é codificada internamente através de um **código numérico**.
- O código mais comum é o **código binário**.

Por que é utilizado o sistema binário?

Por que binário?

Como os computadores representam as informações utilizando apenas **dois estados possíveis** - eles são totalmente adequados para números binários. 0 - desligado e 1 - ligado).

- Número binário no computador: **bit** [de “Binary digIT”]
 - A unidade de informação.
 - Uma quantidade computacional que pode tomar um de dois valores, tais como verdadeiro e falso ou 1 e 0, respectivamente (lógica positiva).

Um bit está ligado(*set*) quando vale 1, desligado ou limpo (*reset or clear*) quando vale 0; comutar, ou inverter (*toggle or invert*) é passar de 0 para 1 ou de 1 para 0.

Conjunto de bits

- Um bit pode representar apenas 2 símbolos (0 ou 1).
- Desse modo, é necessário uma unidade maior, formada por um **conjunto de bits**, para representar números e outros símbolos, como os caracteres e os sinais de pontuação que usamos nas linguagens escritas.
- Precisamos ter bits suficientes para representar **todos os símbolos** que possam ser usados:
 - dígitos numéricos,
 - letras maiúsculas e minúsculas do alfabeto,
 - sinais de pontuação,
 - símbolos matemáticos e assim por diante.

Quantos símbolos?

Necessidade:

| | |
|---------------------------------------|------------|
| Caracteres alfabéticos maiúsculos | 26 |
| Caracteres alfabéticos minúsculos | 26 |
| Algarismos | 10 |
| Sinais de pontuação e outros símbolos | 32 |
| Caracteres de controle | 24 |
| Total | 118 |

Quantos bits?

Capacidade de representação:

| | |
|----------|------------|
| 2 | 4 |
| 3 | 8 |
| 4 | 16 |
| 5 | 32 |
| 6 | 64 |
| 7 | 128 |
| 8 | 256 |
| \vdots | \vdots |
| n | 2^n |

E o byte?

Byte (Binary TErm):

- Grupo ordenado de 8 bits, para efeitos de manipulação interna mais eficiente.
- Tratado de forma individual, como unidade de armazenamento e transferência.
- Unidade de memória usada para representar um **caractere**.

E o byte?

- Com 8 bits, podemos arranjar 256 configurações diferentes: dá para 256 caracteres, ou para números de 0 a 255, ou de -128 a 127, por exemplo.
- O termo `bit` apareceu em 1949, inventado por John Tukey, um pioneiro dos computadores. Segundo Tukey, era melhor que as alternativas *bigit* ou *binit*.
- O termo `byte` foi criado por Werner Buchholz em 1956 durante o desenho do computador IBM Stretch. Inicialmente era um grupo de 1 a 6 bits, mas logo se transformou num de 8 bits.

Representação da informação

- Todas as letras, números e outros caracteres são **codificados** e **decodificados** pelos equipamentos através dos bytes que os representam, permitindo, dessa forma, a comunicação entre o usuário e a máquina.
- Sistemas mais importantes desenvolvidos para representar símbolos com números binários (bits):
 - **EBCDIC** - Código Ampliado de Caracteres Decimais Codificados em Binário para o Intercâmbio de Dados.
 - **ASCII** - Código Padrão Americano para o Intercâmbio de Informações;
 - **UNICODE** - Unicódigo.

Sistemas de representação

- **EBCDIC**

- Código de 8 bits (256 símbolos).
- Usado em mainframe IBM e em sistemas de médio porte, raramente encontrado em microcomputadores.

- **ASCII**

- Padrão definido pela organização ANSI.
- Código de 7 bits (128 combinações de caracteres).
- No PC existe o ASCII Estendido (utiliza outros 128 códigos para símbolos gráficos, e línguas diferentes do inglês).

- **UNICODE**

- Novo padrão para representação de dados, oferecerá 2 bytes para a representação de símbolos (mais de 65.000 símbolos)

Palavra de dados

1 byte = 8 bits = 1 caractere (letra, número ou símbolo)

Palavra → Um conjunto de bits que representa uma informação útil para os computadores. A palavra nos computadores é um valor fixo e constante para um dado processador (ex.: 32 bits, 64 bits).

Conjunto de caracteres ASCII

Partes do conjunto de caracteres ASCII

| Caracteres | Binário | Hexadecimal |
|------------|-----------|-------------|
| A | 0100 0001 | 41 |
| B | 0100 0010 | 42 |
| a | 0110 0001 | 61 |
| b | 0100 0010 | 41 |
| i | 0011 1100 | 3C |
| = | 0011 1101 | 3D |
| ESC | 0001 1011 | 1B |
| DEL | 0111 1111 | 7F |

Como os principais códigos de representação de caracteres utilizam grupos de 8 bits por caractere, os conceitos `byte` e `caractere` tornam-se semelhantes, e as, palavras, quase sinônimas.

Unidades de medida de dados

Indicações numéricas dos computadores

| Byte | B | 8 bits | |
|------------|----|-------------|--------------------------------------|
| "Kilobyte" | KB | 1.024 bytes | $2^{10} = 1.024$ |
| Megabyte | MB | 1.024 KB | $2^{20} = 1.048.576$ |
| Gigabyte | GB | 1.024 MB | $2^{30} = 1.073.741.824$ |
| Terabyte | TB | 1.024 GB | $2^{40} = 1.099.511.627.776$ |
| Petabyte | PB | 1.024 TB | $2^{50} = 1.125.899.906.842.624$ |
| Exabyte | EB | 1.024 PB | $2^{60} = 1.152.921.504.606.846.976$ |

Os valores utilizados em computação para indicar capacidade de memória são normalmente compostos de um número (entre 0 e 999) e uma das abreviaturas citadas (ex.: 256KB, 64M, etc.)

Representação da informação

- Os computadores manipulam dados (**sinais brutos e sem significado individual**) para produzir informações.
- A conversão de dados em informações, e estas novamente em dados, é uma parte tão fundamental em relação ao que os computadores fazem que é preciso saber como a conversão ocorre para compreender como o computador funciona.
- Infelizmente os computadores não usam nosso sistema de numeração.

Embora os códigos de caracteres sejam úteis para representar dados textuais e números inteiros (0 a 9), eles não são úteis para números que possuem pontos fracionários, como 1,25. Para representar números com fração, bem como números extremamente grandes, por exemplo, os computadores utilizam a **notação de ponto flutuante** (a ser vista posteriormente).

Seção 2

Sistemas de Numeração

Sistema de numeração

Conjunto de símbolos utilizados para representação de quantidades e de regras que definem a forma de representação.

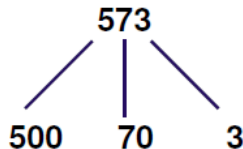
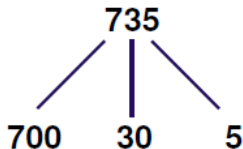
- Cada sistema de numeração é apenas um método diferente de representar **quantidades**. As quantidades em si não mudam; mudam apenas os **símbolos** usados para representá-las.
- A quantidade de algarismos disponíveis em um dado sistema de numeração é chamada de **base**.
- Representação numérica mais empregada: **notação posicional**.

Notação posicional

O valor atribuído a um símbolo **depende da posição** em que ele se encontra no conjunto de símbolos que representa uma quantidade.

- O valor total do número é a **soma** dos valores relativos de cada algarismo (decimal).

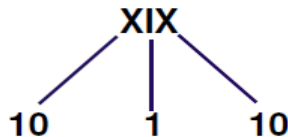
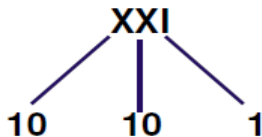
Sistema de numeração decimal



Notação não posicional

O valor atribuído a um símbolo é **inalterável**, independente da posição em que se encontre no conjunto de símbolos que representam uma quantidade.

Sistema de Numeração *Romano*



Exemplos de sistemas de numeração

- Binário
- Ternário
- Octal
- Decimal
- Duodecimal
- Hexadecimal
- ...

Algarismos dos sistemas de numeração

Exemplos de sistemas de numeração

| Sistema | Base | Algarismos |
|-------------|------|---------------------------------|
| Binário | 2 | 0,1 |
| Ternário | 3 | 0,1,2 |
| Octal | 8 | 0,1,2,3,4,5,6,7 |
| Decimal | 10 | 0,1,2,3,4,5,6,7,8,9 |
| Duodecimal | 12 | 0,1,2,3,4,5,6,7,8,9,A,B |
| Hexadecimal | 16 | 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F |

Como os números representados em **base 2** são muito extensos e, portanto, de **difícil manipulação** visual, costuma-se representar externamente os valores binários em **outras bases** de valor mais elevado (octal ou hexadecimal). Isso permite **maior compactação** de algoritmos e **melhor visualização** dos valores.

Padrões de representação

Como os símbolos usados nas diferentes representações são similares, temos que **explicitar em qual base** estamos apresentando o número.

- Letra após número para indicar a base;
- Número entre parênteses e a base como um índice do número.
- Exemplo: 2763**D** ou (2763)₁₀ ou 2763₁₀ (mais usado).

Sistema Decimal (Base 10)

- O sistema decimal é o sistema numérico mais utilizado.
- Possui 10 símbolos para representar quantidades.
- No sistema decimal cada algarismo tem um valor posicional, ou seja, cada algarismo tem um **peso** de acordo com a sua posição na representação do valor.
- Os pesos são: unidade, dezena, (dez unidades), centena (cem unidades), milhar (mil unidades), dezena de milhar, centena de milhar, etc.
- Ex.: 2574 é composto por 4 unidades, 7 dezenas, 5 centenas e 2 milhares, ou $2000 + 500 + 70 + 4 = 2574$.

Sistema Binário (Base 2)

- O sistema binário é o sistema numérico mais utilizado pelos **computadores**.
- Utiliza dois símbolos para representar quantidades, 0 e 1.
- Segue as regras do sistema decimal - válidos os conceitos de peso e posição. Posições não têm nome específico.
- Cada algarismo é chamado de **bit**. Exemplo: 101_2
- Expressão oral - diferente dos números decimais.
 - Caractere mais à esquerda - *Most-Significative-Bit*, o “MSB”.
 - Caractere mais à direita - *Least-Significative-Bit*, o “LSB”.

Outros sistemas

- O sistema **Octal** utiliza oito símbolos para representar quantidades = $\{0,1,2,3,4,5,6,7\}$.
- Exemplos = 563_8
- O sistema **Hexadecimal** utiliza dezesseis símbolos para representar quantidades = $\{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F\}$.
- O uso das letras facilita o manuseio.
- Exemplos = $5A3_{16}$

Sistemas numéricos

Ao trabalhar com sistemas de numeração, em qualquer base, deve-se observar o seguinte:

- O **número de dígitos** usado no sistema é igual à **base**.
- O maior dígito é sempre menor que a base.
- O dígito mais significativo está à esquerda, e o menos significativo à direita.
- Um “vai-um” de uma posição para outra tem um peso igual a uma potência da base, ou seja, avançar uma casa é incrementar o peso
- Em geral se toma a base decimal como referência.

Valores dos sistemas numéricos

| Decimal | Binário | Octal | Hexadecimal |
|---------|---------|-------|-------------|
| 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 |
| ⋮ | ⋮ | ⋮ | ⋮ |

Seção 3

Conversão entre Sistemas

Notação polinomial

Notação Polinomial ou posicional

- Válida para qualquer base numérica.
- Número:

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

- Onde a_n = algarismo, b = base do número e n é a quantidade de algarismos -1.

Exemplos de conversão entre sistemas numéricos

Exemplos:

$$\text{a) } (1111101)_2 = (?)_{10}$$

$$(1111101)_2 =$$

$$1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 125_{10}$$

$$\text{b) } (21A)_{16} = (?)_{10}$$

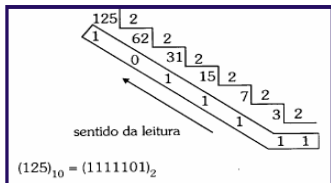
$$(21A)_{16} = 2 \times 16^2 + 1 \times 16^1 + 10 \times 16^0 = 538_{10}$$

Conversão por divisão pela base

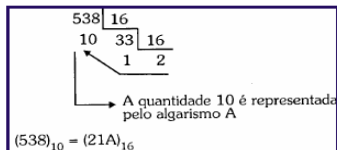
Decimal \Rightarrow Outro sistema

- Divisão inteira (do quociente) sucessiva pela base, até que resto seja menor do que a base.
- Valor na base = composição do último quociente (MSB) com restos (primeiro resto é bit menos significativo - LSB)

Ex.: $(125)_{10} = (?)_2$



$(538)_{10} = (?)_{16}$

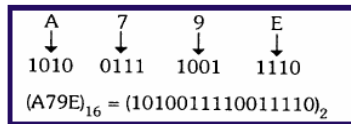
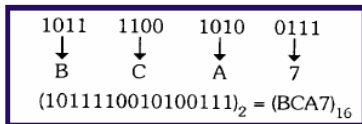


Conversão por agrupamento de bits

Sistemas octal e hexa \implies binário (e vice versa)

- Associando 3 bits ou 4 bits (quando octal ou hexadecimal, respectivamente) e vice-versa.
- Valor na base = composição do último quociente (MSB) com restos (primeiro resto é bit menos significativo - LSB)

Ex.: $(1011110010100111)_2 = (?)_{16}$ $(A79E)_{16} = (?)_2$



Conversão entre hexadecimal e octal

Sistemas octal \implies hexadecimal

- Não é realizada diretamente - não há relação de potências entre as bases oito e dezesseis.
- Semelhante à conversão entre duas bases quaisquer - base intermediária (base binária).
- Conversão em duas etapas:
 - 1 - Número: base octal (hexadecimal) \implies binária.
 - 2 - Resultado intermediário: binário \implies hexadecimal(octal).

Exemplos de conversão entre hexadecimal e octal

Exemplos:

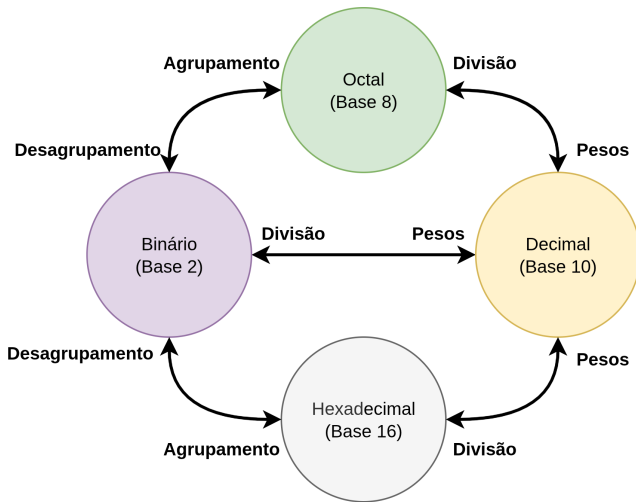
❶ $(175)_8 \Rightarrow (?)_{16}$

$$(175)_8 \Rightarrow (001)_2(111)_2(101)_2 \Rightarrow (1111101)_2 \Rightarrow (7D)_{16}$$

❷ $(21A)_{16} \Rightarrow (?)_8$

$$(21A)_{16} \Rightarrow (001000011010)_2 \Rightarrow (1032)_8$$

Resumo da conversão entre sistemas de numeração



Exercícios

- $(45)_{10} \Rightarrow$ binário
- $(301)_{10} \Rightarrow$ octal
- $(256)_{10} \Rightarrow$ hexadecimal
- $(563)_{16} \Rightarrow$ binário
- $(ADF)_{16} \Rightarrow$ decimal
- $(1D09)_{16} \Rightarrow$ octal
- $(1010010)_2 \Rightarrow$ octal
- $(100111)_2 \Rightarrow$ decimal
- $(1010101101)_2 \Rightarrow$ hexadecimal
- $(477)_8 \Rightarrow$ decimal
- $(1452)_8 \Rightarrow$ hexadecimal
- $(760)_8 \Rightarrow$ binário

Sistemas Numéricos



Universidade Federal do Ceará - Campus Quixadá

Pedro Botelho
pedro.botelho@ufc.br

18 de Setembro de 2025

Arquitetura de Computadores