

Criar e simular circuitos lógicos

Introdução aos sistemas digitais e à representação binária

Objetivos da aula

No final da aula, os alunos deverão ser capazes de:

- reconhecer situações em que existem dois estados possíveis;
- relacionar esses dois estados com os valores **0 e 1**;
- compreender o significado de **bit**;
- distinguir os sistemas decimal, binário e hexadecimal;
- compreender que o valor de um algarismo depende da sua posição;
- determinar o valor decimal de números binários simples.

Como "pensa" um computador?

Nós utilizamos normalmente números como: 7, 25, 100, 2026...

Mas um computador trabalha internamente sobretudo com dois estados: **0 e 1**

Porquê? Porque é relativamente simples construir circuitos eletrónicos capazes de distinguir entre **dois estados**.

Podemos imaginar:

Estado	Valor
Desligado	0
Ligado	1



Que outros exemplos conheces em que só existam dois estados possíveis?

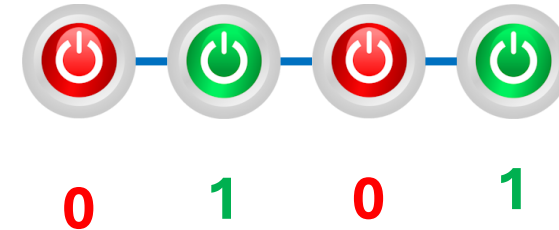


Uma máquina de 0 e 1

Imagina quatro interruptores: OFF — ON — OFF — ON

Podemos representá-los por: 0 — 1 — 0 — 1

Ou simplesmente: 0101



É desta ideia muito simples que surge a representação **binária**.

Cada posição pode ter apenas um de dois valores: **0** ou **1**.

O que é um bit?

Cada **0** ou **1** chama-se um **bit**.

bit = **b**inary **dig**it = dígito binário

Exemplos:

0 → 1 bit

10 → 2 bits

1011 → 4 bits

10110110 → 8 bits

Um conjunto de **8 bits** é normalmente designado por **byte**.



Quantos bits existem em 101101?

Quantas combinações conseguimos fazer?

Com **1 bit**:
0
1 } Temos **2 possibilidades**.

Com **2 bits**:
00
01
10
11 } Temos **4 possibilidades**.

E com **3 bits**?
000
001
010
011
100
101
110
111 } Temos **8 possibilidades**.

Nós usamos normalmente o sistema decimal

No dia a dia utilizamos o **sistema decimal**.

Temos **dez** algarismos: 0 1 2 3 4 5 6 7 8 9

Por isso dizemos que é um sistema de **base 10**.

Quando chegamos ao 9, precisamos de uma nova posição: ... 7, 8, 9, 10, 11 ...

Mas há aqui uma coisa que fazemos todos os dias sem pensar nela...

A posição importa

Observa:

325

O algarismo **3** não significa simplesmente 3.

Neste número temos:

- 3 centenas → **300**
- 2 dezenas → **20**
- 5 unidades → **5**

Portanto:

$$325 = 300 + 20 + 5$$

ou:

$$325 = 3 \times 100 + 2 \times 10 + 5 \times 1$$

É a **posição** que determina o valor de cada algarismo.



No número **747**, os dois algarismos 7 têm o mesmo valor?

E se tivermos apenas 0 e 1?

No sistema binário só podemos utilizar: $\begin{matrix} 0 \\ 1 \end{matrix} \longrightarrow$ Por isso é um sistema de **base 2**.

Em vez das posições:

Centenas	Dezenas	Unidades
100	10	1

temos:

8	4	2	1

Estes valores duplicam sempre: $1 \rightarrow 2 \rightarrow 4 \rightarrow 8 \rightarrow 16 \rightarrow 32 \rightarrow 64 \rightarrow 128 \dots$

Vamos decifrar um número binário

Temos: **1011**

Colocamos cada algarismo na posição correspondente:

8	4	2	1
1	0	1	1

Agora basta verificar as posições que estão **ligadas (1)**:

$$8 + 2 + 1 = 11$$

Portanto: **$1011_2 = 11_{10}$**

Podemos pensar no 1 como:
"Utiliza este valor."

e no 0 como:
"Ignora este valor."

Outro exemplo:

Qual é o valor de: **1101** ?

8	4	2	1
1	1	0	1

As posições ativas são:

$$8 + 4 + 1$$

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



E qual o valor de 1111?

Um terceiro sistema: **hexadecimal**

Mais tarde iremos trabalhar também com o **sistema hexadecimal**.

Neste sistema existem **16 símbolos**: **0 1 2 3 4 5 6 7 8 9 A B C D E F**

As letras representam números:

A	=	10
B	=	11
C	=	12
D	=	13
E	=	14
F	=	15

Resumindo:

Sistema	Base	Símbolos
Decimal	10	0 a 9
Binário	2	0 e 1
Hexadecimal	16	0 a 9 e A a F