

# Conversão entre Diferentes Bases Numéricas

Prof. Americo Cunha

Universidade do Estado do Rio de Janeiro – UERJ

[americo.cunha@uerj.br](mailto:americo.cunha@uerj.br)

[www.americocunha.org](http://www.americocunha.org)



# Como converter um natural de decimal para binário?

Para converter um natural  $N$ , escrito em base decimal, para a base binária utilizamos os restos de sucessivas divisões por 2.

$$N = q \times d + r$$

- $N$  – dividendo
- $d$  – divisor
- $q$  – quociente
- $r$  – resto

Qual a representação binária de 156?



# Como converter um natural de decimal para binário?

Para converter um natural  $N$ , escrito em base decimal, para a base binária utilizamos os restos de sucessivas divisões por 2.

$$N = q \times d + r$$

- $N$  – dividendo
- $d$  – divisor
- $q$  – quociente
- $r$  – resto

Qual a representação binária de 156?

- $156 = 2 \times 78 + 0$



## Como converter um natural de decimal para binário?

Para converter um natural  $N$ , escrito em base decimal, para a base binária utilizamos os restos de sucessivas divisões por 2.

$$N = q \times d + r$$

- $N$  – dividendo
- $d$  – divisor
- $q$  – quociente
- $r$  – resto

Qual a representação binária de 156?

- $156 = 2 \times 78 + 0$
- $78 = 2 \times 39 + 0$



# Como converter um natural de decimal para binário?

Para converter um natural  $N$ , escrito em base decimal, para a base binária utilizamos os restos de sucessivas divisões por 2.

$$N = q \times d + r$$

- $N$  – dividendo
- $d$  – divisor
- $q$  – quociente
- $r$  – resto

Qual a representação binária de 156?

- $156 = 2 \times 78 + 0$
- $78 = 2 \times 39 + 0$
- $39 = 2 \times 19 + 1$



# Como converter um natural de decimal para binário?

Para converter um natural  $N$ , escrito em base decimal, para a base binária utilizamos os restos de sucessivas divisões por 2.

$$N = q \times d + r$$

- $N$  – dividendo
- $d$  – divisor
- $q$  – quociente
- $r$  – resto

Qual a representação binária de 156?

- $156 = 2 \times 78 + 0$
- $78 = 2 \times 39 + 0$
- $39 = 2 \times 19 + 1$
- $19 = 2 \times 9 + 1$



# Como converter um natural de decimal para binário?

Para converter um natural  $N$ , escrito em base decimal, para a base binária utilizamos os restos de sucessivas divisões por 2.

$$N = q \times d + r$$

- $N$  – dividendo
- $d$  – divisor
- $q$  – quociente
- $r$  – resto

Qual a representação binária de 156?

- $156 = 2 \times 78 + 0$
- $78 = 2 \times 39 + 0$
- $39 = 2 \times 19 + 1$
- $19 = 2 \times 9 + 1$
- $9 = 2 \times 4 + 1$



# Como converter um natural de decimal para binário?

Para converter um natural  $N$ , escrito em base decimal, para a base binária utilizamos os restos de sucessivas divisões por 2.

$$N = q \times d + r$$

- $N$  – dividendo
- $d$  – divisor
- $q$  – quociente
- $r$  – resto

Qual a representação binária de 156?

- $156 = 2 \times 78 + 0$
- $78 = 2 \times 39 + 0$
- $39 = 2 \times 19 + 1$
- $19 = 2 \times 9 + 1$
- $9 = 2 \times 4 + 1$
- $4 = 2 \times 2 + 0$





# Como converter um natural de decimal para binário?

Para converter um natural  $N$ , escrito em base decimal, para a base binária utilizamos os restos de sucessivas divisões por 2.

$$N = q \times d + r$$

- $N$  – dividendo
- $d$  – divisor
- $q$  – quociente
- $r$  – resto

Qual a representação binária de 156?

- $156 = 2 \times 78 + 0$
- $78 = 2 \times 39 + 0$
- $39 = 2 \times 19 + 1$
- $19 = 2 \times 9 + 1$
- $9 = 2 \times 4 + 1$
- $4 = 2 \times 2 + 0$
- $2 = 2 \times 1 + 0$



# Como converter um natural de decimal para binário?

Para converter um natural  $N$ , escrito em base decimal, para a base binária utilizamos os restos de sucessivas divisões por 2.

$$N = q \times d + r$$

- $N$  – dividendo
- $d$  – divisor
- $q$  – quociente
- $r$  – resto

Qual a representação binária de 156?

- $156 = 2 \times 78 + 0$
- $78 = 2 \times 39 + 0$
- $39 = 2 \times 19 + 1$
- $19 = 2 \times 9 + 1$
- $9 = 2 \times 4 + 1$
- $4 = 2 \times 2 + 0$
- $2 = 2 \times 1 + 0$
- $1 = 2 \times 0 + 1$



# Como converter um natural de decimal para binário?

Para converter um natural  $N$ , escrito em base decimal, para a base binária utilizamos os restos de sucessivas divisões por 2.

$$N = q \times d + r$$

- $N$  – dividendo
- $d$  – divisor
- $q$  – quociente
- $r$  – resto

Qual a representação binária de 156?

- $156 = 2 \times 78 + 0$
- $78 = 2 \times 39 + 0$
- $39 = 2 \times 19 + 1$
- $19 = 2 \times 9 + 1$
- $9 = 2 \times 4 + 1$
- $4 = 2 \times 2 + 0$
- $2 = 2 \times 1 + 0$
- $1 = 2 \times 0 + 1$
- $0 = 2 \times 0 + 0$



## Como converter um natural de decimal para binário?

Para converter um natural  $N$ , escrito em base decimal, para a base binária utilizamos os restos de sucessivas divisões por 2.

$$N = q \times d + r$$

- $N$  – dividendo
- $d$  – divisor
- $q$  – quociente
- $r$  – resto

Qual a representação binária de 156?

- $156 = 2 \times 78 + 0$
- $78 = 2 \times 39 + 0$
- $39 = 2 \times 19 + 1$
- $19 = 2 \times 9 + 1$
- $9 = 2 \times 4 + 1$
- $4 = 2 \times 2 + 0$
- $2 = 2 \times 1 + 0$
- $1 = 2 \times 0 + 1$
- $0 = 2 \times 0 + 0$

$$(156)_{10} = (10011100)_2$$



# Por que o algoritmo funciona?



## Por que o algoritmo funciona?

Seja  $N_0$  um inteiro positivo cuja representação binária possui  $k + 1$  dígitos, i.e.,  $N_0 = (b_k b_{k-1} \cdots b_1 b_0)_2$



## Por que o algoritmo funciona?

Seja  $N_0$  um inteiro positivo cuja representação binária possui  $k + 1$  dígitos, i.e.,  $N_0 = (b_k b_{k-1} \cdots b_1 b_0)_2$

$$N_0 = b_k \times 2^k + b_{k-1} \times 2^{k-1} + \cdots + b_1 \times 2^1 + b_0$$



## Por que o algoritmo funciona?

Seja  $N_0$  um inteiro positivo cuja representação binária possui  $k + 1$  dígitos, i.e.,  $N_0 = (b_k b_{k-1} \cdots b_1 b_0)_2$

$$\begin{aligned} N_0 &= b_k \times 2^k + b_{k-1} \times 2^{k-1} + \cdots + b_1 \times 2^1 + b_0 \\ &= 2 \times \left( b_k \times 2^{k-1} + b_{k-1} \times 2^{k-2} + \cdots + b_2 \times 2^1 + b_1 \right) + b_0 \end{aligned}$$





## Por que o algoritmo funciona?

Seja  $N_0$  um inteiro positivo cuja representação binária possui  $k + 1$  dígitos, i.e.,  $N_0 = (b_k b_{k-1} \cdots b_1 b_0)_2$

$$\begin{aligned} N_0 &= b_k \times 2^k + b_{k-1} \times 2^{k-1} + \cdots + b_1 \times 2^1 + b_0 \\ &= 2 \times \underbrace{\left( b_k \times 2^{k-1} + b_{k-1} \times 2^{k-2} + \cdots + b_2 \times 2^1 + b_1 \right)}_{N_1} + b_0 \end{aligned}$$



## Por que o algoritmo funciona?

Seja  $N_0$  um inteiro positivo cuja representação binária possui  $k + 1$  dígitos, i.e.,  $N_0 = (b_k b_{k-1} \cdots b_1 b_0)_2$

$$\begin{aligned} N_0 &= b_k \times 2^k + b_{k-1} \times 2^{k-1} + \cdots + b_1 \times 2^1 + b_0 \\ &= 2 \times \underbrace{\left( b_k \times 2^{k-1} + b_{k-1} \times 2^{k-2} + \cdots + b_2 \times 2^1 + b_1 \right)}_{N_1} + b_0 \end{aligned}$$

$$N_1 = 2 \times \left( b_k \times 2^{k-2} + b_{k-1} \times 2^{k-3} + \cdots + b_3 \times 2^1 + b_2 \right) + b_1$$



## Por que o algoritmo funciona?

Seja  $N_0$  um inteiro positivo cuja representação binária possui  $k + 1$  dígitos, i.e.,  $N_0 = (b_k b_{k-1} \cdots b_1 b_0)_2$

$$\begin{aligned} N_0 &= b_k \times 2^k + b_{k-1} \times 2^{k-1} + \cdots + b_1 \times 2^1 + b_0 \\ &= 2 \times \underbrace{\left( b_k \times 2^{k-1} + b_{k-1} \times 2^{k-2} + \cdots + b_2 \times 2^1 + b_1 \right)}_{N_1} + b_0 \end{aligned}$$

$$N_1 = 2 \times \underbrace{\left( b_k \times 2^{k-2} + b_{k-1} \times 2^{k-3} + \cdots + b_3 \times 2^1 + b_2 \right)}_{N_2} + b_1$$



## Por que o algoritmo funciona?

Seja  $N_0$  um inteiro positivo cuja representação binária possui  $k + 1$  dígitos, i.e.,  $N_0 = (b_k b_{k-1} \cdots b_1 b_0)_2$

$$\begin{aligned} N_0 &= b_k \times 2^k + b_{k-1} \times 2^{k-1} + \cdots + b_1 \times 2^1 + b_0 \\ &= 2 \times \underbrace{\left( b_k \times 2^{k-1} + b_{k-1} \times 2^{k-2} + \cdots + b_2 \times 2^1 + b_1 \right)}_{N_1} + b_0 \end{aligned}$$

$$N_1 = 2 \times \underbrace{\left( b_k \times 2^{k-2} + b_{k-1} \times 2^{k-3} + \cdots + b_3 \times 2^1 + b_2 \right)}_{N_2} + b_1$$

$\vdots$



## Por que o algoritmo funciona?

Seja  $N_0$  um inteiro positivo cuja representação binária possui  $k + 1$  dígitos, i.e.,  $N_0 = (b_k b_{k-1} \cdots b_1 b_0)_2$

$$\begin{aligned} N_0 &= b_k \times 2^k + b_{k-1} \times 2^{k-1} + \cdots + b_1 \times 2^1 + b_0 \\ &= 2 \times \underbrace{\left( b_k \times 2^{k-1} + b_{k-1} \times 2^{k-2} + \cdots + b_2 \times 2^1 + b_1 \right)}_{N_1} + b_0 \end{aligned}$$

$$N_1 = 2 \times \underbrace{\left( b_k \times 2^{k-2} + b_{k-1} \times 2^{k-3} + \cdots + b_3 \times 2^1 + b_2 \right)}_{N_2} + b_1$$

$\vdots$

$$N_{k-1} = 2 \times b_k + b_{k-1}$$



## Por que o algoritmo funciona?

Seja  $N_0$  um inteiro positivo cuja representação binária possui  $k + 1$  dígitos, i.e.,  $N_0 = (b_k b_{k-1} \cdots b_1 b_0)_2$

$$\begin{aligned} N_0 &= b_k \times 2^k + b_{k-1} \times 2^{k-1} + \cdots + b_1 \times 2^1 + b_0 \\ &= 2 \times \underbrace{\left( b_k \times 2^{k-1} + b_{k-1} \times 2^{k-2} + \cdots + b_2 \times 2^1 + b_1 \right)}_{N_1} + b_0 \end{aligned}$$

$$N_1 = 2 \times \underbrace{\left( b_k \times 2^{k-2} + b_{k-1} \times 2^{k-3} + \cdots + b_3 \times 2^1 + b_2 \right)}_{N_2} + b_1$$

$\vdots$

$$N_{k-1} = 2 \times \underbrace{b_k}_{N_k} + b_{k-1}$$



## Por que o algoritmo funciona?

Seja  $N_0$  um inteiro positivo cuja representação binária possui  $k + 1$  dígitos, i.e.,  $N_0 = (b_k b_{k-1} \cdots b_1 b_0)_2$

$$\begin{aligned} N_0 &= b_k \times 2^k + b_{k-1} \times 2^{k-1} + \cdots + b_1 \times 2^1 + b_0 \\ &= 2 \times \underbrace{\left( b_k \times 2^{k-1} + b_{k-1} \times 2^{k-2} + \cdots + b_2 \times 2^1 + b_1 \right)}_{N_1} + b_0 \end{aligned}$$

$$N_1 = 2 \times \underbrace{\left( b_k \times 2^{k-2} + b_{k-1} \times 2^{k-3} + \cdots + b_3 \times 2^1 + b_2 \right)}_{N_2} + b_1$$

$\vdots$

$$N_{k-1} = 2 \times \underbrace{b_k}_{N_k} + b_{k-1}$$

$$N_k = b_k$$



## Por que o algoritmo funciona?

Seja  $N_0$  um inteiro positivo cuja representação binária possui  $k + 1$  dígitos, i.e.,  $N_0 = (b_k b_{k-1} \cdots b_1 b_0)_2$

$$\begin{aligned} N_0 &= b_k \times 2^k + b_{k-1} \times 2^{k-1} + \cdots + b_1 \times 2^1 + b_0 \\ &= 2 \times \underbrace{\left( b_k \times 2^{k-1} + b_{k-1} \times 2^{k-2} + \cdots + b_2 \times 2^1 + b_1 \right)}_{N_1} + b_0 \end{aligned}$$

$$N_1 = 2 \times \underbrace{\left( b_k \times 2^{k-2} + b_{k-1} \times 2^{k-3} + \cdots + b_3 \times 2^1 + b_2 \right)}_{N_2} + b_1$$

$\vdots$

$$N_{k-1} = 2 \times \underbrace{b_k}_{N_k} + b_{k-1}$$

$$N_k = b_k$$

$$N_j = 2 \times N_{j+1} + b_j \longrightarrow b_j \text{ é o resto da divisão de } N_j \text{ por } 2$$





E se tivermos um número fracionário?

0,625



E se tivermos um número fracionário?

$$0,625 = 5/8 = 1/2 + 1/8 = 2^{-1} + 2^{-3}$$



E se tivermos um número fracionário?

$$0,625 = 5/8 = 1/2 + 1/8 = 2^{-1} + 2^{-3}$$

$$0,625 = 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$$



E se tivermos um número fracionário?

$$0,625 = 5/8 = 1/2 + 1/8 = 2^{-1} + 2^{-3}$$

$$0,625 = 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$$



E se tivermos um número fracionário?

$$0,625 = 5/8 = 1/2 + 1/8 = 2^{-1} + 2^{-3}$$

$$0,625 = 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$$

$$(0,625)_{10} = (0,101)_2$$



E se tivermos um número fracionário?

$$0,625 = 5/8 = 1/2 + 1/8 = 2^{-1} + 2^{-3}$$

$$0,625 = 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$$

$$(0,625)_{10} = (0,101)_2$$

⇒ Representação finita



E se tivermos um número fracionário?

0,1



E se tivermos um número fracionário?

$$0,1 = b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + b_{-3} \times 2^{-3} + \dots$$





E se tivermos um número fracionário?

$$0,1 = b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + b_{-3} \times 2^{-3} + \dots$$

Multiplicando por 2:

## E se tivermos um número fracionário?

$$0,1 = b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + b_{-3} \times 2^{-3} + \dots$$

Multiplicando por 2:

$$0,2 = \underbrace{b_{-1}}_{\text{inteiro}} + \underbrace{b_{-2} \times 2^{-1} + b_{-3} \times 2^{-2} + \dots}_{\text{fracionário}}$$



## E se tivermos um número fracionário?

$$0,1 = b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + b_{-3} \times 2^{-3} + \dots$$

Multiplicando por 2:

$$0,2 = \underbrace{b_{-1}}_{\text{inteiro}} + \underbrace{b_{-2} \times 2^{-1} + b_{-3} \times 2^{-2} + \dots}_{\text{fracionário}}$$
$$\Rightarrow b_{-1} = 0$$



## E se tivermos um número fracionário?

$$0,1 = b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + b_{-3} \times 2^{-3} + \dots$$

Multiplicando por 2:

$$0,2 = \underbrace{b_{-1}}_{\text{inteiro}} + \underbrace{b_{-2} \times 2^{-1} + b_{-3} \times 2^{-2} + \dots}_{\text{fracionário}}$$

$$\Rightarrow b_{-1} = 0$$

$$0,4 = \underbrace{b_{-2}}_{\text{inteiro}} + \underbrace{b_{-3} \times 2^{-1} + b_{-4} \times 2^{-2} + \dots}_{\text{fracionário}}$$



## E se tivermos um número fracionário?

$$0,1 = b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + b_{-3} \times 2^{-3} + \dots$$

Multiplicando por 2:

$$0,2 = \underbrace{b_{-1}}_{\text{inteiro}} + \underbrace{b_{-2} \times 2^{-1} + b_{-3} \times 2^{-2} + \dots}_{\text{fracionário}}$$

$$\Rightarrow b_{-1} = 0$$

$$0,4 = \underbrace{b_{-2}}_{\text{inteiro}} + \underbrace{b_{-3} \times 2^{-1} + b_{-4} \times 2^{-2} + \dots}_{\text{fracionário}}$$

$$\Rightarrow b_{-2} = 0$$



E se tivermos um número fracionário?

$$0,1 = b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + b_{-3} \times 2^{-3} + \dots$$

Multiplicando por 2:

## E se tivermos um número fracionário?

$$0,1 = b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + b_{-3} \times 2^{-3} + \dots$$

Multiplicando por 2:

$$0,8 = \underbrace{b_{-3}}_{\text{inteiro}} + \underbrace{b_{-4} \times 2^{-1} + b_{-5} \times 2^{-2} + \dots}_{\text{fracionário}}$$



E se tivermos um número fracionário?

$$0,1 = b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + b_{-3} \times 2^{-3} + \dots$$

Multiplicando por 2:

$$0,8 = \underbrace{b_{-3}}_{\text{inteiro}} + \underbrace{b_{-4} \times 2^{-1} + b_{-5} \times 2^{-2} + \dots}_{\text{fracionário}}$$
$$\Rightarrow b_{-3} = 0$$





## E se tivermos um número fracionário?

$$0,1 = b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + b_{-3} \times 2^{-3} + \dots$$

Multiplicando por 2:

$$0,8 = \underbrace{b_{-3}}_{\text{inteiro}} + \underbrace{b_{-4} \times 2^{-1} + b_{-5} \times 2^{-2} + \dots}_{\text{fracionário}}$$

$$\Rightarrow b_{-3} = 0$$

$$1 + 0,6 = \underbrace{b_{-4}}_{\text{inteiro}} + \underbrace{b_{-5} \times 2^{-1} + b_{-6} \times 2^{-2} + \dots}_{\text{fracionário}}$$



## E se tivermos um número fracionário?

$$0,1 = b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + b_{-3} \times 2^{-3} + \dots$$

Multiplicando por 2:

$$0,8 = \underbrace{b_{-3}}_{\text{inteiro}} + \underbrace{b_{-4} \times 2^{-1} + b_{-5} \times 2^{-2} + \dots}_{\text{fracionário}}$$

$$\Rightarrow b_{-3} = 0$$

$$1 + 0,6 = \underbrace{b_{-4}}_{\text{inteiro}} + \underbrace{b_{-5} \times 2^{-1} + b_{-6} \times 2^{-2} + \dots}_{\text{fracionário}}$$

$$\Rightarrow b_{-4} = 1$$



E se tivermos um número fracionário?

$$0,1 = b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + b_{-3} \times 2^{-3} + \dots$$

Multiplicando por 2:

## E se tivermos um número fracionário?

$$0,1 = b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + b_{-3} \times 2^{-3} + \dots$$

Multiplicando por 2:

$$1 + 0,2 = \underbrace{b_{-5}}_{\text{inteiro}} + \underbrace{b_{-6} \times 2^{-1} + b_{-7} \times 2^{-2} + \dots}_{\text{fracionário}}$$



E se tivermos um número fracionário?

$$0,1 = b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + b_{-3} \times 2^{-3} + \dots$$

Multiplicando por 2:

$$1 + 0,2 = \underbrace{b_{-5}}_{\text{inteiro}} + \underbrace{b_{-6} \times 2^{-1} + b_{-7} \times 2^{-2} + \dots}_{\text{fracionário}}$$
$$\Rightarrow b_{-5} = 1$$



## E se tivermos um número fracionário?

$$0,1 = b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + b_{-3} \times 2^{-3} + \dots$$

Multiplicando por 2:

$$1 + 0,2 = \underbrace{b_{-5}}_{\text{inteiro}} + \underbrace{b_{-6} \times 2^{-1} + b_{-7} \times 2^{-2} + \dots}_{\text{fracionário}}$$

$$\Rightarrow b_{-5} = 1$$

$$0,4 = \underbrace{b_{-6}}_{\text{inteiro}} + \underbrace{b_{-7} \times 2^{-1} + b_{-8} \times 2^{-2} + \dots}_{\text{fracionário}}$$



## E se tivermos um número fracionário?

$$0,1 = b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + b_{-3} \times 2^{-3} + \dots$$

Multiplicando por 2:

$$1 + 0,2 = \underbrace{b_{-5}}_{\text{inteiro}} + \underbrace{b_{-6} \times 2^{-1} + b_{-7} \times 2^{-2} + \dots}_{\text{fracionário}}$$

$$\Rightarrow b_{-5} = 1$$

$$0,4 = \underbrace{b_{-6}}_{\text{inteiro}} + \underbrace{b_{-7} \times 2^{-1} + b_{-8} \times 2^{-2} + \dots}_{\text{fracionário}}$$

$$\Rightarrow b_{-6} = 0$$



## E se tivermos um número fracionário?

$$0,1 = b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + b_{-3} \times 2^{-3} + \dots$$

Multiplicando por 2:

$$1 + 0,2 = \underbrace{b_{-5}}_{\text{inteiro}} + \underbrace{b_{-6} \times 2^{-1} + b_{-7} \times 2^{-2} + \dots}_{\text{fracionário}}$$

$$\Rightarrow b_{-5} = 1$$

$$0,4 = \underbrace{b_{-6}}_{\text{inteiro}} + \underbrace{b_{-7} \times 2^{-1} + b_{-8} \times 2^{-2} + \dots}_{\text{fracionário}}$$

$$\Rightarrow b_{-6} = 0$$

*A partir desse ponto os coeficientes começam a se repetir!*





E se tivermos um número fracionário?

$$0,1 = b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + b_{-3} \times 2^{-3} + \dots$$



E se tivermos um número fracionário?

$$0,1 = b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + b_{-3} \times 2^{-3} + \dots$$

- $b_{-1} = 0$
- $b_{-2} = 0$
- $b_{-3} = 0$
- $b_{-4} = 1$
- $b_{-5} = 1$
- $b_{-6} = 0$
- $b_{-7} = 0$
- $b_{-8} = 1$
- $b_{-9} = 1$
- $b_{-10} = 0$
- $b_{-11} = 0$
- $b_{-12} = 1$
- $b_{-13} = 1$
- ...



E se tivermos um número fracionário?

$$0,1 = b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + b_{-3} \times 2^{-3} + \dots$$

- $b_{-1} = 0$
- $b_{-2} = 0$
- $b_{-3} = 0$
- $b_{-4} = 1$
- $b_{-5} = 1$
- $b_{-6} = 0$
- $b_{-7} = 0$
- $b_{-8} = 1$
- $b_{-9} = 1$
- $b_{-10} = 0$
- $b_{-11} = 0$
- $b_{-12} = 1$
- $b_{-13} = 1$
- ...

$$(0,1)_{10} = (0,00011)_{2}$$



E se tivermos um número fracionário?

$$0,1 = b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + b_{-3} \times 2^{-3} + \dots$$

- $b_{-1} = 0$
- $b_{-2} = 0$
- $b_{-3} = 0$
- $b_{-4} = 1$
- $b_{-5} = 1$
- $b_{-6} = 0$
- $b_{-7} = 0$
- $b_{-8} = 1$
- $b_{-9} = 1$
- $b_{-10} = 0$
- $b_{-11} = 0$
- $b_{-12} = 1$
- $b_{-13} = 1$
- ...

$$(0,1)_{10} = (0,0001\overline{1})_2$$

⇒ Representação infinita periódica  
(dízima em base 2)



E se quisermos ir de binário para decimal?

$(1101,01)_2$



E se quisermos ir de binário para decimal?

$$(1101, 01)_2 = (1101)_2 + (0, 01)_2$$



E se quisermos ir de binário para decimal?

$$(1101, 01)_2 = (1101)_2 + (0, 01)_2$$

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



E se quisermos ir de binário para decimal?

$$(1101, 01)_2 = (1101)_2 + (0, 01)_2$$

$$(1101)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13$$





E se quisermos ir de binário para decimal?

$$(1101, 01)_2 = (1101)_2 + (0, 01)_2$$

$$(1101)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13$$

$$(0, 01)_2 = 0 \times 2^{-1} + 1 \times 2^{-2}$$



E se quisermos ir de binário para decimal?

$$(1101, 01)_2 = (1101)_2 + (0, 01)_2$$

$$(1101)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13$$

$$(0, 01)_2 = 0 \times 2^{-1} + 1 \times 2^{-2} = 0,25$$



E se quisermos ir de binário para decimal?

$$(1101, 01)_2 = (1101)_2 + (0, 01)_2$$

$$(1101)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13$$

$$(0, 01)_2 = 0 \times 2^{-1} + 1 \times 2^{-2} = 0,25$$



E se quisermos ir de binário para decimal?

$$(1101, 01)_2 = (1101)_2 + (0, 01)_2$$

$$(1101)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13$$

$$(0, 01)_2 = 0 \times 2^{-1} + 1 \times 2^{-2} = 0,25$$

$$(1101, 01)_2 = (13, 25)_{10}$$



# Algoritmo de conversão entre decimal e binário


Input:  $N = (d_0, 0)_{10}$

```
1:  $k = 0$ 
2: Compute  $r$  and  $q$  such that  $N = 2 \times q + r$ 
3:  $b_0 = r$ 
4: while  $q \neq 0$  do
5:    $k = k + 1$ 
6:    $N = q$ 
7:   Compute  $r$  and  $q$  such that  $N = 2 \times q + r$ 
8:    $b_k = r$ 
9: end while
```

Output:  $N = (b_k b_{k-1} \cdots b_1 b_0)_2$

Exercício computacional:

Implemente o algoritmo acima no ambiente GNU

Octave. 




# Para pensar em casa ...


## Exercício teórico:

E se tivermos um número irracional? Pense como você poderia obter uma representação binária para  $\pi$ .

## Exercício computacional:

Implemente no ambiente GNU Octave um algoritmo para obter uma representação binária para um dado decimal fracionário. 

## Exercício teórico-computacional:

Pense num algoritmo para converter um decimal inteiro para uma base arbitrária  $\beta$ . Implemente esse algoritmo no ambiente GNU Octave. 



## Como citar esse material?

A. Cunha, *Conversão entre Diferentes Bases Numéricas*, Universidade do Estado do Rio de Janeiro – UERJ, 2020.

Essas notas de aula podem ser compartilhadas nos termos da licença Creative Commons BY-NC-ND 3.0, com propósitos exclusivamente educacionais.

