20/05/2025

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO SUL
Campus Erechim

Curso Superior de Análise e Desenvolvimento de Sistemas - Prof: Andre Luiz Bedendo – Matemática Discreta

10. Sistemas de Numeração

Um sistema de numeração é um conjunto de regras e símbolos utilizados para representar quantidades ou informações de forma organizada. Cada sistema possui uma base numérica, que determina a quantidade de símbolos utilizados e o valor posicional de cada dígito. O sistema de numeração decimal é um sistema de base 10, composto por dez símbolos (0 a 9), onde o valor de cada dígito depende da sua posição na sequência numérica (valor posicional). É o sistema mais utilizado no dia a dia pelas pessoas, sendo aplicado em atividades cotidianas como contagem, operações matemáticas, preços de produtos, medições de tempo e distância, entre outros.

Na área da informática, os sistemas de numeração são fundamentais, pois os computadores utilizam o sistema binário (base 2) para processar e armazenar dados. Outros sistemas, como o octal (base 8) e o hexadecimal (base 16), também são amplamente utilizados para facilitar a leitura, interpretação e manipulação de dados binários em contextos como programação, endereçamento de memória e desenvolvimento de software. Compreender como funcionam esses sistemas é essencial para quem deseja atuar na área de tecnologia, pois eles são a base da comunicação entre o ser humano e as máquinas.

10.1 Sistema de Numeração Decimal (Base 10)

É o sistema que usamos naturalmente no cotidiano. Ele possui **dez algarismos**: **0, 1, 2, 3, 4, 5, 6, 7, 8 e 9**. Cada posição de um número representa uma potência de 10.

Exemplos:

$58 \rightarrow 5 \cdot 10^1 + 8 \cdot 10^0$

$$258{,}37 \rightarrow \underbrace{\underbrace{2 \cdot 10^2}_{200} + \underbrace{5 \cdot 10^1}_{50} + \underbrace{8 \cdot 10^0}_{8}}_{258} \;, \underbrace{+ \underbrace{3 \cdot 10^{-1}}_{0{,}3} + \underbrace{3 \cdot 10^{-2}}_{0{,}07}}_{37}$$

10.2 Sistema de Numeração Binário (Base 2)

O sistema de numeração binário é um sistema de base 2 que utiliza apenas dois dígitos: 0 e 1. A palavra **bit** vem de **"binary digit"**, ou seja, **dígito binário**. Por isso, o bit é a **menor unidade de informação** que um computador pode armazenar ou processar. Eles são fundamentais na computação, pois é o sistema usado internamente pelos computadores para representar e processar todas as informações. No dia a dia, o sistema binário está presente em praticamente tudo que envolve tecnologia digital — como no funcionamento de celulares, computadores, cartões de crédito, aparelhos eletrônicos e até sinais transmitidos por redes Wi-Fi. Cada bit representa **um único dígito** no sistema binário. Quando colocamos **vários bits juntos**, conseguimos representar números, letras, imagens e até vídeos. Ou seja, **os números binários são formados por sequências de bits.**

Exemplos: 10111_2 $\quad 10{,}1_2 \quad 1111110{,}10101010_2 \quad 0{,}1101010101_2$

↳ numero é binário

OBS: Como no caso dos números decimais, a posição relativa de cada algarismo binário indica a potência pela qual ele está multiplicado. Igualmente os coeficientes que multiplicam potências de 2 cujos expoentes sejam < 0 estão separados por uma vírgula daqueles coeficientes cujos expoentes de 2 sejam ≥ 0.

A **tabela de estrutura das unidades de armazenamento** é construída com base na forma como os computadores manipulam dados usando o **sistema binário**, que trabalha com potências de 2. Ela começa com o **bit** (a menor unidade de informação, representando 0 ou 1) e evolui em múltiplos de 1.024 (2^{10}) para formar unidades maiores.

Unidade	Equivalência em Bytes	Potência de 2	Explicação Prática
Bit (b)		$2^1 = 2$ bits	Menor unidade de informação: 0 ou 1
Byte (B)	8 bits	$2^3 = 8$ bits	Representa um caractere, como uma letra
Kilobyte (KB)	1.024 bytes	$2^{10} = 1.024$	Cerca de 1 mil caracteres de texto
Megabyte (MB)	1.024 KB = 1.048.576 bytes	$2^{20} = 1.048.576$	Uma música em MP3 compactada
Gigabyte (GB)	1.024 MB	$2^{30} = 1.073.741.824$	Um filme em qualidade padrão
Terabyte (TB)	1.024 GB	$2^{40} = 1.099.511.627.776$	Milhares de fotos ou vídeos
Petabyte (PB)	1.024 TB	$2^{50} = 1.125.899.906.842.624$	Usado em grandes servidores e datacenters
Exabyte (EB)	1.024 PB	$2^{60} = 1.152.921.504.606.846.976$	Volume de dados de grandes empresas como Google
Zettabyte (ZB)	1.024 EB	$2^{70} = 1.180.591.620.717.411.303.424$	Medida usada para tráfego total da internet em um ano
Yottabyte (YB)	1.024 ZB	$2^{80} = 1.208.925.819.614.629.174.706.176$	Armazenamento em escala planetária ou governamental

Quantos valores diferentes posso representar com bits?

Existe uma fórmula simples: Com n **bits,** é possível representar até 2^n **valores diferentes.**

Bits	Valores possíveis	Combinações possíveis
1	2	0, 1
2	4	00, 01, 10, 11
3	8	000, 001, 010, 011, 100, 101, 110, 111
4	16	0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111,1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111
8	256	Exemplos: 00000000, 00000001, 00001111, 11111111, 10101010, 11001100, ... *(e muitas outras combinações possíveis)*

Interpretações práticas

Para 1 bit: Como **1 bit só pode assumir dois valores possíveis: 0 ou 1**, isso significa que ele pode **representar duas situações diferentes**.

Bit	Interpretação possível
0	Desligado, Falso, Não, Sem sinal, Ausência
1	Ligado, Verdadeiro, Sim, Com sinal, Presença

Exemplos práticos:

Situação	0	1
Interruptor	Desligado	Ligado
Resposta lógica	Falso	Verdadeiro
Condição binária	Não	Sim
Comunicação digital	Sem sinal	Com sinal
Detecção de movimento	Ausência	Presença

Para 2 bits: podemos representar 4 estados diferentes.

Aplicação	00	01	10	11
Semáforo simples	Verde	Amarelo	Vermelho	Desligado
Nível de volume	Mudo	Baixo	Médio	Alto
Emojis de humor	😐	🙂	😄	😂
Controle de acesso	Nenhum	Leitura	Escrita	Total

Com apenas **2 bits**, já temos o **dobro de possibilidades** em relação a 1 bit. Isso mostra o poder do sistema binário: **aumentar 1 bit dobra o número de combinações possíveis**.

10.3 Conversão de Números

Compreender como converter e interpretar valores entre decimal e binário **é o primeiro passo** para entender o funcionamento de computadores, sistemas digitais, linguagens de programação e até de redes e segurança da informação. Essa habilidade é uma base fundamental para qualquer estudante de computação, engenharia ou tecnologia.

Binário para Decimal: para convertermos um número binário para um decimal basta representá-lo no seu equivalente na base decimal.

$$(10111)_2 = \boxed{1|0|1|1|1} \rightarrow 1\cdot 2^4 + 0\cdot 2^3 + 1\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0 \rightarrow 16 + 0 + 4 + 2 + 1 = 23$$

$$(10{,}1101)_2 = 1\cdot 2^1 + 0\cdot 2^0 + 1\cdot 2^{-1} + 1\cdot 2^{-2} + 0\cdot 2^{-3} + 1\cdot 2^{-4} = 2 + 0 + 0{,}5 + 0{,}25 + 0 + 0{,}0625 = \boxed{2{,}8125}$$

$$(10100{,}10)_2 = 1\cdot 2^4 + 0\cdot 2^3 + 1\cdot 2^2 + 0\cdot 2^1 + 0\cdot 2^0 + 1\cdot 2^{-1} + 0\cdot 2^{-2} = 16 + 0 + 4 + 0 + 0 + 0{,}5 + 0 = \boxed{20{,}5}$$

Decimal para Binário: para convertermos um número decimal para um número binário devemos aplicar um método para a parte inteira e outro para a parte fracionária. Para a parte inteira, aplicamos o método das divisões sucessivas, que consiste em dividir o número decimal por 2, em seguida divide-se o quociente obtido por 2 e assim sucessivamente, até que o último quociente encontrado seja 1. O número binário inteiro será, então, formado pela concatenação do último quociente com os restos das divisões no sentido contrário ao que foram obtidos.

DIVIDENDO: 50 | DIVISOR: 5
-50 | QUOCIENTE: 10
RESTO: 0

Exemplo 1: Converter o número decimal 23:

23 | 2
-22 → 11 | 2
1 → -10 → 5 | 2
1 → -4 → 2 | 2
1 → -2 → 1
0

10111_2

Para converter um número fracionário de base decimal para base binária aplicamos o método das multiplicações sucessivas. Ele consiste nas seguintes etapas: multiplicamos o número fracionário por 2; deste resultado, a parte inteira será o primeiro digito do número fracionário binário e a parte fracionária será novamente multiplicada por 2. O processo repete-se até que a parte fracionária do último produto seja zero.

Exemplo 2: Converter o número decimal 0,1875

$0{,}1875 \times 2 = \boxed{0}{,}3750$ → $,3750 \times 2 = \boxed{0}{,}7500$ → $,75 \times 2 = \boxed{1}{,}50$ → $,50 \times 2 = \boxed{1}{,}00$

$\therefore\ 0{,}1875 = 0{,}0011_2$

A conversão de bases para um número decimal que contém parcela inteira e fracionária é constituída por duas partes distintas sobre as quais são aplicadas as respectivas regras de conversão.

Exemplo 3: Converter o número decimal 31,625

PARTE INTEIRA

31 | 2
-30 15 | 2
1 -14 7 | 2
1 -6 3 | 2
1 -2 1
1

→ 11111

PARTE DECIMAL

,625 × 2 = 1,250
,250 × 2 = 0,500
,50 × 2 = 1,00

→ ,101

JUNÇÃO

$11111,101_2$

📌 **Observações:**

- Nem toda fração decimal tem uma representação exata em binário (assim como 1/3 não é exato em decimal).
- A conversão é infinita e periódica (não termina). Ou seja, **não tem representação exata em binário finito**, o resultado será sempre uma **aproximação periódica**. Você pode parar o cálculo da parte fracionária com 3, 4 ou mais casas binárias, dependendo da precisão desejada. Ou ainda, representar esse número como uma sequência infinita e periódica.
- Para representar um número binário com **parte fracionária periódica**, usamos uma notação semelhante à que usamos em números decimais: **três pontinhos ("...")** ou **colocamos a parte repetida entre parênteses**.

$\frac{1}{3} \approx 0.010101\ldots_2$ ou $\frac{1}{3} = 0.(01)_2$

$150{,}35 \approx 10010110.01011001\ldots_2$ ou $150{,}35 \approx 10010110.(01011001)_2$

Exemplo 3: Converter o número $\frac{1}{3}$ para representação binária.

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO SUL
Campus Erechim

Lista 10 – Sistemas de Numeração – Decimal e Binário

1. Um sistema embarcado em uma empresa de automação precisa monitorar o estado de quatro sensores binários instalados em uma sala de controle. Cada sensor pode estar em apenas dois estados:

- 0 → Desligado
- 1 → Ligado

Esses sensores estão conectados a um microcontrolador que lê os quatro estados simultaneamente, formando assim uma combinação binária de 4 bits. Determine:

a) Quantas combinações diferentes de estados é possível representar com esses 4 sensores?

b) Pela combinação 1101, quais sensores estão ligados e quais estão desligados?

c) Um técnico quer acionar um alarme sempre que dois ou mais sensores estiverem ligados. Quantas são as combinações possíveis de 4 bits que devem acionar o alarme.

2. Interpretando Estruturas de Controle com Bits. Um sistema de controle de acesso utiliza um byte (8 bits) para representar o status de permissões de um usuário. Cada bit representa uma permissão específica:

Bit	Permissão
7	Acesso ao servidor remoto
6	Leitura de relatórios
5	Escrita em relatórios
4	Acesso à configuração
3	Exclusão de arquivos
2	Upload de arquivos
1	Download de arquivos
0	Login autorizado

Dado o byte de permissões 10110101, responda:

a) Quais permissões o usuário possui?

b) Qual valor decimal correspondente a esse byte?

c) Qual seria o novo valor binário e decimal se o usuário perdesse a permissão de exclusão de arquivos?

3. Manipulação de Bits com Operações Lógicas. Um determinado sistema representa o estado de uma máquina com 4 bits, sendo:

Bit	Componente
3	Motor ligado
2	Sensor de temperatura
1	Sensor de pressão
0	Emergência ativada

Dado o estado atual da máquina representado pelo número decimal 9, responda:

a) Qual a representação binária deste valor e o estado de cada componente?

b) Usando operações lógicas, qual expressão binária pode ser usada para ativar o sensor de pressão, sem alterar os demais bits?

c) Após essa ativação, qual será o novo valor binário e decimal?

4. Entendendo as Unidades de Armazenamento. Um desenvolvedor está construindo um aplicativo para armazenar registros de sensores ambientais. Cada registro ocupa exatamente 64 bytes de espaço. O sistema deve armazenar dados contínuos por 24 horas, capturando um registro a cada 5 segundos. Utilizando a tabela de armazenamento de dados, determine:

a) Quantos registros serão armazenados em 24 horas?

b) Qual o espaço total ocupado em bytes, kilobytes e megabytes?

c) Quantos registros esse sistema conseguiria armazenar em 1 GB de espaço disponível?

5. Otimização de Armazenamento em um Aplicativo Web. Um desenvolvedor está criando um aplicativo de monitoramento que registra o estado de 10 sensores digitais (ligado/desligado). Ele decide usar apenas 1 bit para representar cada sensor (0 para desligado, 1 para ligado), armazenando o estado de todos os sensores a cada minuto. O aplicativo roda continuamente durante 15 dias, e os dados são armazenados localmente antes de serem enviados para a nuvem. Determine:

a) Quantos bits são necessários para armazenar 1 leitura completa (dos 10 sensores)?

b) Quantos bits serão armazenados em um dia? E em 15 dias?

c) Qual será o consumo total em bytes, kilobytes e megabytes após os 15 dias?

d) Considerando que o sistema armazena leituras continuamente, quantos dias levaria para atingir 1 Terabyte (TB) de dados armazenados, mantendo o mesmo padrão leitura a cada minuto?

6. Realize a conversão dos números a seguir para a base binária.

a) 55 b) 156 c) 72 d) 1235 e) 22,5 f) 150,35 g) 0,75

h) 121,375 i) $\frac{2}{3}$ j) 212,15 k) 129,25 l) 585,10 m) 1010 o) 0,175

7. Realize a conversão dos número a seguir para a base 10.

a) 100111 b) 10011001 c) 100001 d) 1010110 e) 1101011

f) 10111110.0101 g) 11.0101001 h) 111.00011 i) 10000001.01 f) 1010.110

Respostas: Número 1: a) 16 b) Sensores: 1, 2 e 4 Ligados, e 3 Desligado c) 11

Número 2: a) Servidor remoto, Escrita em relatórios, Acesso à configuração, Upload de arquivos, Login autorizado b) 181 c) 181 Número 3: a) 1001 (Ligado, Desligado, Desligado, Ligado) b) 0010→ 1011 c) 11

Número 4: a) 17280 b) 1.105.920 bytes, $\approx 1.080\ KB\ e\ \approx 1.05\ MB$ c) 16.277.216

Número 5: a) 10 bits b) 14400 e 216000 bits c) 27000 bytes $\approx 26.37\ KB\ e\ \approx 0.0285\ MB$

d) $\approx 610.839.793\ dias$ Número 6: a) 110111 b) 10011100 c) 1001000

d) 10011010011 e) 10110.1 f) 10010110.010110(0110) g) 0.11 h) 1111001.011

i) 0.10(10) j) 11010100.001(0011) k) 10000001.01 l) 1001001001.0(0011)

m) 1111110010 n) 0.001(01010) Número 7: a) 39 b) 153 c) 33

d) 86 e) 107 f) 190,3125 g) 3,3203125 h) 7,09375 i) 129,25 j) 10,75