

---



# FUNDAMENTOS DE LÓGICA DE PROGRAMAÇÃO

Conceitos

# Linguagem de programação

---

- Os computadores só entendem **LINGUAGEM DE MÁQUINA.**
- O computador funciona, basicamente, como uma máquina que recebe dados, processa-se e retorna um ou mais dados como resultado.



# Linguagem de Máquina

- Nós humanos usamos o **Sistema Decimal** para contar as coisas. O sistema de numeração decimal utiliza os algarismos **0, 1, 2, 3, 4, 5, 6, 7, 8 e 9**. Todos os outros números são formados pela união desses algarismos, levando em consideração a posição relativa de cada um deles.

O sistema decimal usa a BASE 10.

$$33 = 3 \cdot 10 + 3$$

DEZENAS

UNIDADES

3	3
---	---

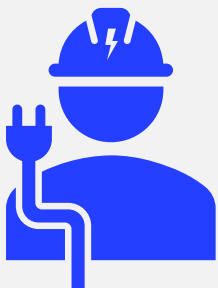
$$221 = 2 \cdot 100 + 2 \cdot 10 + 1$$

CENTENAS

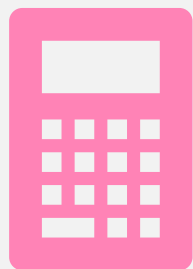
DEZENAS

UNIDADES

2	2	1
---	---	---



Se considerarmos um equipamento que usa corrente elétrica usa 2 estados: **LIGADO** que é representado pelo número 1 e **DESLIGADO** que é representado pelo número 0 (zero).



Então ao invés de usar o sistema de base 10 com 10 algarismos à disposição, ele usa um sistema de BASE 2 chamado de **SISTEMA BINÁRIO** usando os ALGARISMOS ZERO e UM. ( 0 e 1)

No caso de "vinte e quatro", por exemplo, se escreve 11000.  
E "vinte e cinco"? 11001. "Vinte e seis?" 11010.

Importante saber que cada algarismo desse recebe o nome de **BIT**. Um bit pode armazenar 2 valores possíveis (0 ou 1), o conjunto de 8 bits recebe o nome de **BYTE**, ou seja **8 bits = 1 byte**

# Tabela de Códigos ASCII

**ASCII** à **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange (Código Padrão Americano Para Troca de Informações) é um código binário que reúne as cadeias de bits, que codifica um conjunto de 128 sinais, 95 sinais gráficos (letras do alfabeto, sinais de pontuação e sinais matemáticos) e 33 sinais de controle, utilizando portanto apenas 7 bits para representar todos os seus símbolos. Isso significa que a tabela é composta por 256 símbolos para processar uma informação. Veja a tabela de Sinais Gráficos (imprimíveis).



Bin	Oct	Dec	Hex	Sinal
0010 0000	040	32	20	(espaço)
0010 0001	041	33	21	!
0010 0010	042	34	22	"
0010 0011	043	35	23	#
0010 0100	044	36	24	\$
0010 0101	045	37	25	%
0010 0110	046	38	26	&
0010 0111	047	39	27	'
0010 1000	050	40	28	(
0010 1001	051	41	29	)
0010 1010	052	42	2A	*
0010 1011	053	43	2B	+
0010 1100	054	44	2C	,
0010 1101	055	45	2D	-
0010 1110	056	46	2E	.
0010 1111	057	47	2F	/
0011 0000	060	48	30	0
0011 0001	061	49	31	1
0011 0010	062	50	32	2
0011 0011	063	51	33	3
0011 0100	064	52	34	4
0011 0101	065	53	35	5
0011 0110	066	54	36	6
0011 0111	067	55	37	7
0011 1000	070	56	38	8
0011 1001	071	57	39	9
0011 1010	072	58	3A	:
0011 1011	073	59	3B	;
0011 1100	074	60	3C	<
0011 1101	075	61	3D	=
0011 1110	076	62	3E	>
0011 1111	077	63	3F	?

Bin	Oct	Dec	Hex	Sinal
0100 0000	100	64	40	@
0100 0001	101	65	41	A
0100 0010	102	66	42	B
0100 0011	103	67	43	C
0100 0100	104	68	44	D
0100 0101	105	69	45	E
0100 0110	106	70	46	F
0100 0111	107	71	47	G
0100 1000	110	72	48	H
0100 1001	111	73	49	I
0100 1010	112	74	4A	J
0100 1011	113	75	4B	K
0100 1100	114	76	4C	L
0100 1101	115	77	4D	M
0100 1110	116	78	4E	N
0100 1111	117	79	4F	O
0101 0000	120	80	50	P
0101 0001	121	81	51	Q
0101 0010	122	82	52	R
0101 0011	123	83	53	S
0101 0100	124	84	54	T
0101 0101	125	85	55	U
0101 0110	126	86	56	V
0101 0111	127	87	57	W
0101 1000	130	88	58	X
0101 1001	131	89	59	Y
0101 1010	132	90	5A	Z
0101 1011	133	91	5B	[
0101 1100	134	92	5C	\
0101 1101	135	93	5D	]
0101 1110	136	94	5E	^
0101 1111	137	95	5F	_

Bin	Oct	Dec	Hex	Sinal
0110 0000	140	96	60	`
0110 0001	141	97	61	a
0110 0010	142	98	62	b
0110 0011	143	99	63	c
0110 0100	144	100	64	d
0110 0101	145	101	65	e
0110 0110	146	102	66	f
0110 0111	147	103	67	g
0110 1000	150	104	68	h
0110 1001	151	105	69	i
0110 1010	152	106	6A	j
0110 1011	153	107	6B	k
0110 1100	154	108	6C	l
0110 1101	155	109	6D	m
0110 1110	156	110	6E	n
0110 1111	157	111	6F	o
0111 0000	160	112	70	p
0111 0001	161	113	71	q
0111 0010	162	114	72	r
0111 0011	163	115	73	s
0111 0100	164	116	74	t
0111 0101	165	117	75	u
0111 0110	166	118	76	v
0111 0111	167	119	77	w
0111 1000	170	120	78	x
0111 1001	171	121	79	y
0111 1010	172	122	7A	z
0111 1011	173	123	7B	{
0111 1100	174	124	7C	
0111 1101	175	125	7D	}
0111 1110	176	126	7E	~



# Traduza o código:

```
01010000 01110010 01101111 01100111 01110010 01100001
01101101 01100001 01100100 01101111 01110010 00100000
01100110 01110010 01101111 01101110 01110100 00101101
01100101 01101110 01100100 00101100 00100000 01101001
01101110 11101101 01100011 01101001 01101111 00100000
01100100 01100101 00100000 01110100 01110101 01100100
01101111 00101110
```

Programador front-end, início de tudo.

---

**É PRA ISSO QUE SERVE UMA  
LINGUAGEM DE  
PROGRAMAÇÃO, ser  
a PONTE ENTRE A  
LINGUAGEM HUMANA E A  
LINGUAGEM DE MÁQUINA.**



## Tipo de Linguagens

---

*linguagens de  
baixo nível,*

---

*as linguagens  
intermediárias e*

---

*as linguagens de  
alto nível.*

---

Os computadores interpretam qualquer informação na forma de números binários. As linguagens de baixo nível são interpretadas diretamente por eles, tornando o resultado bastante veloz, ou seja estão mais próximas da máquina e mais próxima do entendimento do computador.



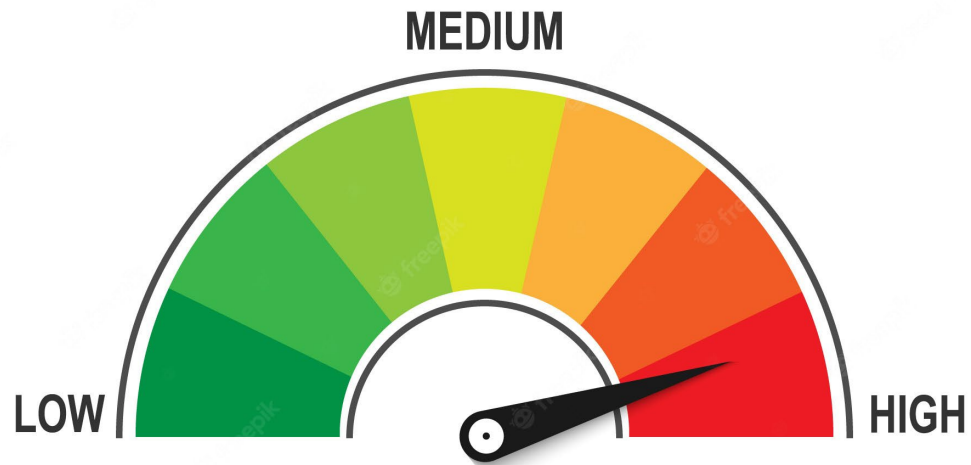


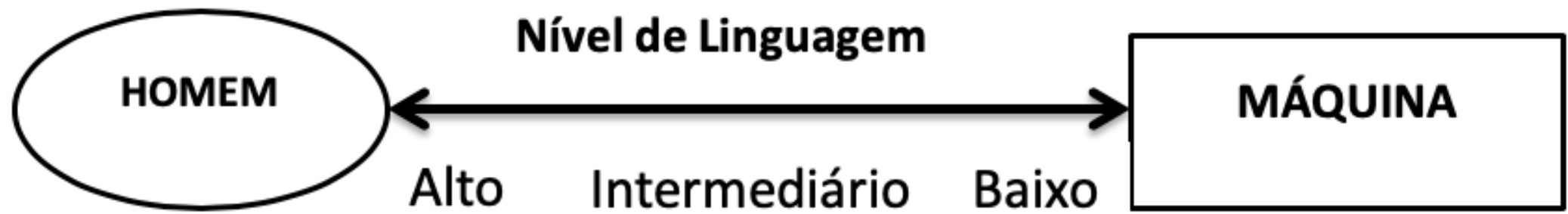
# Linguagens intermediária

São aquelas que estão próximas da máquina (computador) e ao mesmo tempo estão próximas da linguagem humana.

# Linguagens de alto nível

As linguagens de alto nível são mais fáceis de se trabalhar e entender, além de serem mais portáteis, são aquelas que estão mais próximas da linguagem humana é normalmente baseada em palavras do idioma inglês, facilitando a memorização e a lógica de programação.



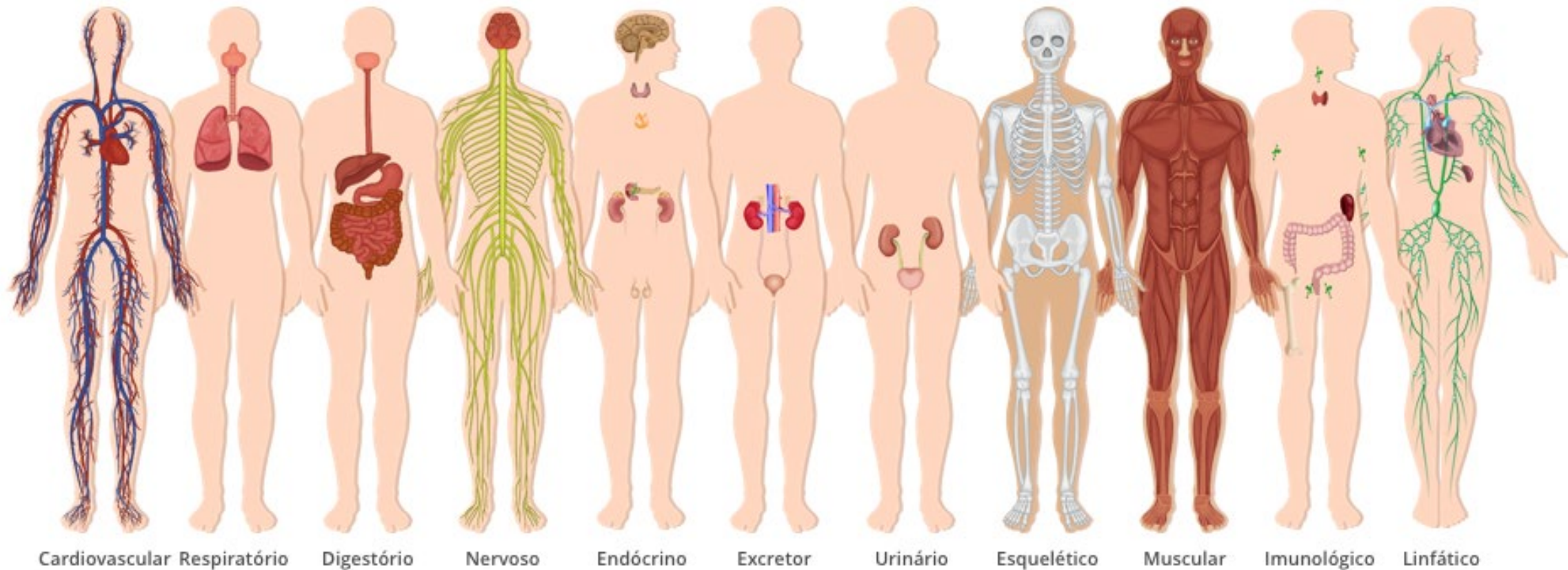




# LÓGICA DE PROGRAMAÇÃO

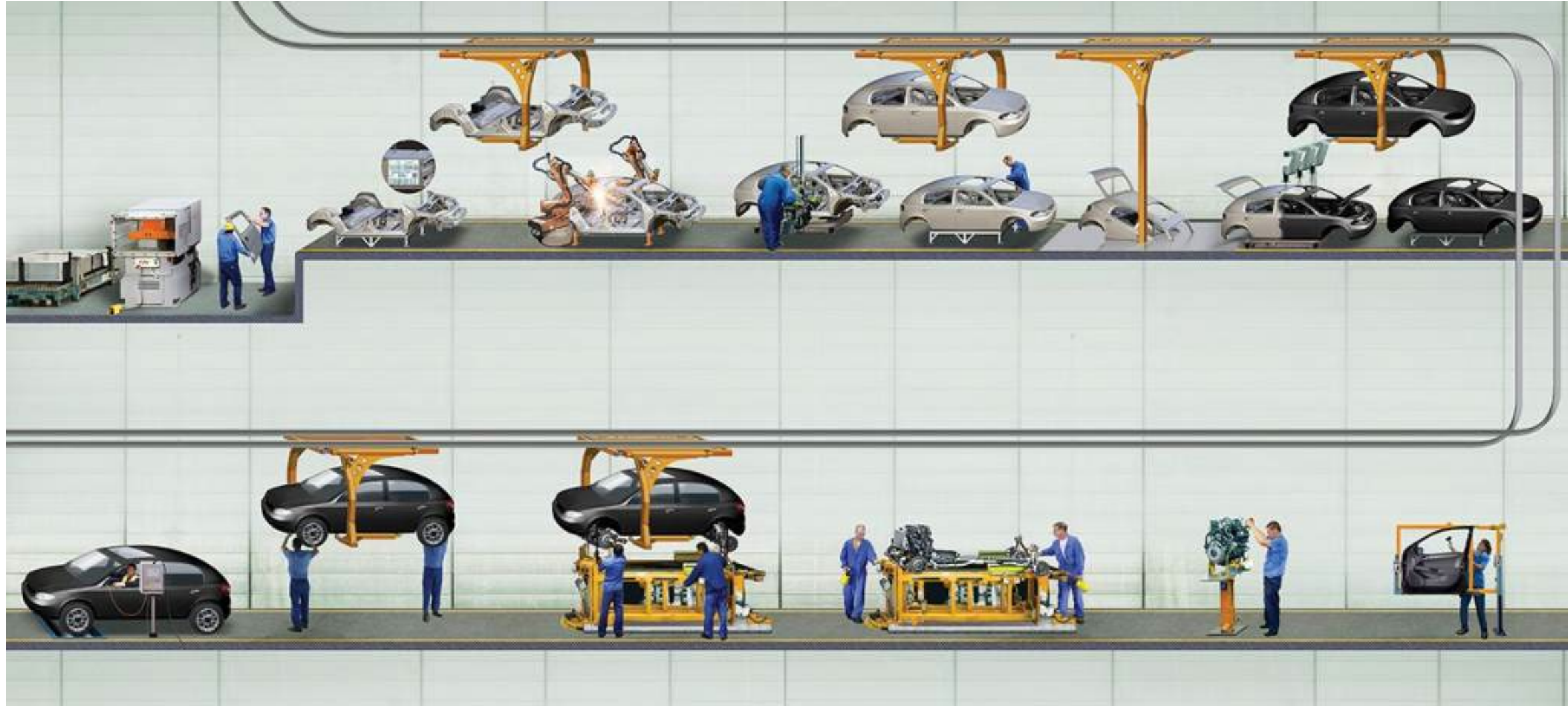


# Sistemas



Na informática, o conceito de sistema é semelhante, ou seja, é um conjunto de subsistemas ou módulos formados por programas que executam operações específicas. Como por exemplo o sistema para a montagem de um carro que é composto por quatro grandes subsistemas: projeto, estruturação, pintura, e montagem.



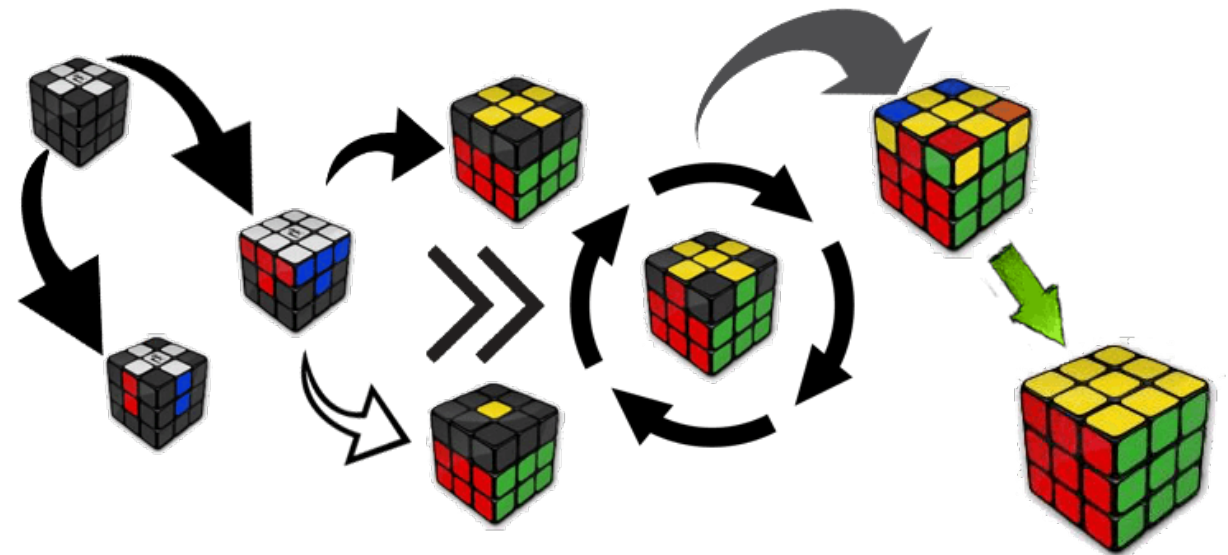




# Lógica

Segundo o dicionário *Aurélio*, lógica é a "**coerência de raciocínio, de ideias**", ou ainda a "sequência coerente, regular e necessária de acontecimentos, de coisas". Dessa forma podemos concluir que a lógica nos ensina a usar de forma organizada e corretamente as leis do pensamento.

**A lógica de programação** significa apenas contextualizar a lógica na programação de computadores, buscando a melhor sequência de ações para solucionar determinado problema. Ou seja é o conhecimento que precisamos para aprender qualquer linguagem de programação.



O primeiro trabalho do **programador** é a partir do problema proposto identificar as **entradas** ( o que preciso ter ), o **processamento** ( o que fazer ) e a **saída** (objetivo alcançado/solução/resposta).







ENTRADA DE DADOS = COLETA  
DADOS, PASSO EM QUE RECEBE AS  
INFORMAÇÕES NECESSÁRIAS



PROCESSAMENTO = TRATAMENTO  
DE DADOS (USO LÓGICO)



SAÍDA DE DADOS = ENTREGA DOS  
DADOS PROCESSADOS,  
APRESENTAÇÃO AO USUÁRIO.

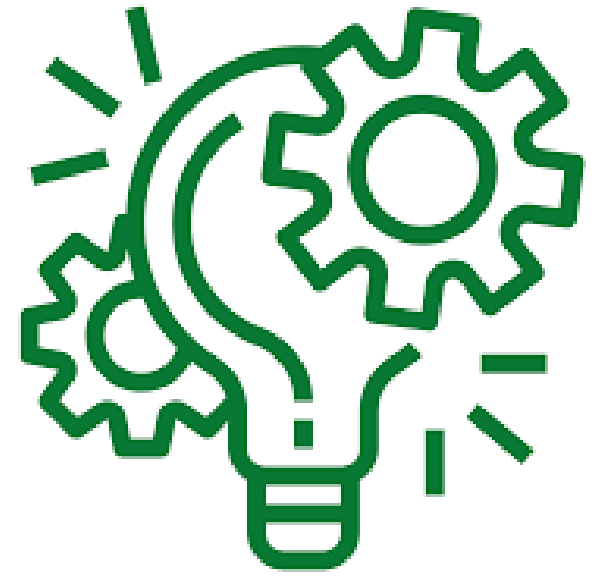


**Lógica:** o número 2 é menor que os números 4 e 5.

Na programação de computadores, o nome dessa sequência de ações é **algoritmo**. É justamente para isso que serve a lógica de programação, para a construção de algoritmos.

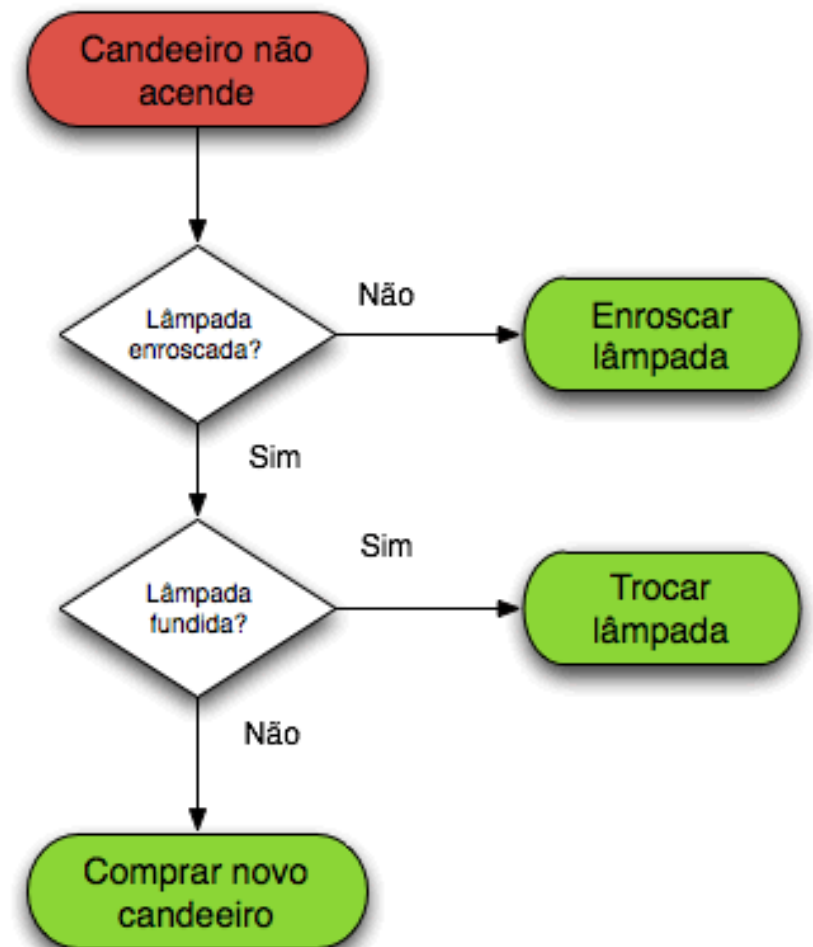
---

Em atividades rotineiras, não costumamos prestar atenção quando seguimos uma mesma ordem para executar tarefas. Porém, quando o assunto é programar, definir as **etapas** do que deve ser feito assume uma grande importância, uma vez que instruir um computador ainda é bem diferente do que instruir uma pessoa.



Quando a necessidade é desenvolver um programa ou rotina a ser executada pelo **computador**, precisamos deixar bem claro a sequência que deve ser seguida para atingir o resultado esperado.

A esse encadeamento lógico na programação, chamamos de **Lógica de Programação**, e a descrição de como fazer, definimos como **Algoritmos**.



---

# Exercícios de raciocínio lógico

1. Qual o número que completa a sequência: 1, 1, 2, 3, 5...
  2. Qual o número que completa a sequência: 8, 6, 7, 5, 6, 4...
  3. Uma mãe tem 30 reais para dividir entre suas duas filhas, que horas são?  
A) 2:30      B) 2:45      C) 1:45      D) 00:00
-

# ALGORITMO



É um conjunto lógico de operações predefinidas que resolva determinado problema.

Os **programas de computadores** nada mais são do que algoritmos escritos numa linguagem de computador (C, C++, C#, VB.Net, Java, Javascript entre outras) e que são interpretados e executados por uma máquina, no caso, um computador.



## Definições:

- 1) Sequência de passos que levam a execução de uma tarefa.
- 2) Conjunto lógico e finito de ações (instruções) que resolvem um determinado problema.
- 3) Descrição passo a passo para resolver um determinado problema para atingir um objetivo.





# Fazer um bolo

Entrada de  
Dados

- Ingredientes

Processamento

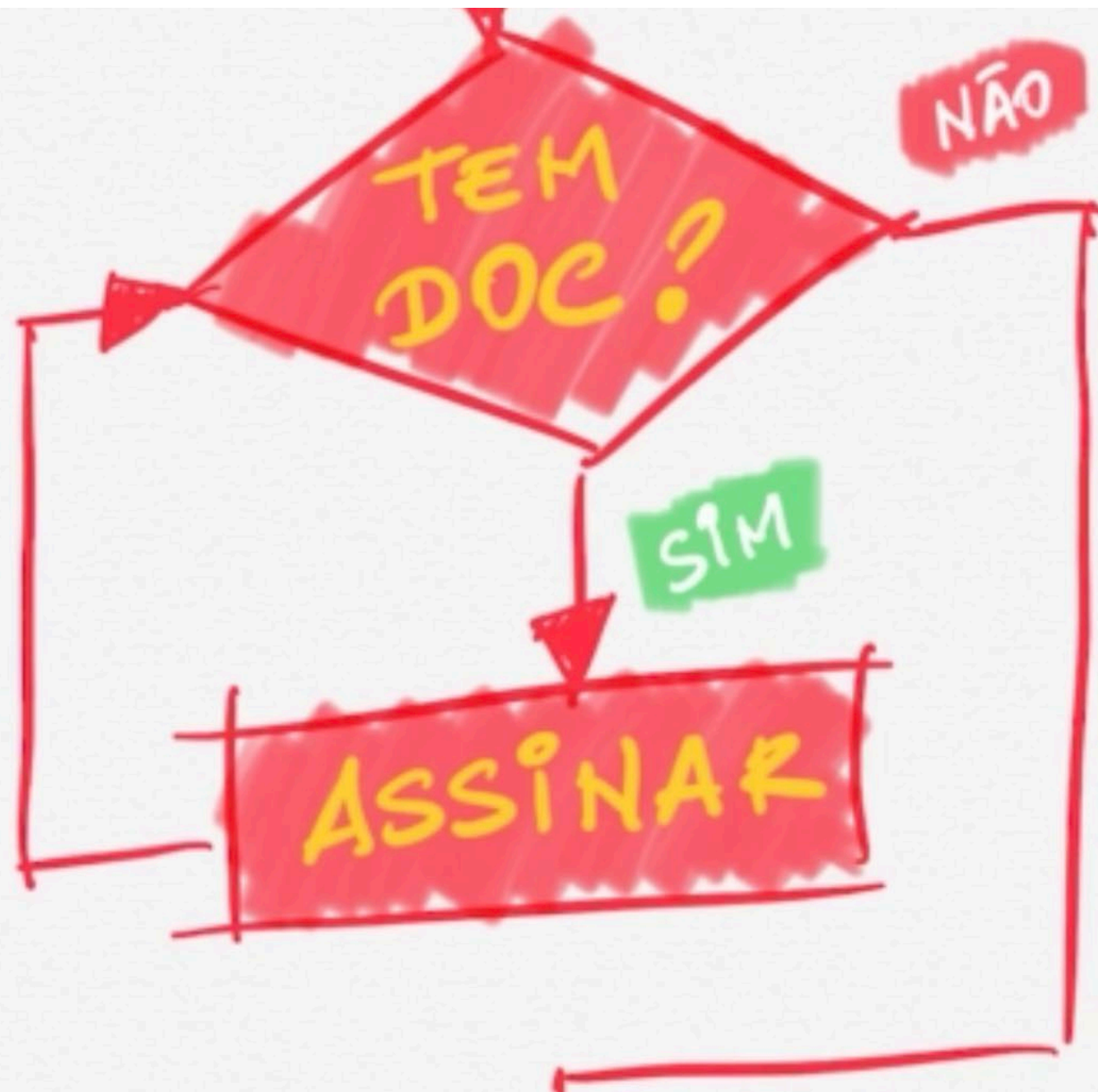
- Modo de preparo

Saída

- Bolo pronto

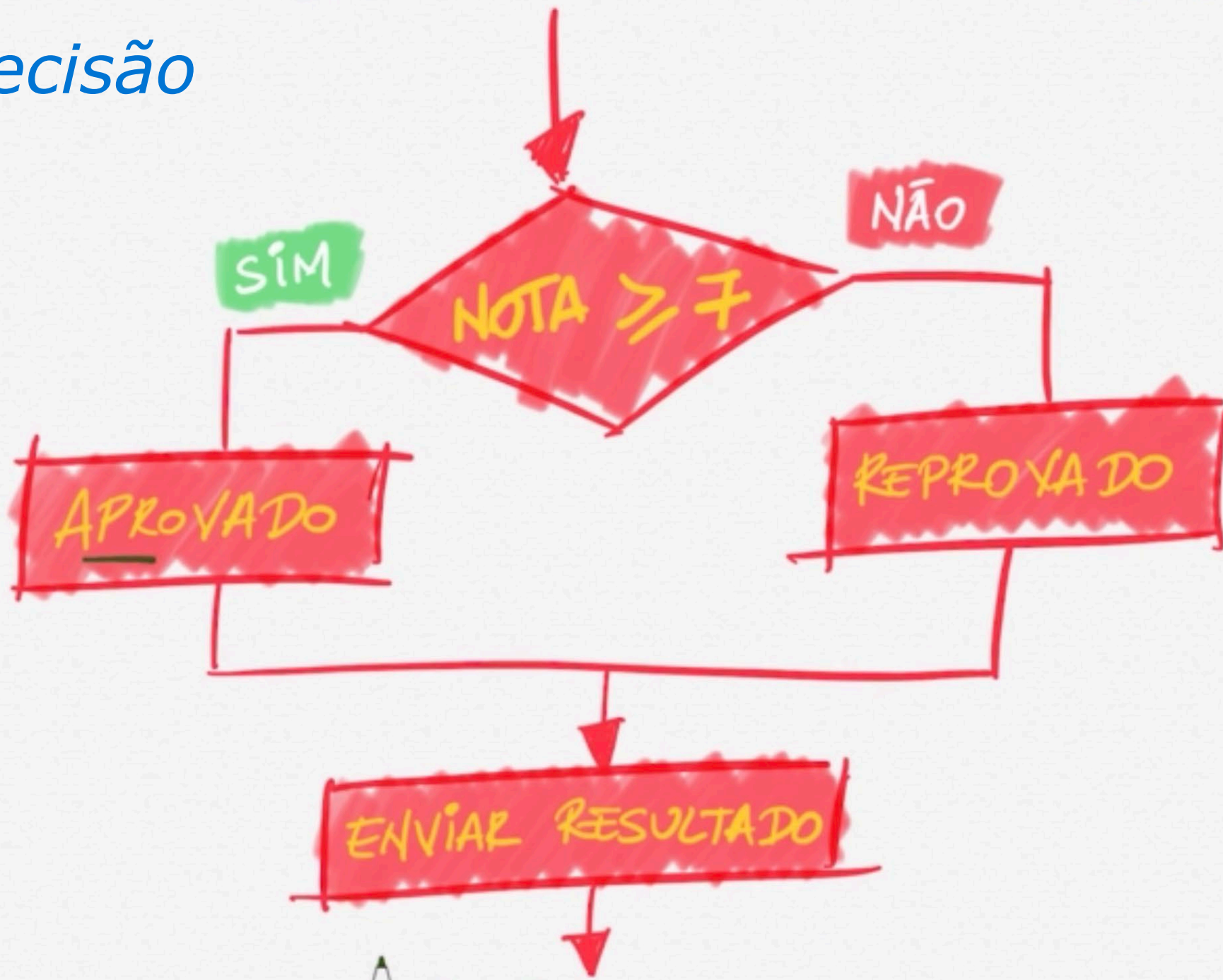
# Exercícios

- Fazer um algoritmo para chupar uma bala.  
Identificar as entradas, o processamento e a saída.
- Fazer um algoritmo para fazer um suco de laranja natural. Identificar as entradas, o processamento e a saída.

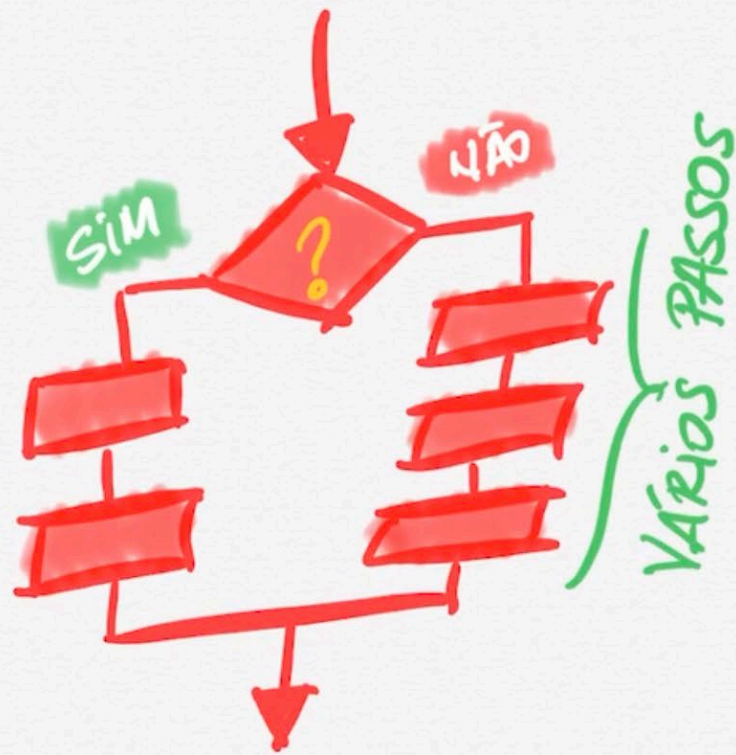




# Decisão



# MAIS SOBRE DECISÃO...





# VARIÁVEIS

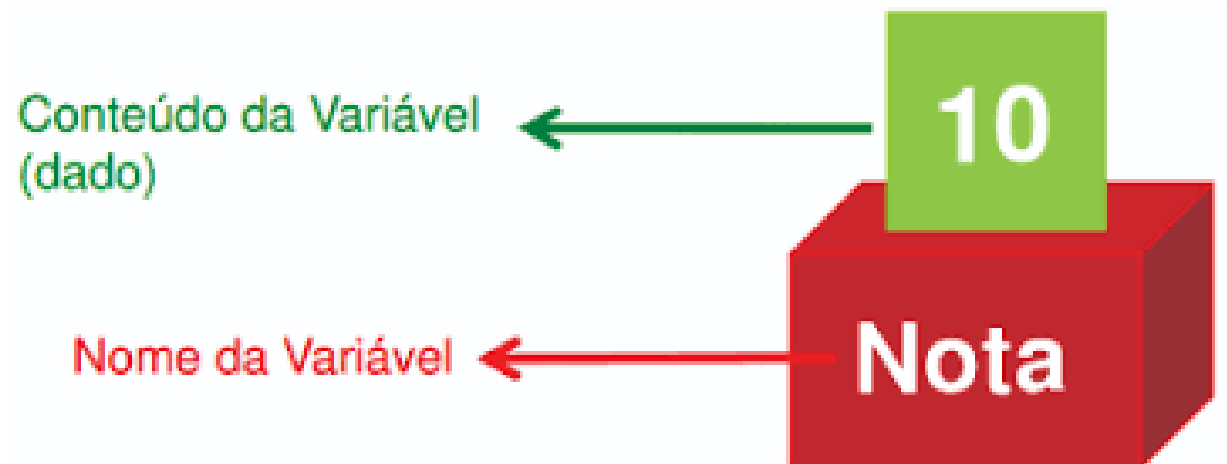
E seus tipos

---



# Variáveis

Variável é o nome utilizado para definir um ou mais valores que são manipulados pelos programas durante a sua operação. O nome “**variável**” é utilizado por ser um tipo de conteúdo que pode apresentar diferentes valores enquanto o sistema está em execução.



# Tipos de variáveis

Ao declarar uma variável nós precisamos informar de qual tipo é aquela variável. Cada linguagem de programação oferece um conjunto de tipos de variáveis, onde os mais comuns são:

**int:** Variável número do tipo inteiro, positivo ou negativo. (inteiro)

**float:** Variável numérica do tipo decimal. (real)

**double:** Variável numérica do tipo decimal de precisão dupla. (real)

**char:** Variável que representa um caractere do tipo texto. (caractere)

**string:** Variável que representa um conjunto de caracteres do tipo texto. (caractere)

**boolean:** Variável booleana, ou seja, que pode representar somente dois valores, verdadeiro ou falso. (lógico)

# Constantes

Ao contrário das variáveis, que podemos alterar o valor conforme a necessidade do algoritmo a ser desenvolvido, as constantes precisam ser inicializadas e **não podem** ter o seu valor alterado.

$$\pi = 3,14159265358979323846\dots$$

VALOR DE PI

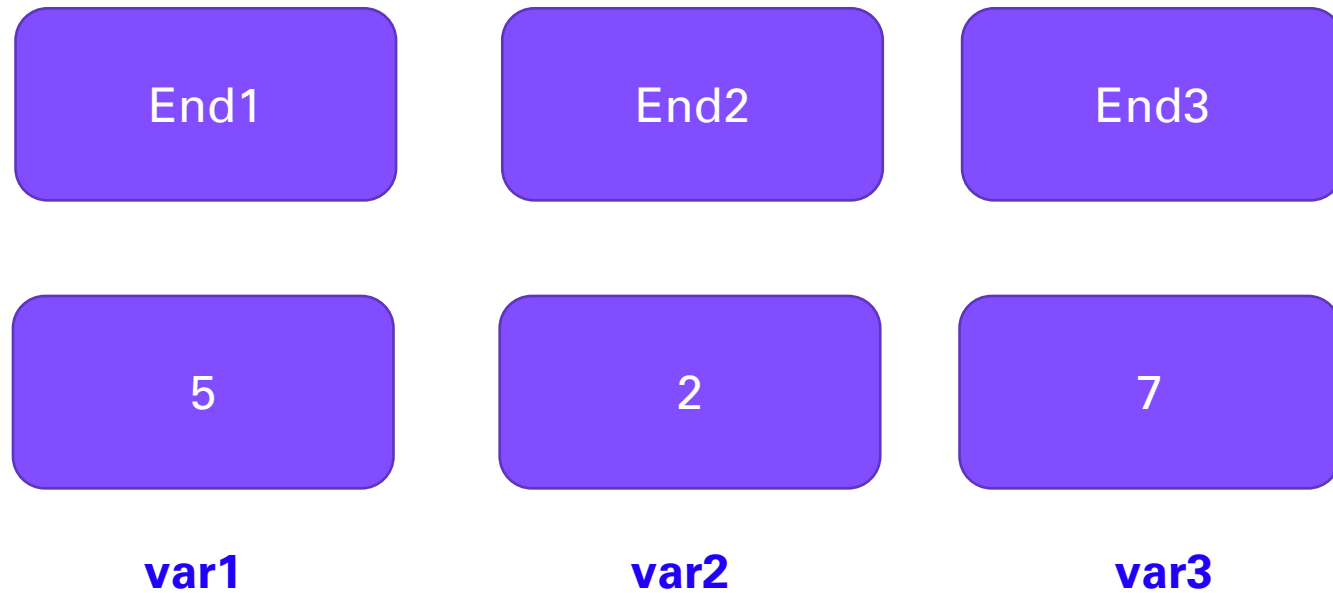
$$g = \frac{GM}{r^2}$$

$$g = \frac{6,7 \cdot 10^{-11} \cdot 6,4 \cdot 10^{23}}{(3,4 \cdot 10^6)^2} = 3,709 \cdot 10^{-11+23-12}$$

$$g \approx 3,71 \text{ m/s}$$

VALOR DA ACELERAÇÃO  
DA GRAVIDADE

É um endereço na memória do computador que guarda um valor usado em um algoritmo.



`var1 = 5`

`var2 = 2`

`var3 = var1 + var2`

`var1 = 5`

`var1 = 7`

`var1 = var1 + 2`



**var1**

```
var1 = "senac"  
var2 = 'javascript'  
var3 = `internet`
```

5

**var1**

2

**var2**

7

**var3**

# REGRAS PARA A NOMENCLATURA

**Podem**  
começar com  
letra, \$ ou \_

**Não podem**  
começar com  
números

**É possível**  
**usar** letras ou  
números

**É possível**  
**usar** acentos  
e símbolos

**Não podem**  
conter  
espaços

**Não podem**  
ser palavras  
**reservadas**

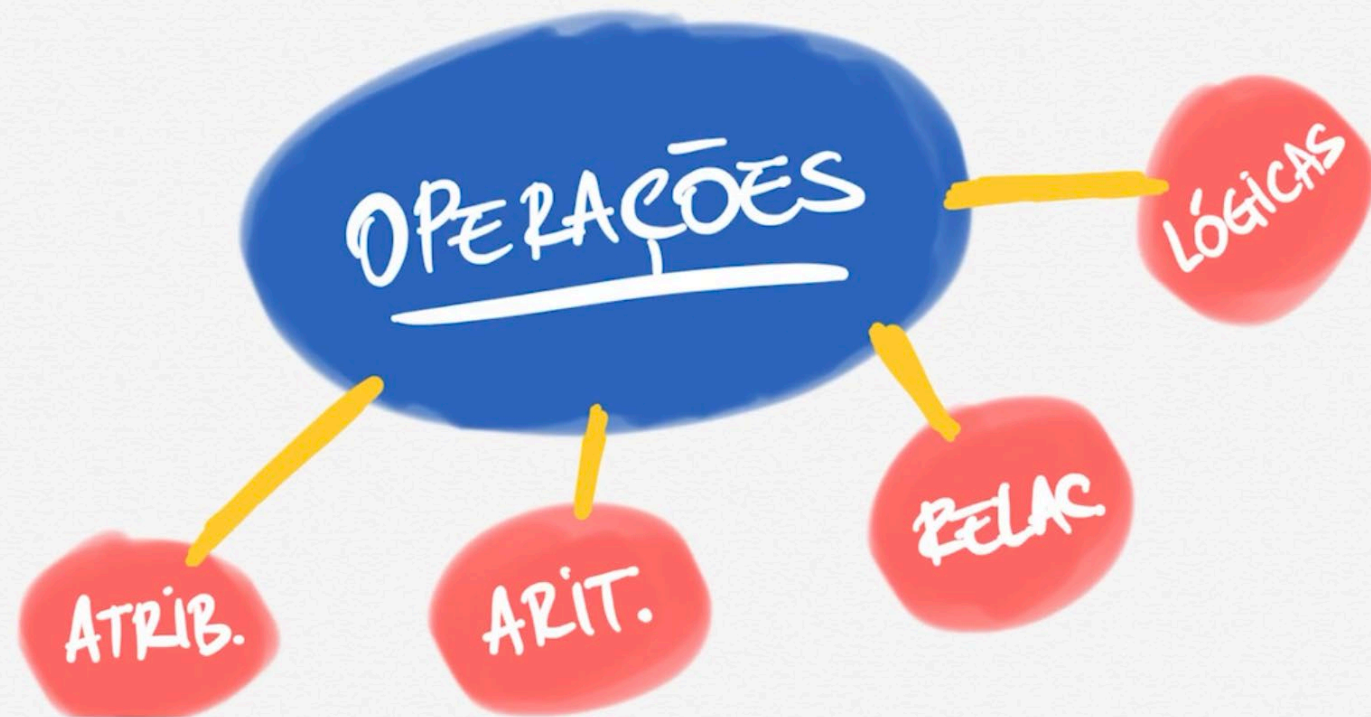
EXEMPLOS NO  
NODEJS

# Dicas para a criação de variáveis

- **Maiúsculas** e **minúsculas** fazem a diferença
- Tente escolher **nomes coerentes** para as variáveis
- Evite se tornar um "**programador alfabeto**" ou um "**programador contador**"



# VARIÁVEIS E CONSTANTES



# Tipo de Variáveis

5 18

-12 0.5

-15.9 8.0

3.14

*number*

"Internet"

'javascript'

*string*

true

false

*boolean*

---

- +
- 
- 

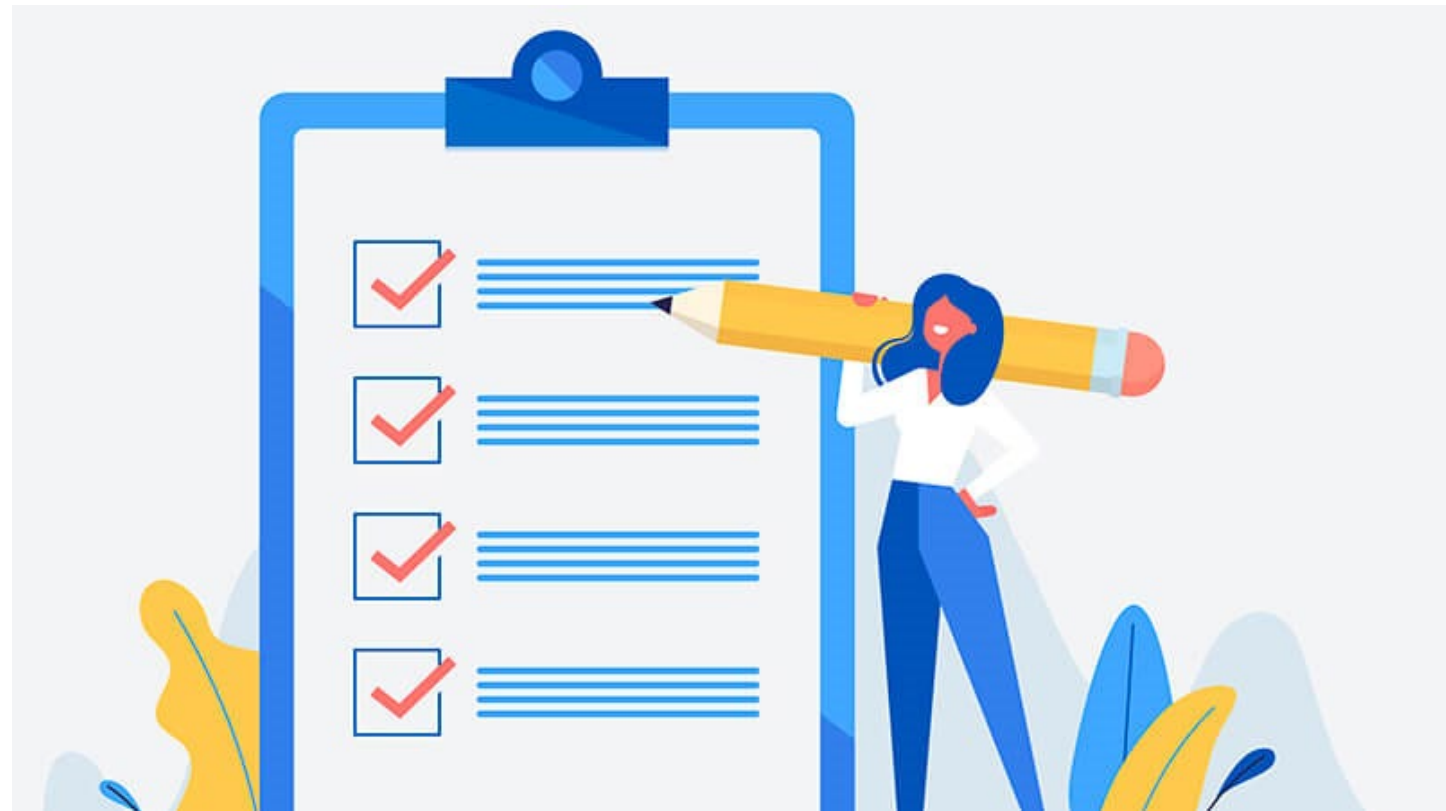
# ESTRUTURA DE DADOS

# Dados

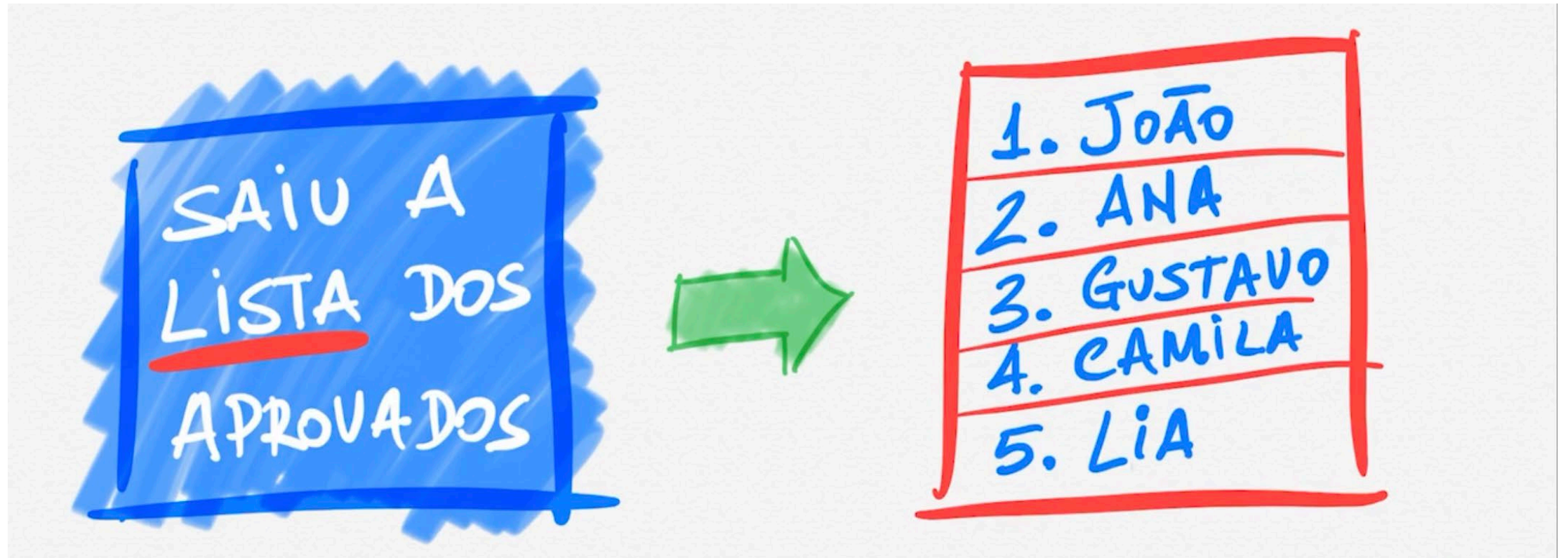


# Estrutura de Dados

Organizar os dados existentes

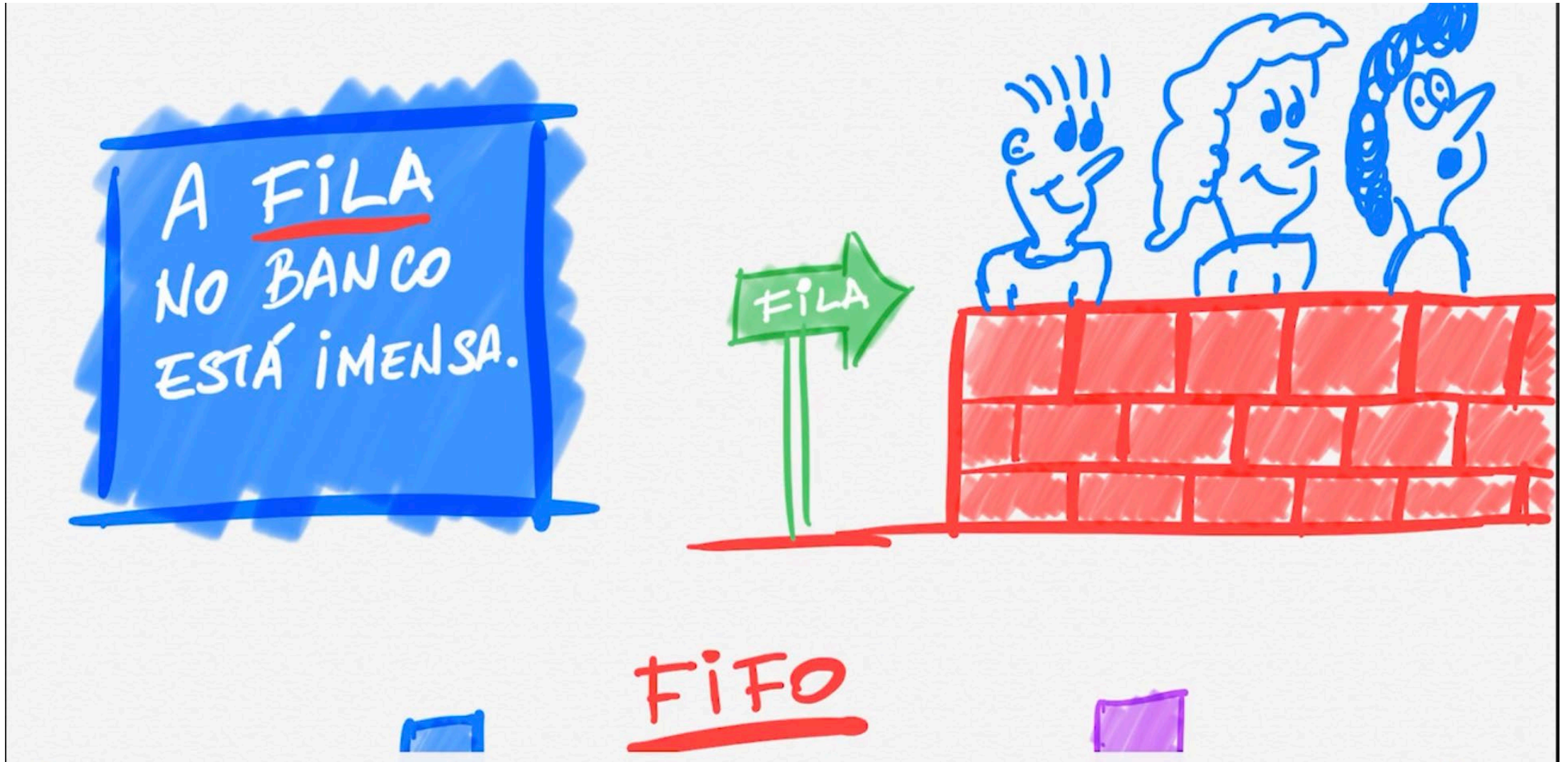


# Lista

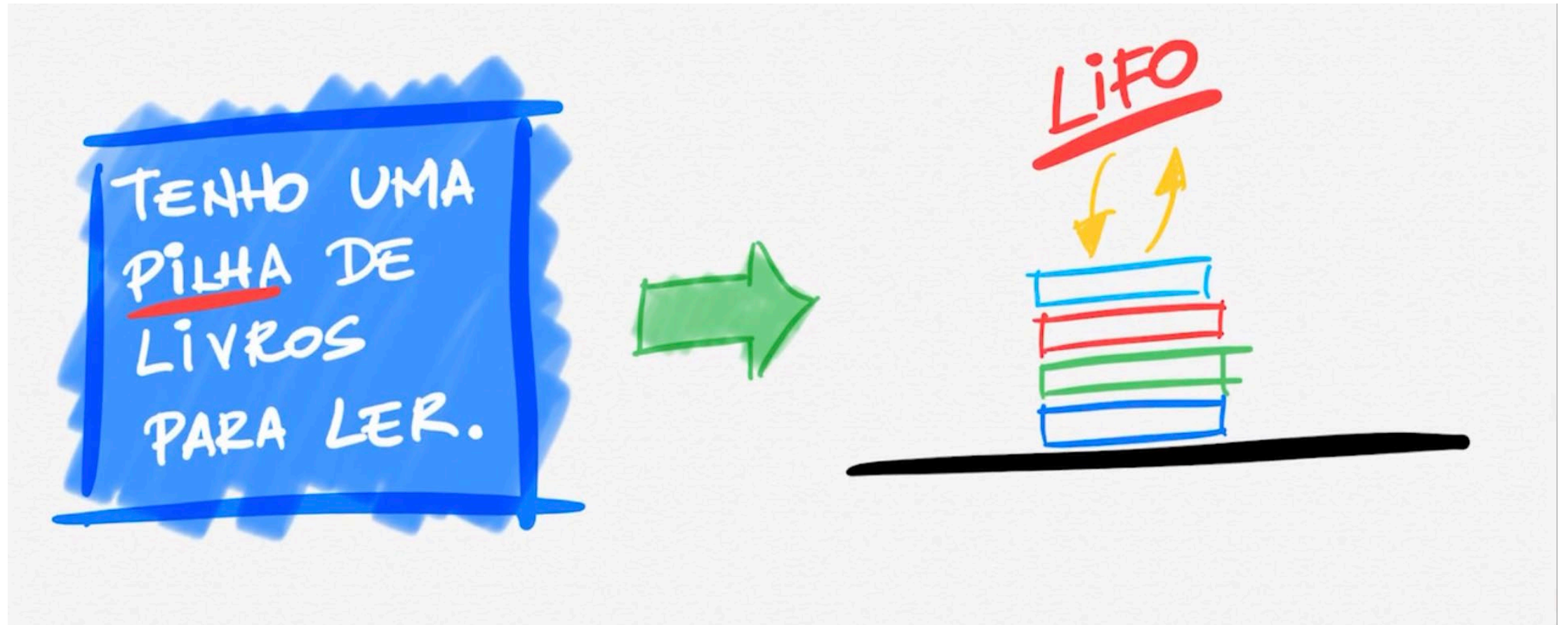




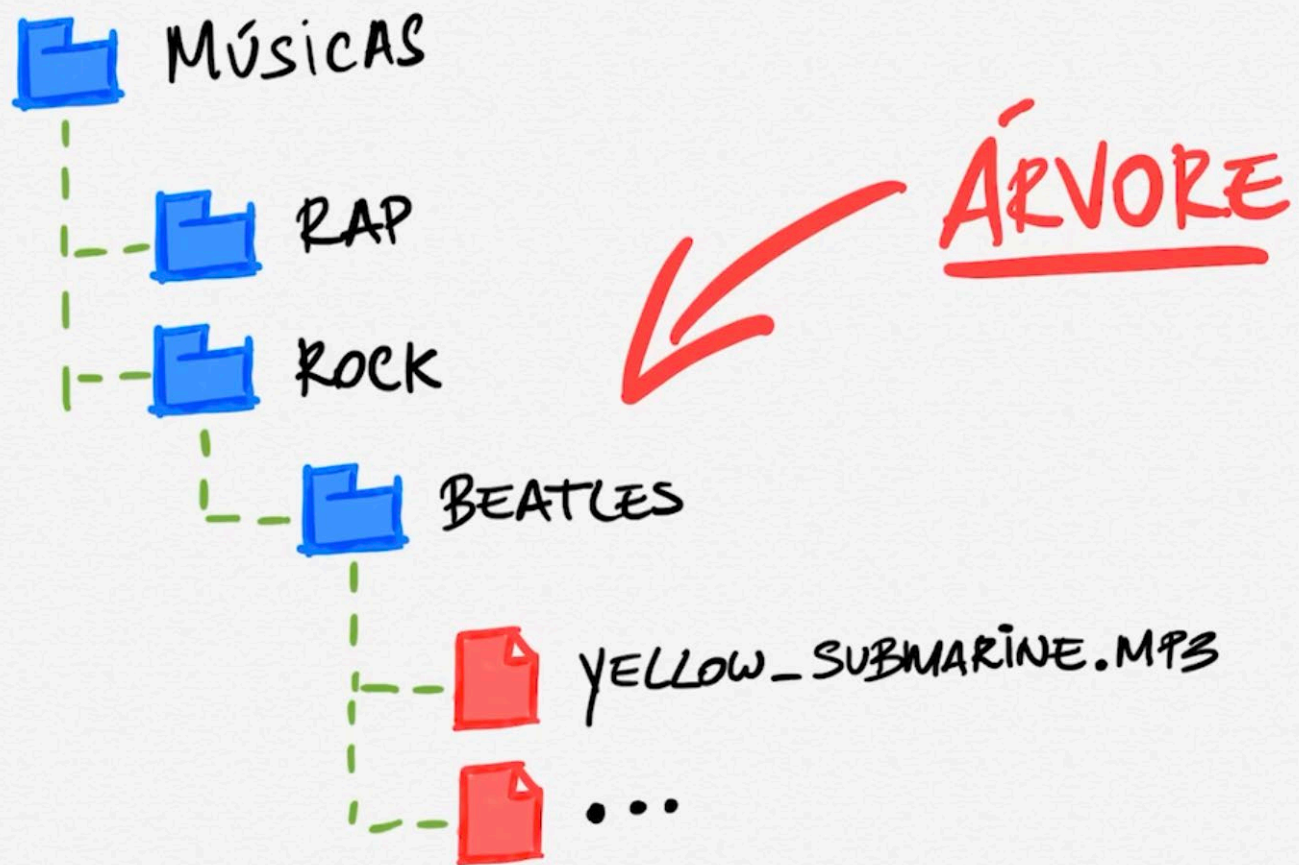
# Fila



# Pilha







# Tabela

COD.	NOME	STATUS
1	MARIA	A
2	João	A
3	PEDRO	I
4	ANA	A

TABELA

