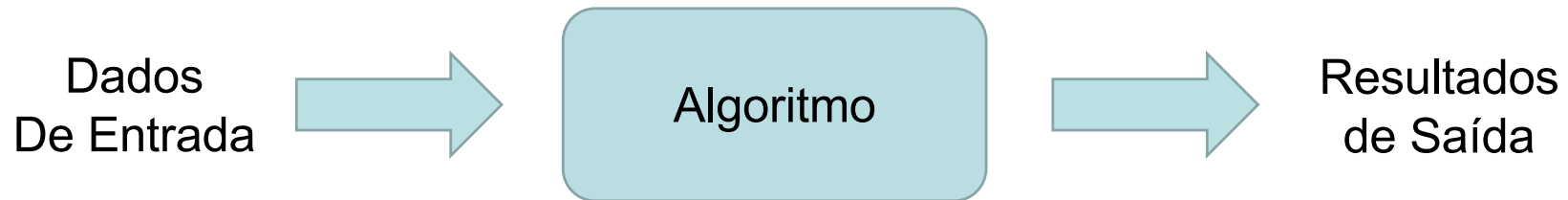


- **Tipos de Dados**
- **Variáveis e Constantes**
- **Operadores**
 - **Aritméticos**
 - **Lógicos**
 - **Relacionais**
- **Expressões**

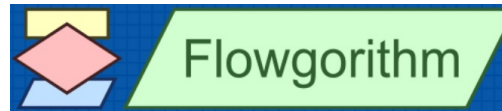


<https://dev.to/wmitrut/001-basicao-do-js-tipos-de-dados-em-javascript-parte-1-primitivos-1ihj>



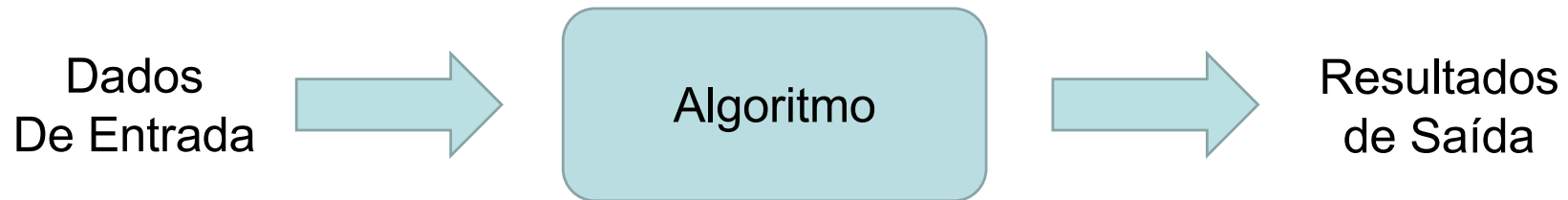
Tipo de Dados:

- Números inteiros: tipo `inteiro`
- Números reais: tipo `real`
- Texto (caracteres individuais): tipo `carácter`
- Texto (sequências de caracteres): tipo `cadeia` (ou `string`)
- Valores lógicos: tipo `lógico` (ou `booleano`)



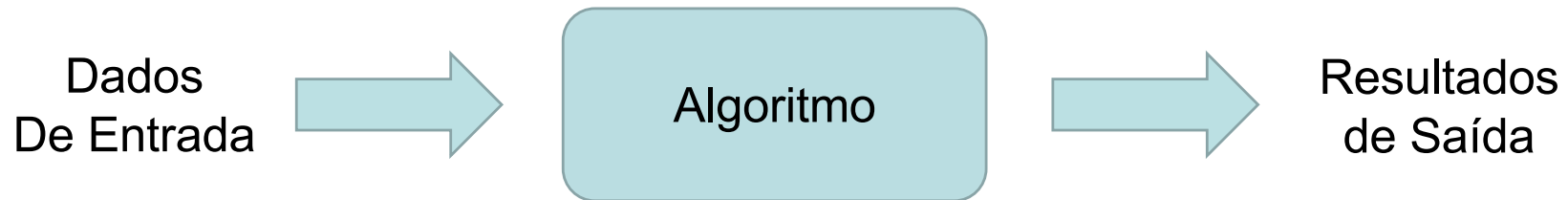
Data Type	Notes
Boolean	Stores either Boolean true or false
Real	Stores a real number.
Integer	Stores an integer number.
String	Stores textual data.

<http://www.flowgorithm.org/documentation/language/types.html>



- Os valores usados por um algoritmo (dados de entrada, resultados de saída, valores usados internamente) são guardados em **variáveis**;
- As **constantes** destinam-se a representar valores inalteráveis;
- Uma variável ou uma constante é representada por um nome ou **identificador**;
- Os identificadores obedecem a regras para a sua construção.

(Flowgorithm: ver <http://www.flowgorithm.org/documentation/language/keywords.html>)



- Regras para os identificadores:
 - Podem ser compostos por caracteres alfabéticos e/ou números;
 - Devem começar por um carácter alfabético;
 - Não podem conter espaços ou caracteres especiais;
 - Não podem ser usadas palavras reservadas (*keywords*).
- Por convenção, as **variáveis** devem ser escritas em minúsculas (na sua maior parte) e as **constantes** devem ser escritas em maiúsculas.

Quais dos seguintes identificadores são válidos (linguagem algorítmica)?

soma	✓	aluno1	✓
nome2	✓	TOTAL	✓
nomeDoAluno	✓	inteiro	✗
PI	✓	a+b	✗
nome aluno	✗	soma?	✗
2alunos	✗	nome-aluno	✗
nomedoaluno	✓	nome_aluno	✓

Uma **variável**:

- É representada por um **identificador** (nome da variável);
- Possui um **conteúdo** (o valor que a variável representa);
- Ocupa um **endereço de memória** (valor numérico que indica a posição na memória do computador);
- Tem um **tipo de dados** associado (o tipo de dados define a gama de valores possíveis e o espaço ocupado na memória);
- Tem de ser **inicializada** (ser-lhe atribuído um valor inicial) antes de ser usada em qualquer operação ou instrução (usando os operadores de atribuição = ou <-).

- As **constantes** têm características semelhantes às variáveis mas o valor inicialmente atribuído não pode mais ser alterado.
- **Quando usar constantes?**
 - Quando existem valores inalteráveis usados com frequência no algoritmo.
(por exemplo, o valor da constante π , $PI = 3.1416$)
 - Quando existem valores, fixos num dado momento, mas que podem ser alterados em versões futuras do fluxograma.
(por exemplo, uma taxa de juro)
 - Quando se pretende tornar o algoritmo mais legível.
(por exemplo, escrever PI em vez de 3.1416)

Operadores Aritméticos

Operador	Operação	Síntaxe
+	Adição	$a + b$
-	Subtração	$a - b$
*	Multiplicação	$a * b$
/	Divisão	a / b
%	Resto da divisão inteira	$a \% b$
^	Potenciação	$a ^ b$
+ -	Sinal	$+a \ -b$

Operadores Aritméticos

Operador	Associatividade	Prioridade
\wedge	Direita para a esquerda	MAIOR
Sinal + e -	Direita para a esquerda	
* / %	Esquerda para a direita	
+ -	Esquerda para a direita	MENOR

Operadores Relacionais

Operador	Operação	Síntaxe
<code>==</code>	“igual a”	<code>a == b</code>
<code><</code>	“menor que”	<code>a < b</code>
<code><=</code>	“menor ou igual a”	<code>a <= b</code>
<code>></code>	“maior que”	<code>a > b</code>
<code>>=</code>	“maior ou igual a”	<code>a >= b</code>
<code>!=</code>	“diferente de”	<code>a != b</code>

Operadores Relacionais

Operador	Associatividade	Prioridade
<, <=, >, >=	Esquerda para a direita	MAIOR
=, !=	Esquerda para a direita	MENOR

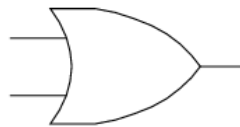
Álgebra de Boole

(George Boole: matemático e filósofo inglês do séc. XIX)

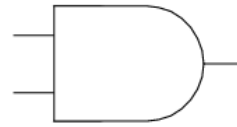
- A Álgebra de Boole define operações lógicas e a forma de as usar em expressões lógicas.
- Usa variáveis lógicas (ou *booleanas*) que apenas podem assumir o valor lógico *verdadeiro* ou *falso*.
- **Operações lógicas e operadores lógicos:**
 - **Negação** – não – NOT
 - **Conjunção** – e – AND
 - **Disjunção** – ou – OR

Operações lógicas e operadores lógicos:

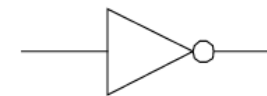
- As operações lógicas são também reproduzidas por portas lógicas.
- As portas lógicas são circuitos eletrónicos que operam sobre um ou mais sinais de entrada para produzirem um sinal de saída.
- Nas tecnologias mais comuns, o circuito lógico distingue dois intervalos distintos de tensão, que são interpretados como 1 ou 0.



OR

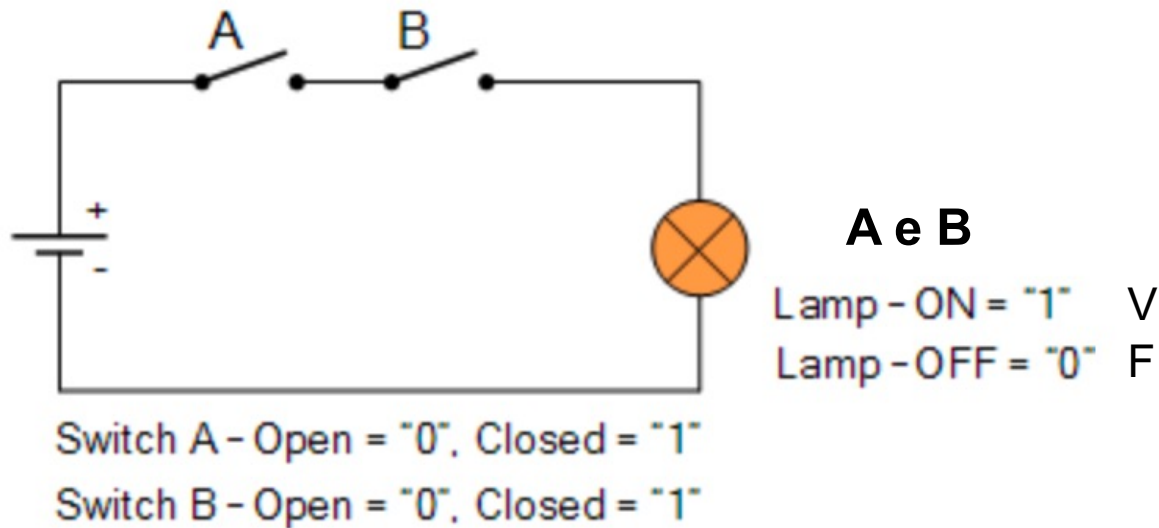


AND



NOT

Conjunção – e – AND



https://www.electronics-tutorials.ws/boolean/bool_1.html

Tabela de Verdade

A	B	A e B
V	V	V
V	F	F
F	V	F
F	F	F

A lâmpada (A e B) só acende (V) se ambos os interruptores estiverem ligados (V)

Disjunção – ou – OR

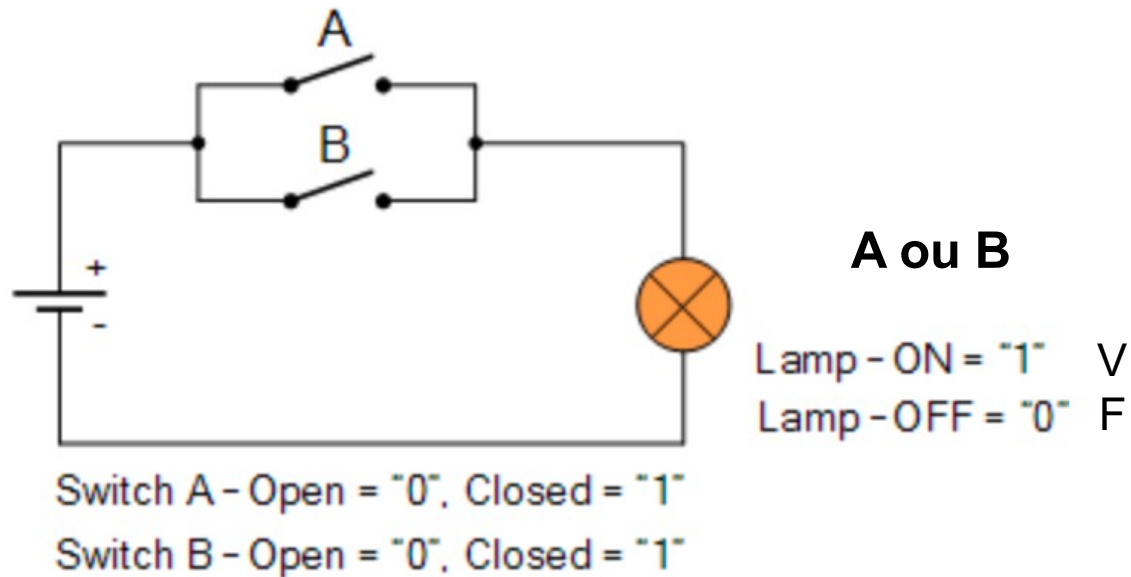


Tabela de Verdade

A	B	A ou B
V	V	V
V	F	V
F	V	V
F	F	F

https://www.electronics-tutorials.ws/boolean/bool_2.html

A lâmpada (A e B) acende (V) se um ou ambos os interruptores estiverem ligados (V)

Operadores Lógicos

a	b	a e b	a ou b	não a
V	V	V	V	
V	F	F	V	F
F	V	F	V	V
F	F	F	F	

Operadores Lógicos

Operador	Associatividade	Prioridade
não	-	MAIOR
e	-	
ou	-	
		MENOR

Propriedades da Álgebra de Boole

- **Leis comutativas**

- $a \text{ e } b = b \text{ e } a$
- $a \text{ ou } b = b \text{ ou } a$

- **Leis associativas**

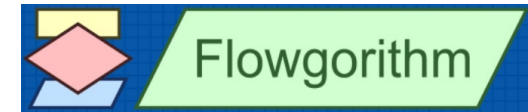
- $a \text{ e } (b \text{ e } c) = (a \text{ e } b) \text{ e } c$
- $a \text{ ou } (b \text{ ou } c) = (a \text{ ou } b) \text{ ou } c$

- **Leis distributivas**

- $a \text{ e } (b \text{ ou } c) = (a \text{ e } b) \text{ ou } (a \text{ e } c)$
- $a \text{ ou } (b \text{ e } c) = (a \text{ ou } b) \text{ e } (a \text{ ou } c)$

- **Leis de De Morgan (matemático inglês do séc. XIX)**

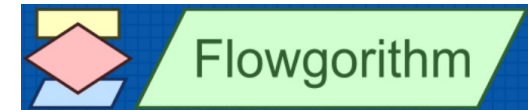
- $\text{não } (a \text{ e } b) = \text{não } a \text{ ou não } b$
- $\text{não } (a \text{ ou } b) = \text{não } a \text{ e não } b$



Operator	C Family
Equality	==
Inequality	!=
Less Than or Equal	<=
Greater Than Or Equal	>=
Logical Not	!
Logical And	&&
Logical Or	
Multiply	*
Divide	/
Modulo	%

BASIC Family
not
and
or

<http://www.flowgorithm.org/documentation/language/operators.html>



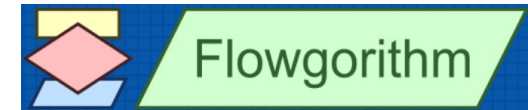
Name	Basic Family
String Concatenation	&
Exponent	^

String concatenation (concatenação/junção de cadeias de caracteres):

“algoritmos” & “e” & “programação” resulta em “algoritmos e programação”

<http://www.flowgorithm.org/documentation/language/operators.html>

Expression	Result
$1 + 3^2$	10
$10 * 2 + 5 * 6$	50
$7 * (4 - 1)$	21
$6 / 3 * 2$	4
$10 \bmod 3$	1
$10 \% 3$	1



<http://www.flowgorithm.org/documentation/language/operators.html>

Exemplo:

- **A que equivale a negação da expressão $a > b$ e $c \leq 5$?**

A negação de $a > b$ e $c \leq 5$ é $a \leq b$ ou $c > 5$.

Verifica-se pela aplicação das leis de De Morgan:

$\text{não}(a > b \text{ e } c \leq 5)$ equivale a $\text{não}(a > b)$ ou $\text{não}(c \leq 5)$,
ou seja, $a \leq b$ ou $c > 5$

- **Na expressão $a \leq b$ ou $c > 5$,
se $a=5$, $b=5$ e $c=5$, qual o valor lógico da expressão?**

$a \leq b$ corresponde a $5 \leq 5$ (verdadeiro)

$c > 5$ corresponde a $5 > 5$ (falso)

a tabela de verdade da disjunção (ou) diz que
quando temos V ou F o resultado é V

Determine o valor lógico de cada uma das seguintes expressões, sabendo que

$a = \text{falso}$, $b = \text{verdadeiro}$, $c = \text{verdadeiro}$

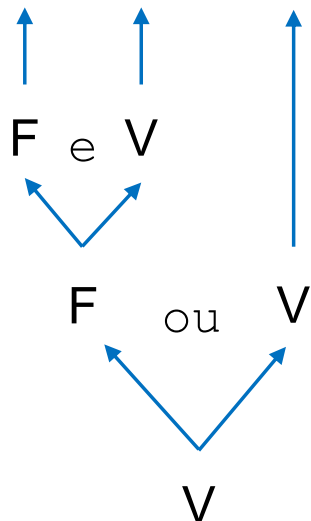
- | | |
|--|---|
| a) $a \text{ e } b \text{ ou } c$ | h) $a \text{ e } (c \text{ ou } b)$ |
| b) $\text{não}(a \text{ e } b) == \text{não } a \text{ ou não } b$ | i) $a \text{ ou } b \text{ e } c$ |
| c) $\text{não } b \text{ e não } a \text{ e } c$ | j) $b \text{ e } a \text{ ou não } c$ |
| d) $a \text{ ou } b \text{ ou } c$ | k) $(a \text{ ou } b) \text{ e não } (c \text{ ou } b)$ |
| e) $b \text{ ou } a \text{ e } c$ | l) $\text{não } (a \text{ e } b) \text{ ou não } c$ |
| f) $b \text{ e } a \text{ ou } c$ | m) $\text{não } a \text{ ou não } b \text{ e não } c$ |
| g) $a \text{ e } c \text{ ou } b$ | n) $\text{não } (b \text{ ou } c)$ |

Determine o valor lógico de cada uma das seguintes expressões, sabendo que

`a = falso, b = verdadeiro, c = verdadeiro`

`a e b ou c` (também se pode escrever `a && b || c` ou `a and b or c`)

equivale a `(a e b) ou c` porque `e` tem maior prioridade



A expressão `a e b ou c` é **verdadeira** para

`a = falso`

`b = verdadeiro`

`c = verdadeiro`

Calcule o resultado das seguintes expressões aritméticas, sabendo que

$x = 10, y = 8, z = 25$ e $t = 10$ (valores inteiros)

a) $x * y + (z - t) / 2.0$

87.5

b) $t * z + z \% x$

255

c) $2 ^ t + z \% y + z / y$

1028

d) $y * x ^ 2 ^ 2 + y$

80 008

Qual o resultado de cada uma das seguintes operações:

a) $4 + 6$

b) $3 + 4.7$

c) $5.7 + 2.7$

d) $5 - 2$

e) $0.33 - 2.8$

f) $5 * 2.6$

g) $13 \% 4$

h) $4 < -2$

i) $2 > -3.5$

j) $4.55 == 2.44$

k) $13 / 5$

l) $2 ^ 3$

m) $6 / 3$

n) $3.55 < 6.33$

o) $(6 < 8) \text{ ou } (3 > 7)$

p) $10 \% 6$

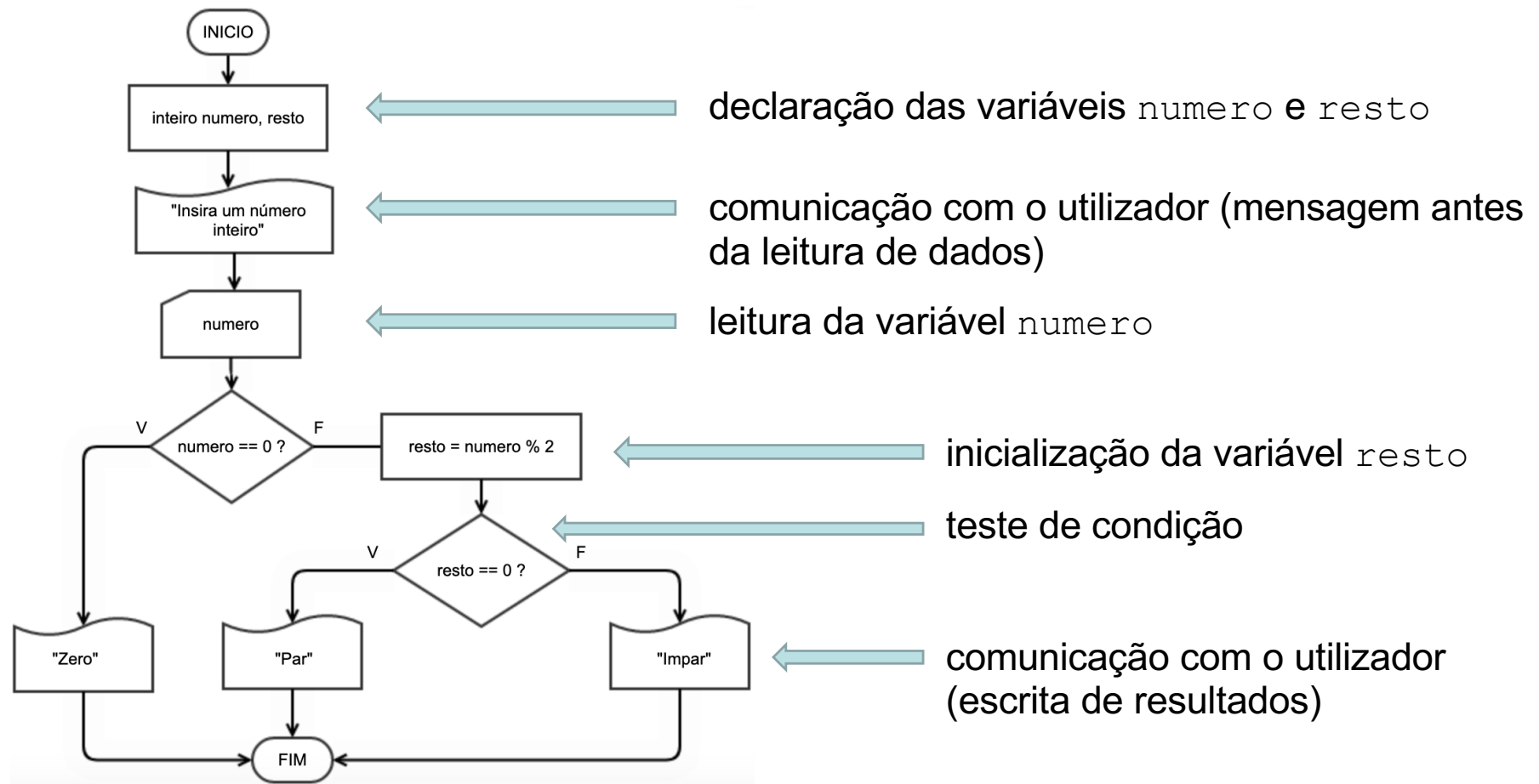
q) $4 \% 2$

r) $4 / 2$

s) $10 / 6$

t) $-2 ^ 3$

Todas as variáveis devem ser declaradas e inicializadas:



Todas as variáveis devem ser declaradas e inicializadas:

