

# Fundamentos de Programação

## Introdução a Fluxogramas

**Prof. Luiz Fernando Carvalho**

luizfcarvalho@utfpr.edu.br

# Algoritmos

- Definições:
  - Procedimento passo a passo para a solução de um problema
  - Sequência detalhada de ações a serem executadas para realizar alguma tarefa
  - Conjunto de regras para a solução de um problema
- O termo **algoritmo** não é restrito a computação
- Pode ser aplicado a qualquer circunstância que exija a descrição de uma solução



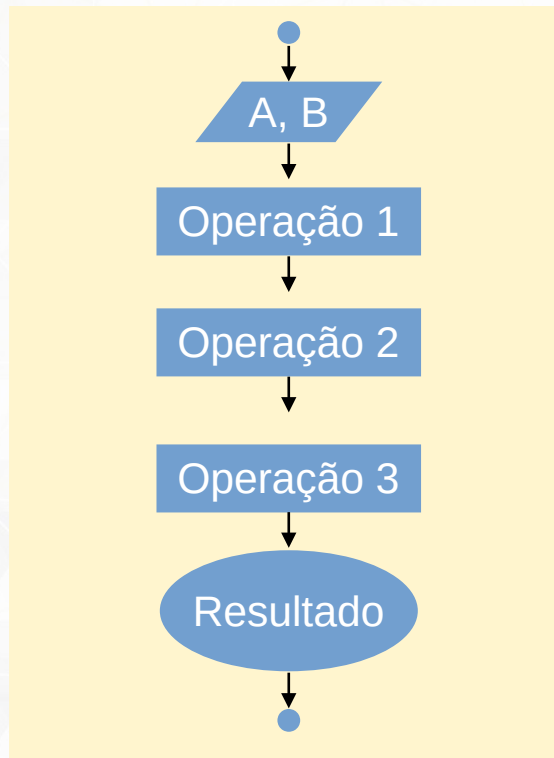


# Tipos de Representação de Algoritmos

- Os três tipos mais utilizados de algoritmos são a descrição narrativa, o fluxograma e o pseudocódigo (portugol)

1. Receber os dados
2. Executar operação 1
3. Executar operação 2
4. Executar operação 3
5. Mostrar o resultado

Narrativa



Fluxograma

```
Algoritmo Operacoes
Início
    Inteiro: A, B, resultado;
    Operacao 1;
    Operacao 2;
    Operacao 3;
    escreva(resultado);
Fim
```

Portugol

# Fluxograma

- O fluxograma consiste em analisar o enunciado do problema e escrever, utilizando símbolos gráficos predefinidos, os passos que devem ser seguidos para a resolução do problema

## Vantagem

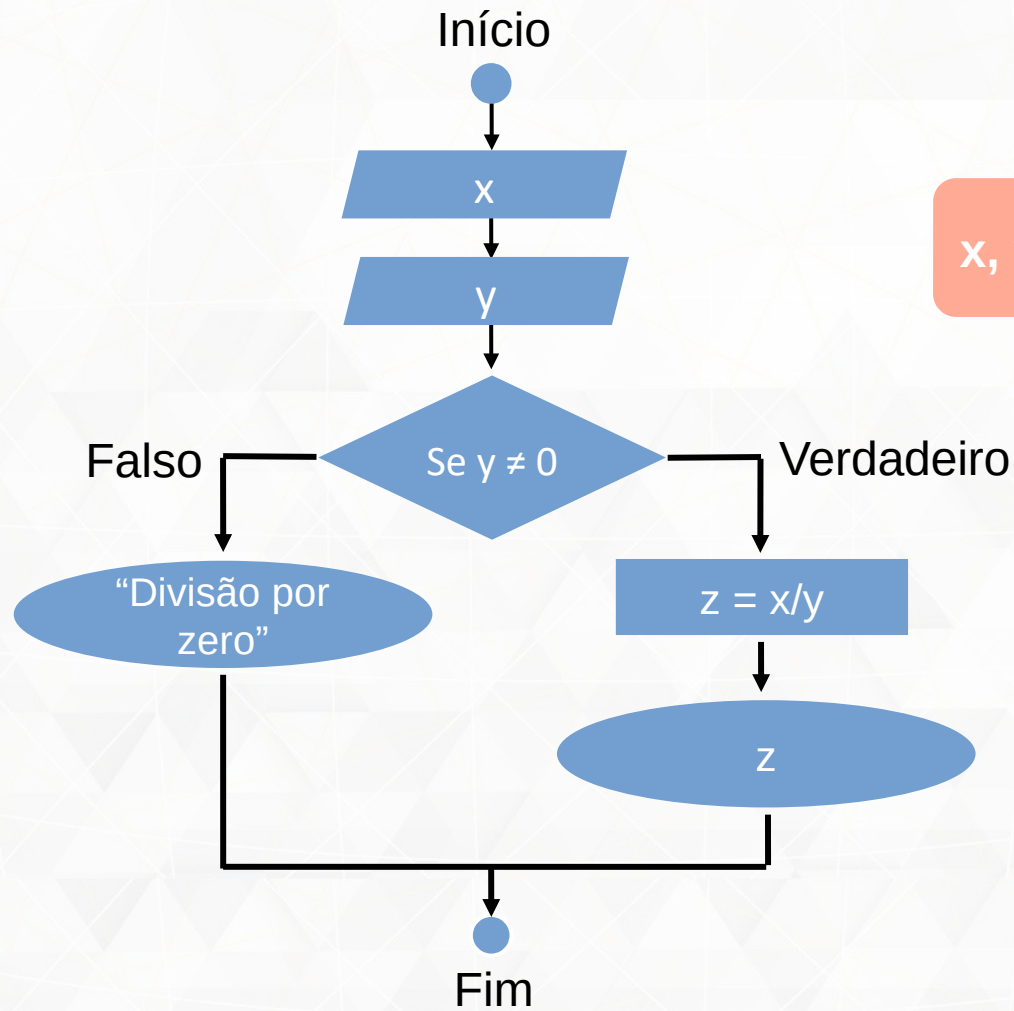
- A compreensão de elementos gráficos é mais fácil do que a de textos.
- Diminui a ambiguidade presente na descrição narrativa.

## Desvantagem

É necessário aprender a simbologia dos fluxogramas e, além disso, o algoritmo resultante não apresenta muitos detalhes, dificultando **(pouco)** a sua transcrição para um programa.



# Fluxograma



x, y e z são variáveis

# Variáveis

- Uma variável é uma localização na memória do computador que é utilizada para **armazenar temporariamente** os dados que são **utilizados pelo programa**
- As variáveis possuem algumas características, como:
  - Identificação (nome da variável);
  - Tipo;
  - Endereço;
  - Tamanho;
  - Valor (conteúdo da variável);

# Variáveis



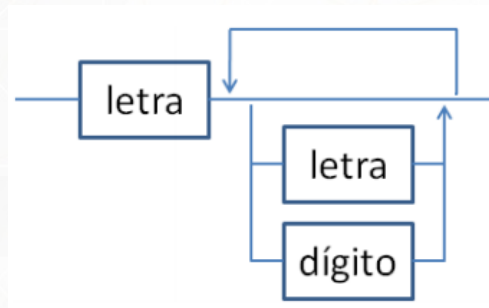
Cada espaço de memória tem um endereço

Similar à memória do computador



# Variáveis

## Tipo Identificador



Identificadores (nomes):

- 1.Devem começar com uma **letra**;
- 2.Os próximos caracteres podem ser **letras** ou **números**;
- 3.Não pode utilizar nenhum símbolo, exceto ***underline***;
- 4.Não pode conter **espaços em branco**;
- 5.Não pode conter **letras com acentos**;

# Variáveis

Tipos:

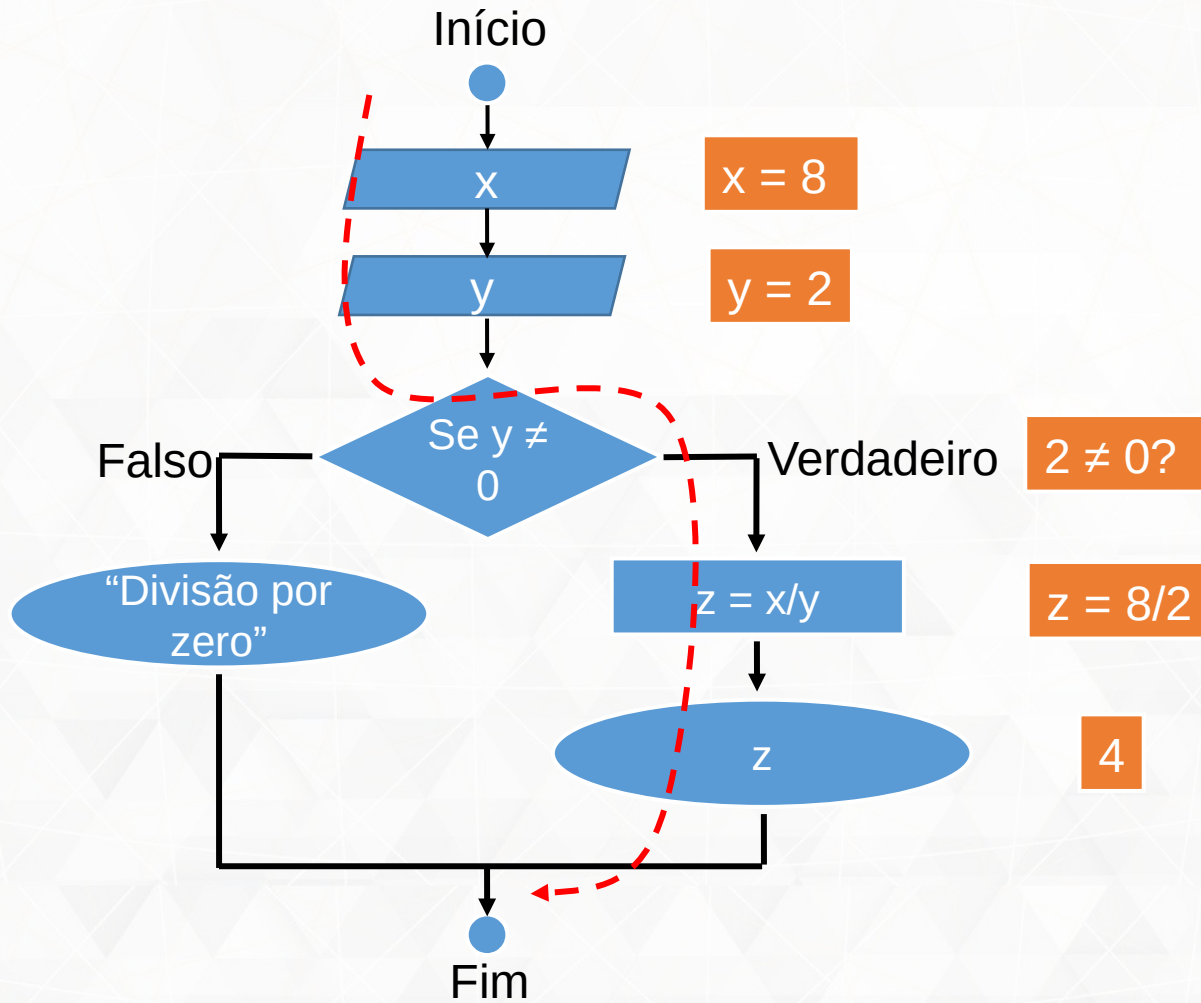
Inteiro (**int**): **1 3 -5 198 0**

Real (**float** – ponto flutuante): **0.5 5.0 9.8 -77.3 3.1415**

Caractere (**char** - character): **“programa” ‘c’ “algoritmo” “123”**

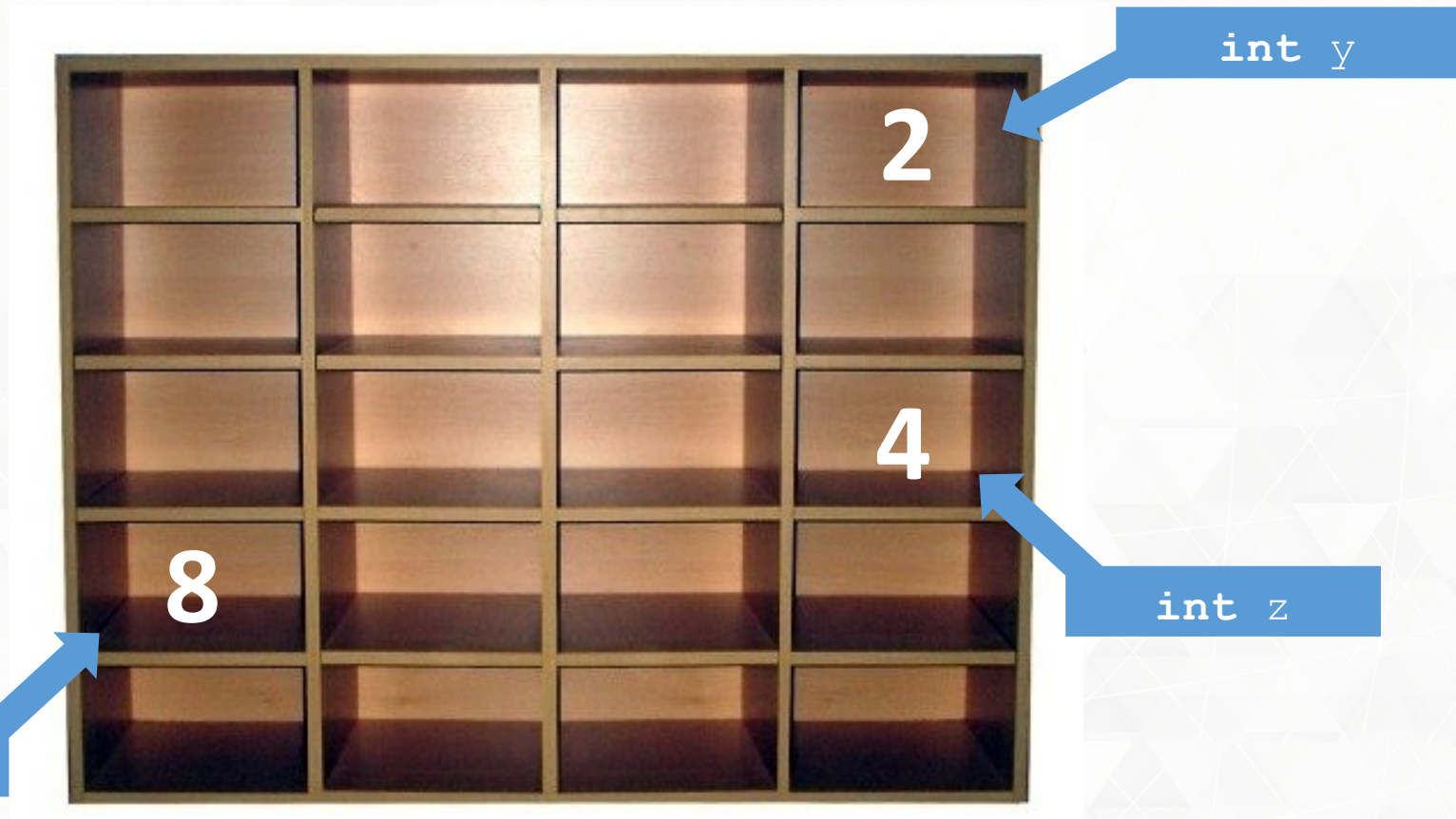
Lógico (**bool** - booleano): **Verdadeiro Falso**

# Fluxograma

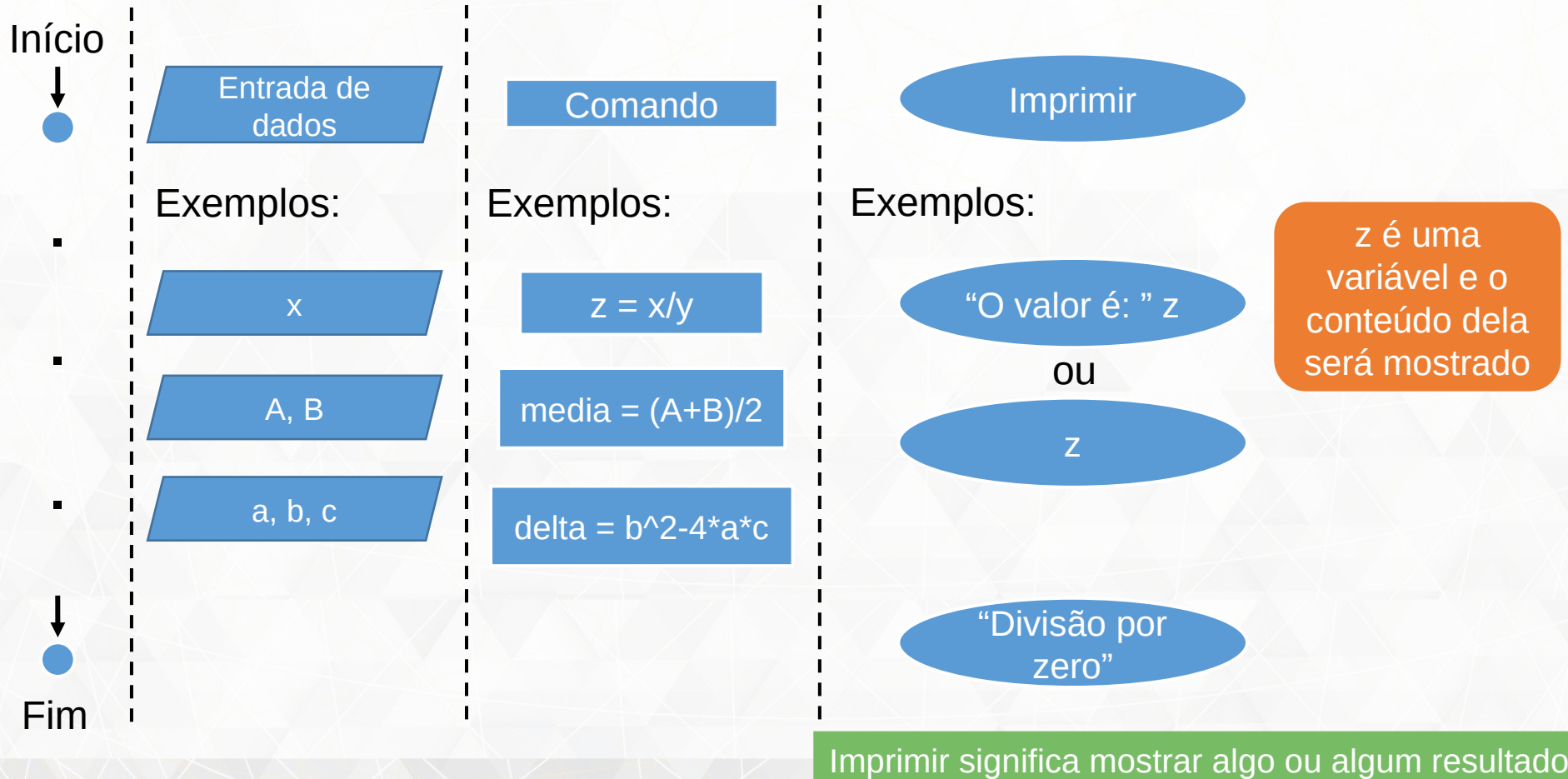




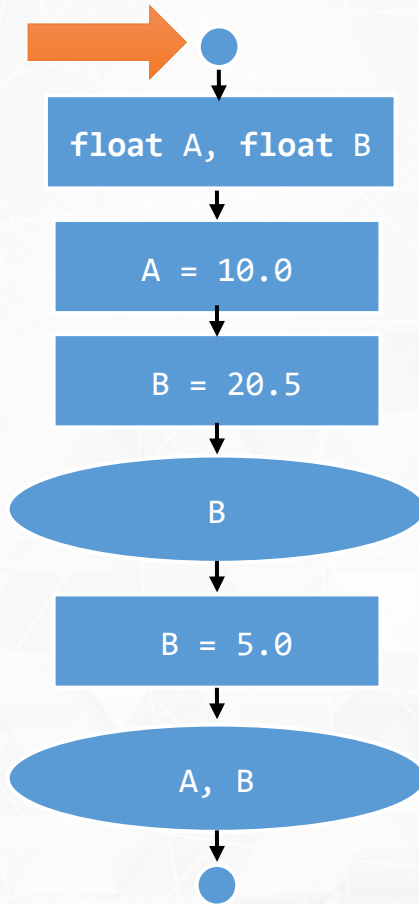
# Variáveis



# Fluxograma



# Funcionamento do computador

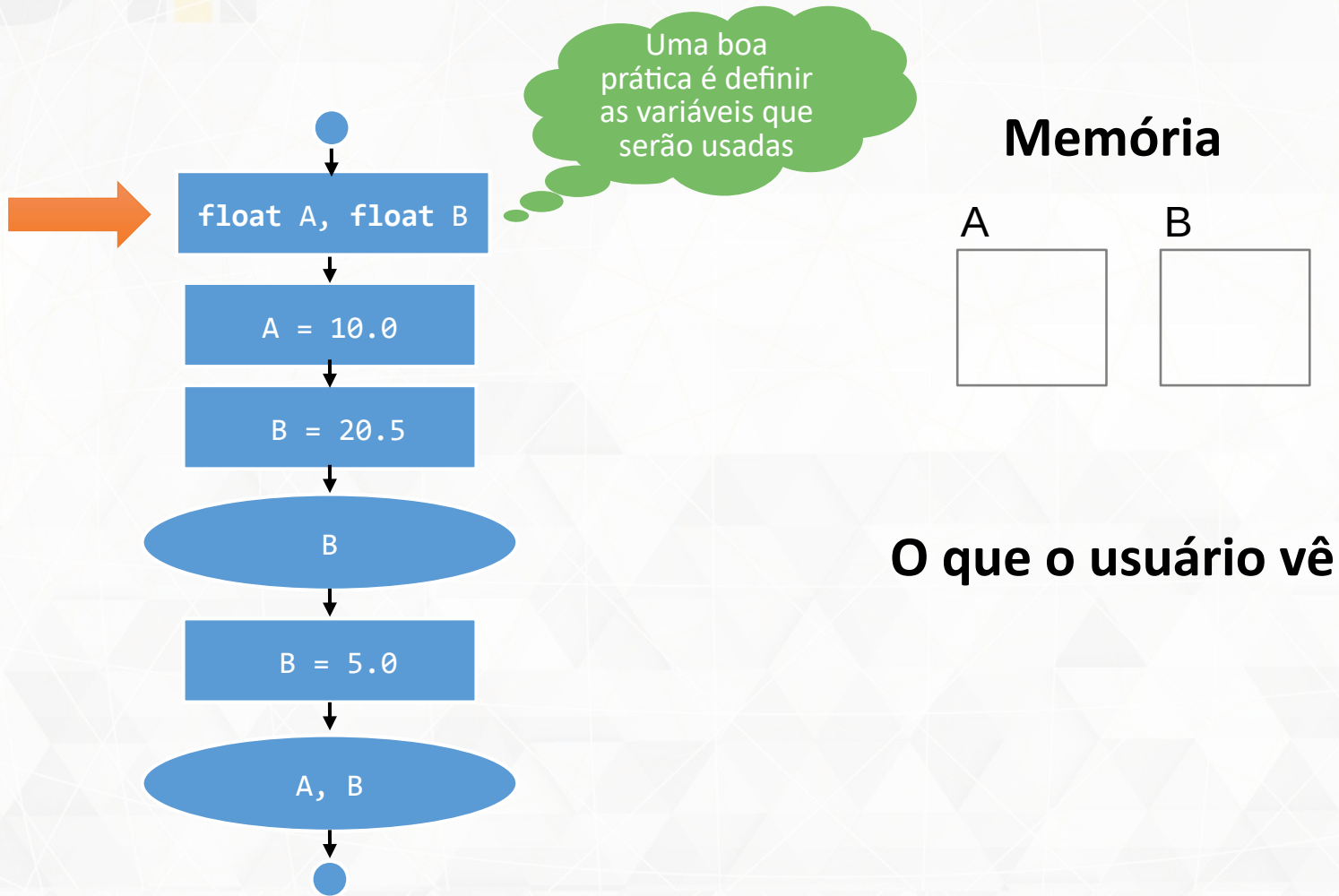


**Memória**

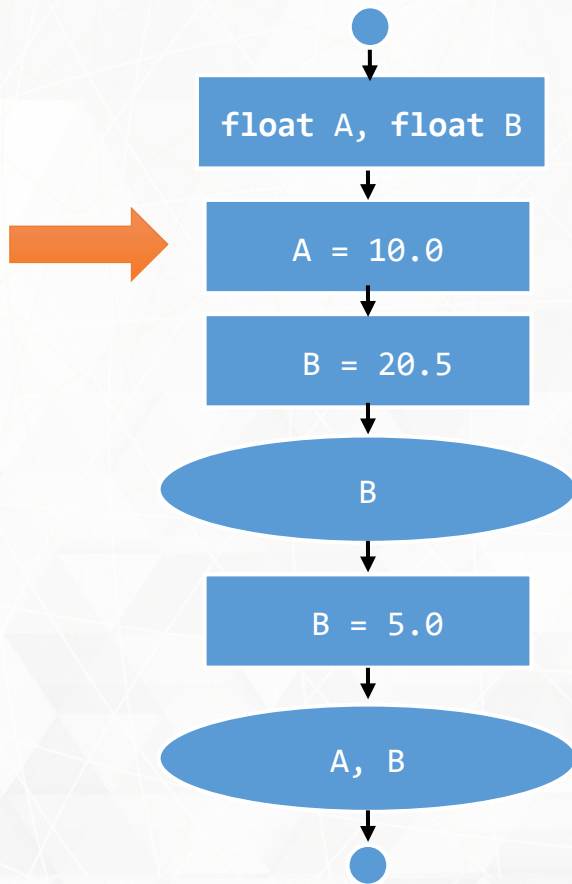
**O que o usuário vê**



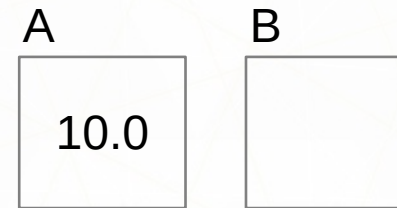
# Funcionamento do computador



# Funcionamento do computador

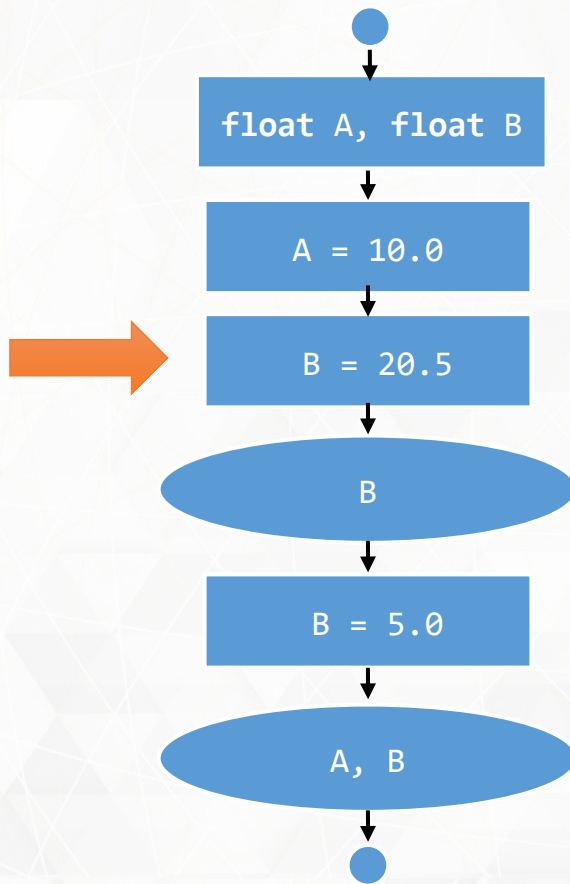


## Memória



O que o usuário vê

# Funcionamento do computador



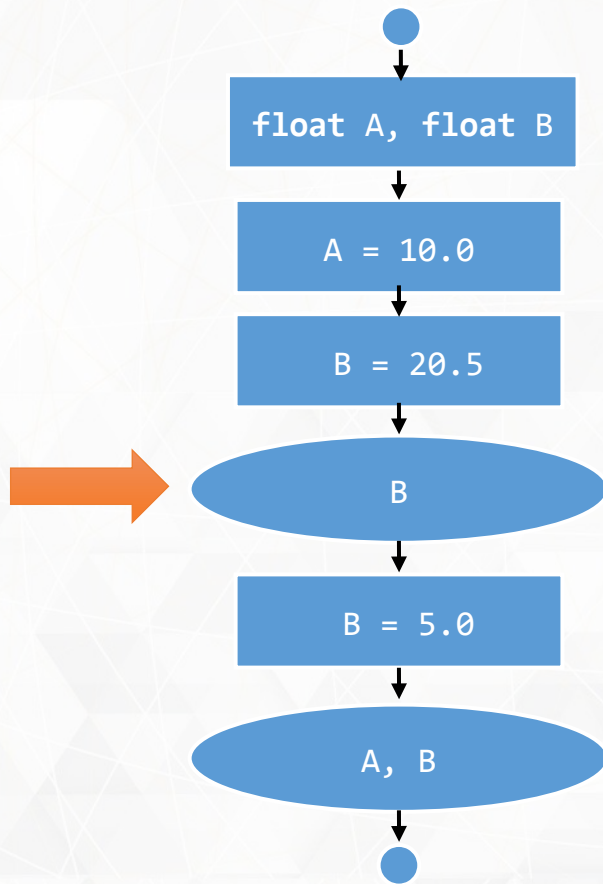
## Memória

A	B
10.0	20.5

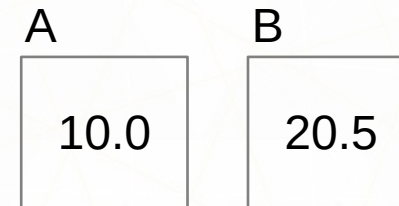
O que o usuário vê



# Funcionamento do computador



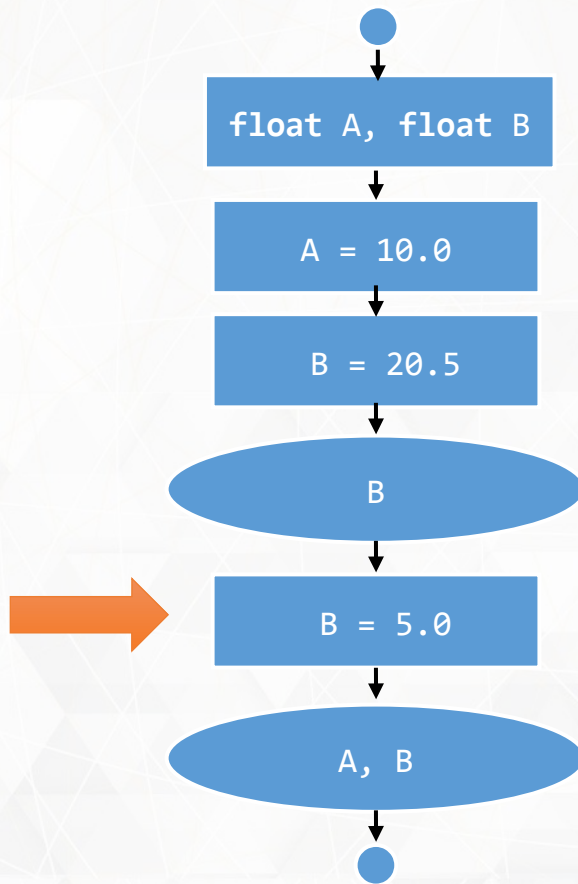
## Memória



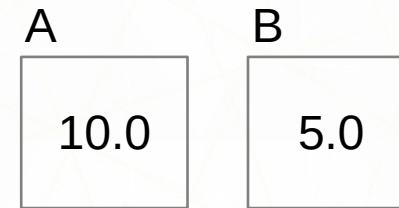
## O que o usuário vê

20.5

# Funcionamento do computador



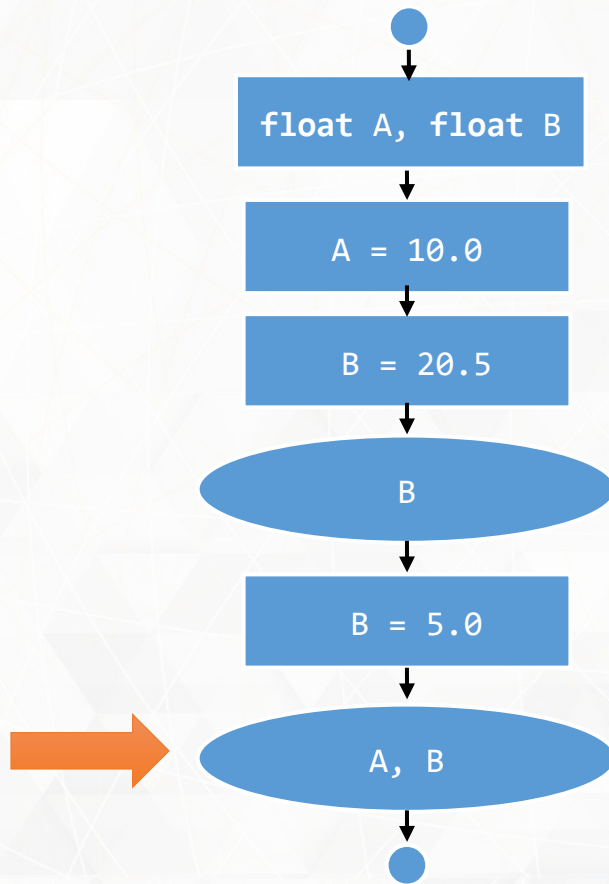
## Memória



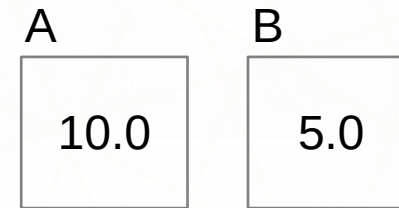
## O que o usuário vê

20.5

# Funcionamento do computador



## Memória



## O que o usuário vê

20.5  
10.0 5.0



# Exercícios

- O que será impresso na tela para o usuário?

a)

```
A = 30
B = 20
C = A + B
Imprimir C
B = 10
Imprimir B, C
C = A + B
Imprimir A, B, C
```

```
50
10 50
30 10 40
```

b)

```
A = 10
B = 20
C = A
B = C
A = B
Imprimir A, B, C
```

```
10 10 10
```

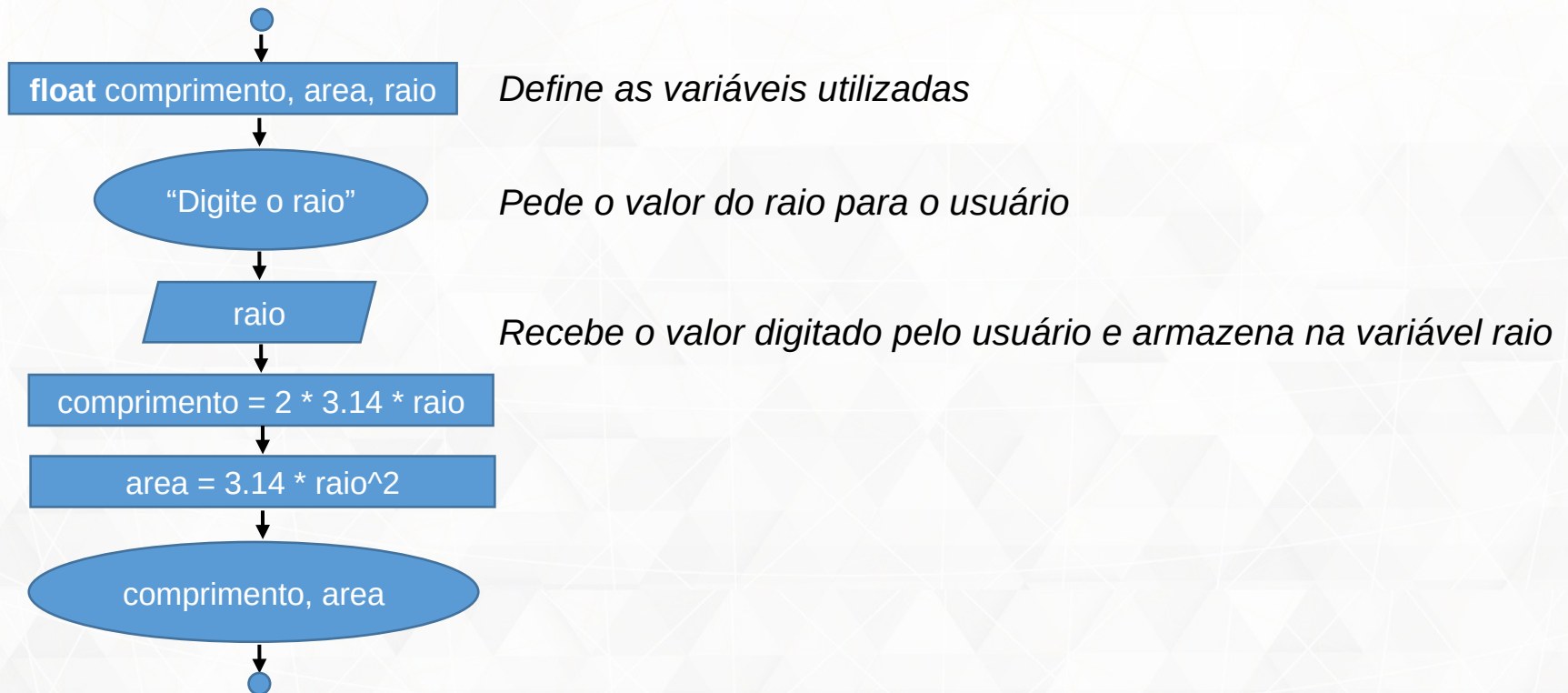
c)

```
A = 10
B = A + 1
A = B + 1
B = A + 1
Imprimir A
A = B + 1
Imprimir A, B
```

```
12
14 13
```

# Exemplo

- Perguntando o raio para o usuário e calculando a área do círculo e comprimento da circunferência



# Operadores Aritméticos

Ordem de  
precedência



Operador	Operação
( )	Parênteses
	Exponenciação
%	Módulo
/	Divisão
*	Multiplicação
+	Soma
-	Subtração

O operador % retorna o resto da divisão entre dois operandos

$$4 \% 2 = 0$$

$$5 \% 2 = 1$$

$$17 \% 3 = 2$$

$$100 \% 50 = 0$$

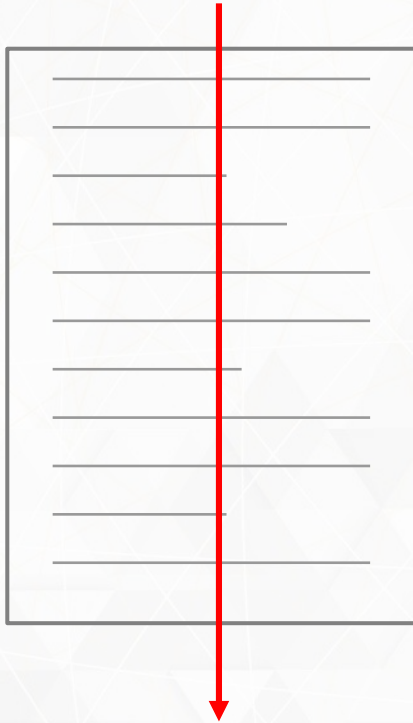
Operando é um elemento (valor ou variável, por exemplo) que participa de uma operação



# Estrutura de Controle

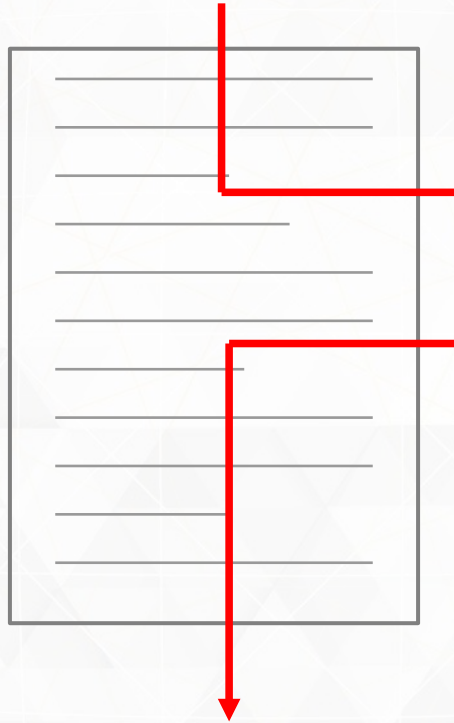
- Programa de computador
  - Conjunto de instruções organizadas de forma a produzir a solução de um determinado problema.
- Fluxo de execução
  - Começa na primeira linha e avança sequencialmente
    - de cima para baixo;
  - Em muitas circunstâncias é necessário executar instruções em uma ordem diferente;
  - Necessidade de decisão entre fluxos alternativos de execução ou da repetição de determinadas instruções
  - Pode haver bifurcações, repetição de código e tomada de decisão

# Estrutura de Controle



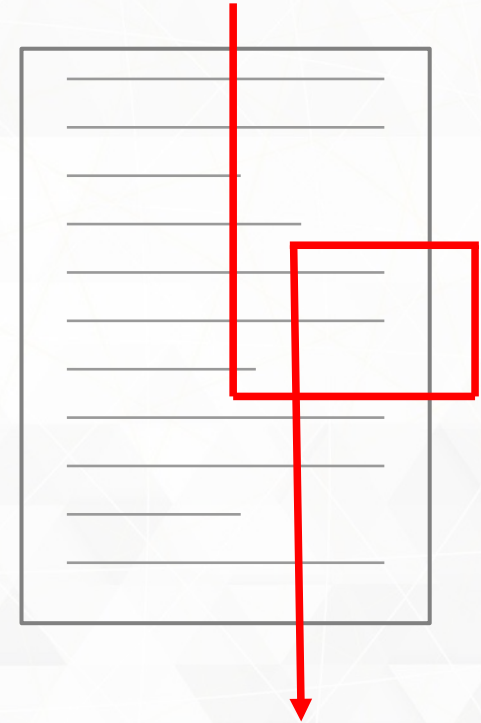
Fluxo de execução  
sequencial

*Comandos são executados  
um após o outro*



Fluxo de execução  
com desvio

*Comandos são executados  
dependendo do valor de uma  
condição*



Fluxo de execução  
repetitivo

*Comandos são executados de  
forma repetida*

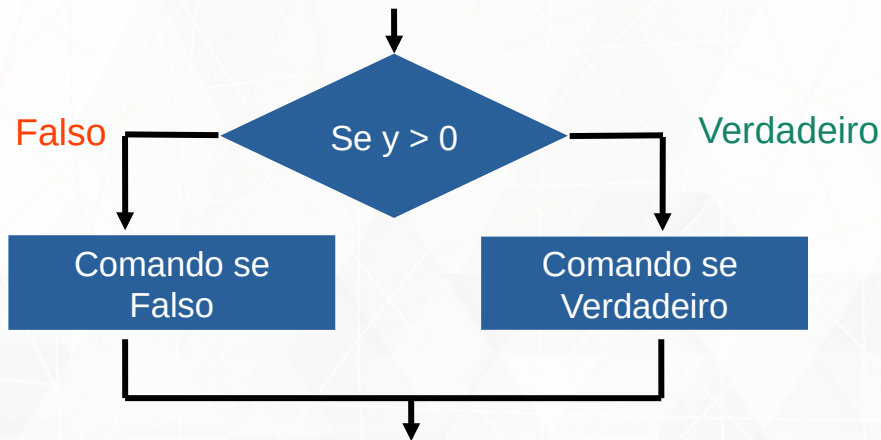
# Estrutura de Controle

- As estruturas de controle dividem-se em:
  - Estruturas de seleção;
  - Estruturas de repetição (*loops* de repetição).
- As estruturas de controle estão vinculadas às condições que determinam se instruções serão ou não executadas
- Uma condição de controle está relacionada aos operadores relacionais e lógicos



# Estrutura de Controle

## Seleção

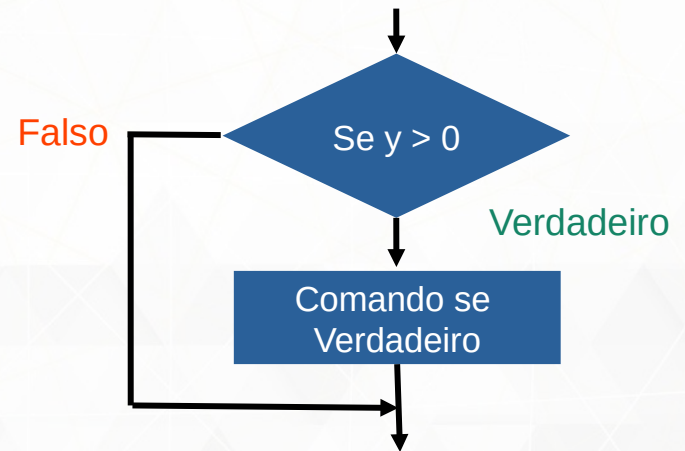


**Se y é maior que zero**

Executa Comando se **Verdadeiro**

**Senão**

Executa Comando se **Falso**



**Se y é maior que zero**

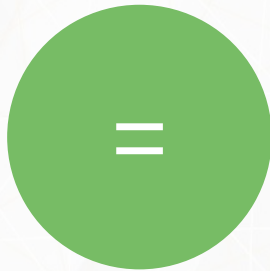
Executa Comando se **Verdadeiro**

# Operadores relacionais

- Os operadores relacionais comparam dois valores e retornam um valor booleano (**verdadeiro** ou **falso**);
- Usados em estruturas condicionais e laços de repetição

Operador	Descrição	X	Y	Lógico	Resultado
==	Igual a	2	3	$X == Y$	Falso
!=	Diferente de	2	3	$X != Y$	Verdadeiro
>	Maior que	2	3	$X > Y$	Falso
>=	Maior ou igual	2	3	$X >= Y$	Falso
<	Menor que	2	3	$X < Y$	Verdadeiro
<=	Menor ou igual	2	3	$X <= Y$	Verdadeiro

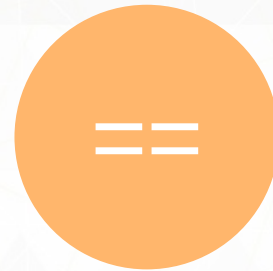
# Operador de atribuição



Operador de atribuição

`X = 5`

X recebe o valor 5



Operador relacional

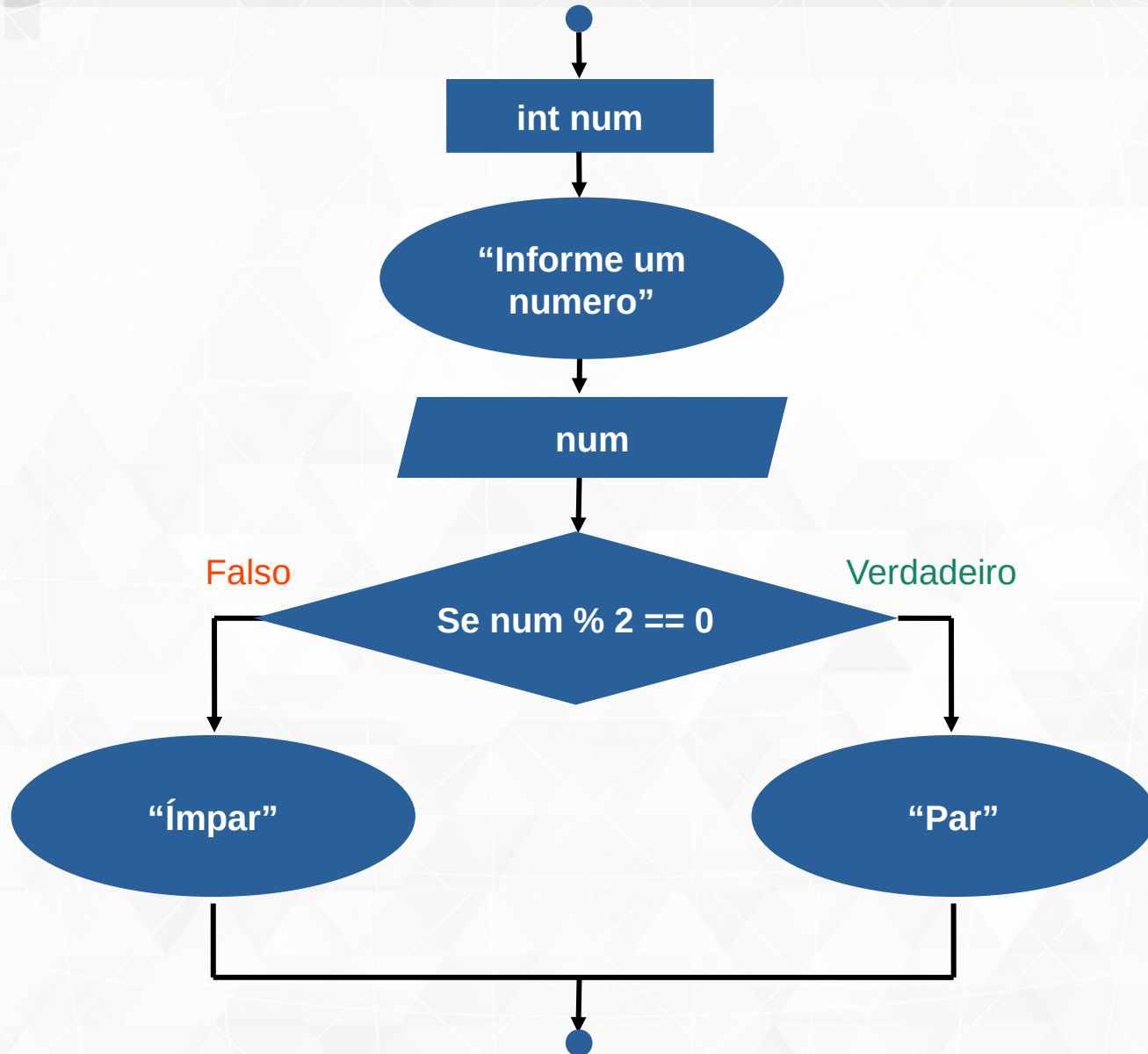
`X == 5`

O valor armazenado em X  
é igual a 5?

É um erro de programação utilizar = quando se deve utilizar == (e vice-versa)



# Exemplo – Par ou ímpar?



# Exercícios

1. Faça um programa que leia dois números quaisquer e imprima o maior deles
2. Faça um programa que receba duas notas de um aluno, calcule e mostre a média aritmética das notas e uma mensagem conforme a tabela a seguir

Média	Mensagem
De 0.0 a 3.9	Reprovado
De 4.0 a 5.9	Exame
De 6.0 a 10.0	Aprovado

3. Faça um programa que leia dois números quaisquer, e imprima o resultado do cálculo do maior dividido pelo menor.
4. Refaça o exercício anterior, mas agora imprima uma mensagem de erro caso puder ocorrer uma divisão por zero

# Exercícios

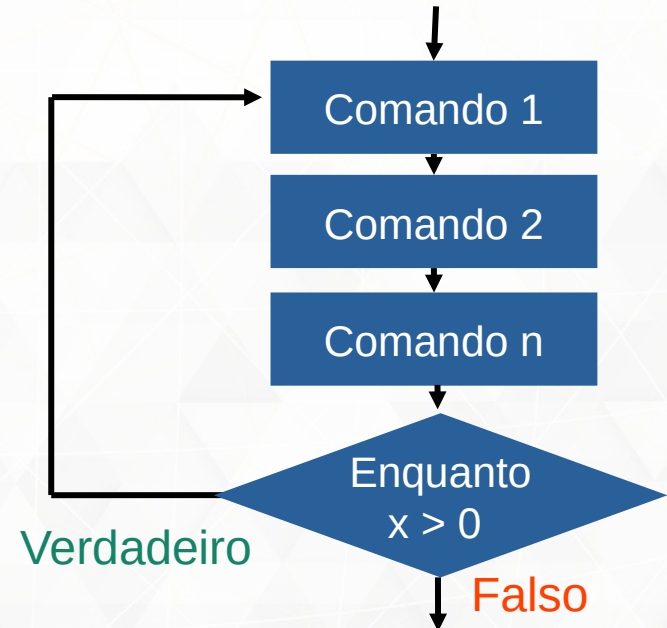
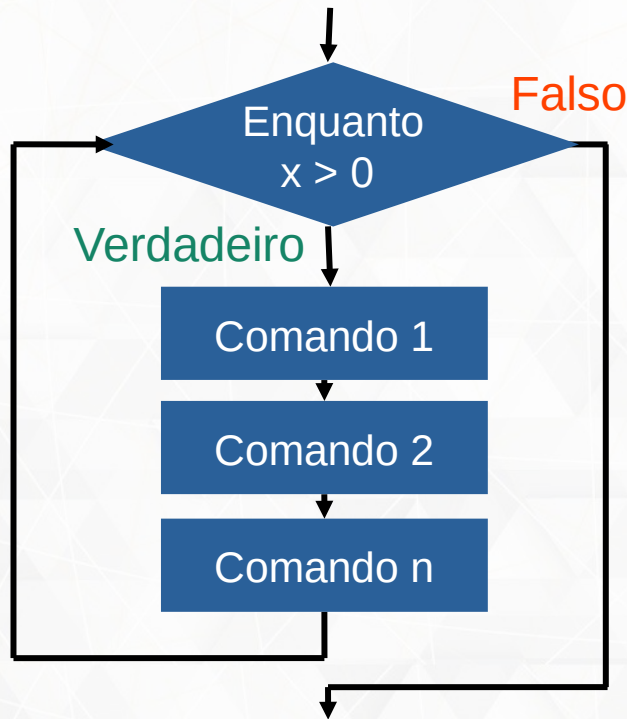
5. Desenvolver a lógica para um programa que efetue o cálculo do reajuste de salário de um funcionário
  - Salário  $\leq 500$ , reajuste de 15%
  - Salário  $> 500$ , mas Salário  $\leq 1000$ , reajuste será de 10%
  - Salário  $> 1000$ , reajuste será de 5%
  
6. Faça um algoritmo que leia os valores A, B, C e imprima o resultado somente se a soma de  $A + B$  é menor que C. Caso não seja, não faça nada e encerre o algoritmo



# Estrutura de Controle

## Laço de Repetição

Enquanto a condição é satisfeita, execute o comando



NUNCA executa o verdadeiro e falso simultaneamente

# Variáveis: acumulador e contador

- **ACUMULADOR (SOMADOR):** É uma variável que atua acumulando os valores a cada vez que o código é executado. Por exemplo, poderíamos implementar um somador num caixa de supermercado, acumulando na variável total todas as compras. Essa implementação é feita fazendo com que a variável total receba o seu próprio valor + o valor parcial de cada execução.
  - Ex.: `total = total + valor`
- **CONTADOR:** Os contadores acumulam seu próprio valor, acrescentando 1 a cada execução do programa. No mesmo exemplo anterior, o `total_de_itens` receberia seu próprio valor + 1 a cada item que passasse pelo caixa.
  - Ex.: `total_de_itens = total_de_itens + 1`

# Exercícios

7. Faça um algoritmo que leia 10 valores e escreva no final a soma dos valores lidos
8. Faça um algoritmo que receba várias idades e que calcule e mostre a média das idades fornecidas. Finalize o algoritmo quando a idade igual a zero for passada como entrada
9. Faça um algoritmo que imprima os números pares de 0 até 20
10. Imprima a tabuada do 7
11. Faça um programa que receba a idade de dez pessoas, calcule e mostre a quantidade de pessoas com idade maior ou igual a 18 anos
12. Faça um programa que leia números até quando o número zero for informado. Ao final do programa, o algoritmo deve apresentar o maior e o menor número informado



# Exercícios

13. João tem atualmente 1,10 m de altura enquanto Maria tem 1,30. João cresce 3 cm ao ano e Maria 1 cm ao ano. Faça um fluxograma que apresente a altura de ambos a cada ano até o momento em que João se tornar maior que Maria. Ao final do algoritmo deverá ser exibido quantos anos foram necessários para que João fosse maior que Maria

# Operadores lógicos

- Trabalham com valores booleanos e seu resultado também é booleano (verdadeiro ou falso)
- Eles são usados somente em expressões lógicas
- Um conjunto especial de operadores é necessário para combinar condições simples, criando condições complexas: operadores lógicos.

Operador	Significado	Equivalente
!	Negação	Não
&&	Conjunção	E
	Disjunção	OU

# Operadores lógico

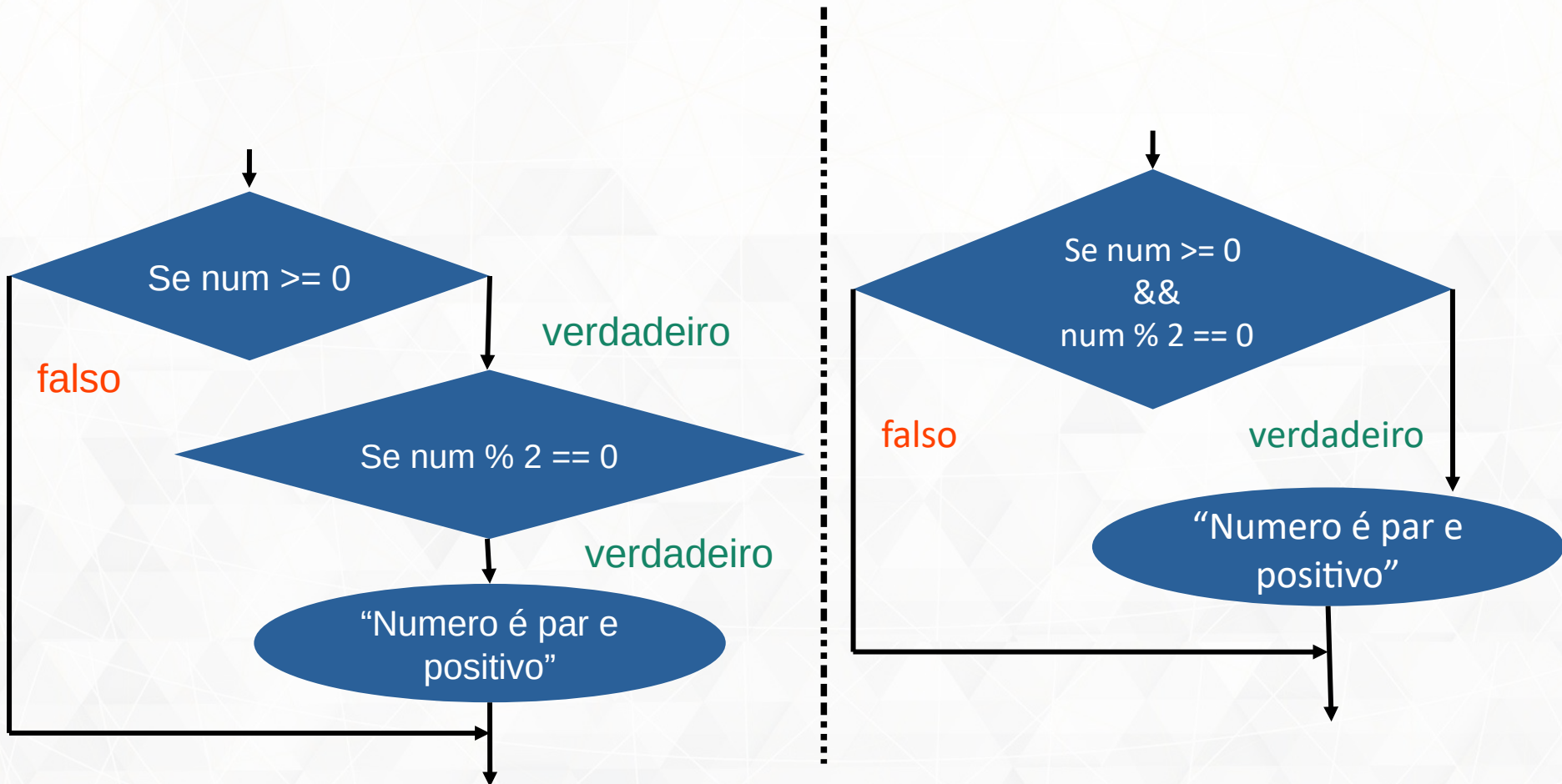
- Sejam A e B duas variáveis booleanas (bool), que assume valores:
  - Verdadeiro (V) ou
  - Falso (F)

A	B	!B	A && B	A    B
V	V	F	V	V
V	F	V	F	V
F	V	F	F	V
F	F	V	F	F




# Operadores lógico

- Se quisermos testar se um número num é positivo e par



# Ordem de precedência



<b>Aritmético</b>	( )
	^
	* / %
	+ -
<b>Relacional</b>	> < != == >= <=
<b>Lógico</b>	&&
	!

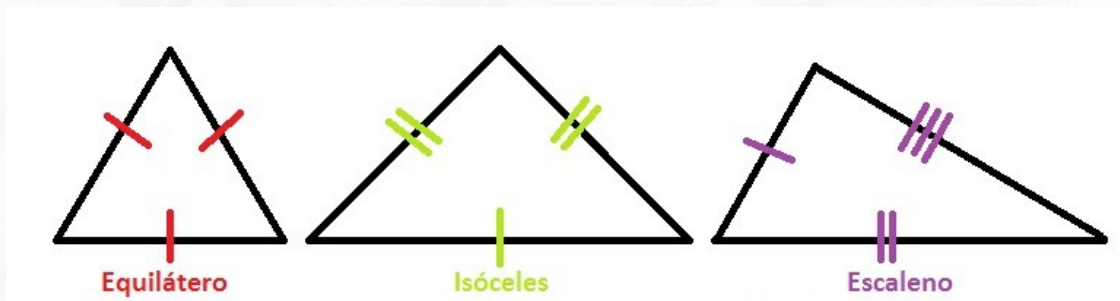
# Exercícios

14. Sabendo que  $A = 5$ ,  $B = 4$ ,  $C = 3$  e  $D = 6$ , informe se as expressões abaixo são verdadeiras ou falsas

- $(A > C) \ \&\& \ (C \leq D)$
- $(A + B) > 10 \ || \ (A + B) == (C + D)$
- $(A \geq C) \ \&\& \ (D \geq C)$

15. Leia 3 valores ( $L1$ ,  $L2$  e  $L3$ ) que correspondem aos lados de um triângulo. Verifique se o triângulo é:

- Equilátero (3 lados com mesma medida);
- Escaleno (nenhum dos lados tem a mesma medida)





# Exercícios

16. Elaborar um algoritmo que efetue a leitura de um valor que esteja entre a faixa de 1 a 9. Após a leitura do valor fornecido pelo usuário, o algoritmo deverá indicar uma de duas mensagens: “O valor está na faixa permitida”, caso o usuário forneça o valor nesta faixa, ou a mensagem “O valor está fora da faixa permitida”, caso o usuário forneça valores menores que 1 ou maiores que 9.
17. Fazer um algoritmo para ler o ano de nascimento de uma pessoa, calcular e mostrar sua idade e, também, verificar e mostrar se ela já tem idade para votar (16 anos ou mais) e para conseguir a Carteira de Habilitação (18 anos ou mais).
18. Faça um algoritmo que receba 10 números e que calcule e mostre a quantidade de números entre 30 e 90.

# Exercícios

19. Faça um algoritmo que receba 10 idades, pesos e alturas e que calcule e mostre:

- A média das idades das pessoas;
- A quantidade de pessoas com peso superior a 90kg e altura inferior a 1,5m.
- A porcentagem de pessoas com idade entre 10 e 30 anos que medem mais de 1,8m.

20. Faça um algoritmo que receba a idade e o sexo de sete pessoas e que calcule e mostre: a idade média do grupo, a idade média das mulheres, a idade média dos homens.