

Técnicas de Programação A

Luiz Fernando Carvalho

luzfcarvalhoo@gmail.com

Sumário

- Fluxograma para representação de algoritmos;
- Variáveis;
- Estruturas de controle
 - Desvio condicionais “se”;
 - Laços de repetição “enquanto”;
- Operadores e ordem de precedência
 - Aritméticos;
 - Relacionais;
 - Lógicos;
- Exercícios.

Algoritmos Computacionais

Lógica de
Programação



Linguagem de Programação



Aplicação/Programa



Programador
formula a 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;
 imprima(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.

O PROBLEMA DE SER PROGRAMADOR



MINHA MULHER DISSE:

**– AMOR, VÁ ATÉ O MERCADO E COMPRE 1 GARRAFA DE LEITE,
SE ELES TIVEREM OVOS, TRAGA 6**

~ EU VOLTEI PARA CASA COM 6 GARRAFAS DE LEITE ~

ELA DISSE:

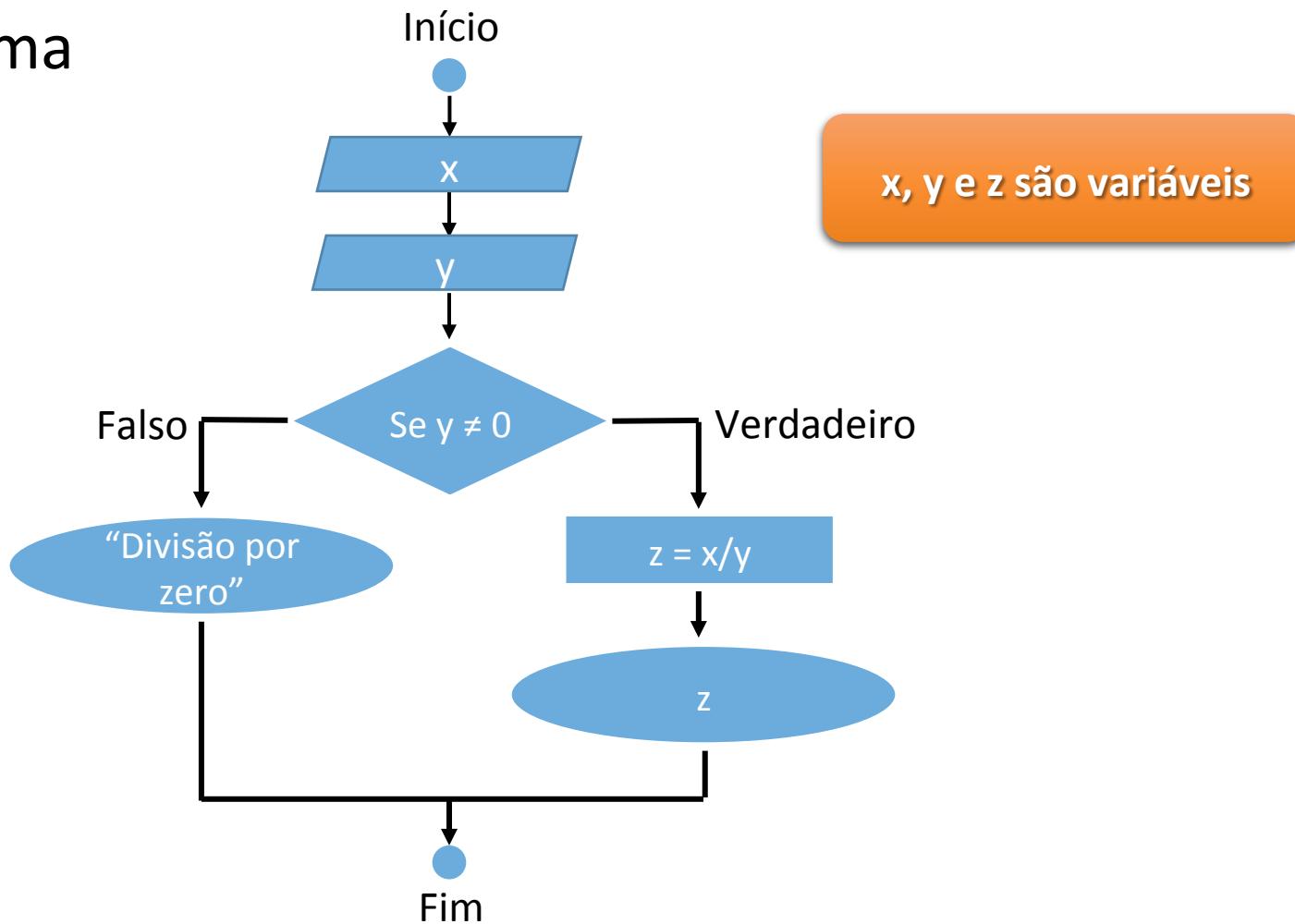
– PORQUE DIABOS VOCÊ COMPROU 6 GARRAFAS DE LEITE?

EU RESPONDI:

– PORQUE ELES TINHAM OVOS.

Lógica de Programação

- Fluxograma



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);

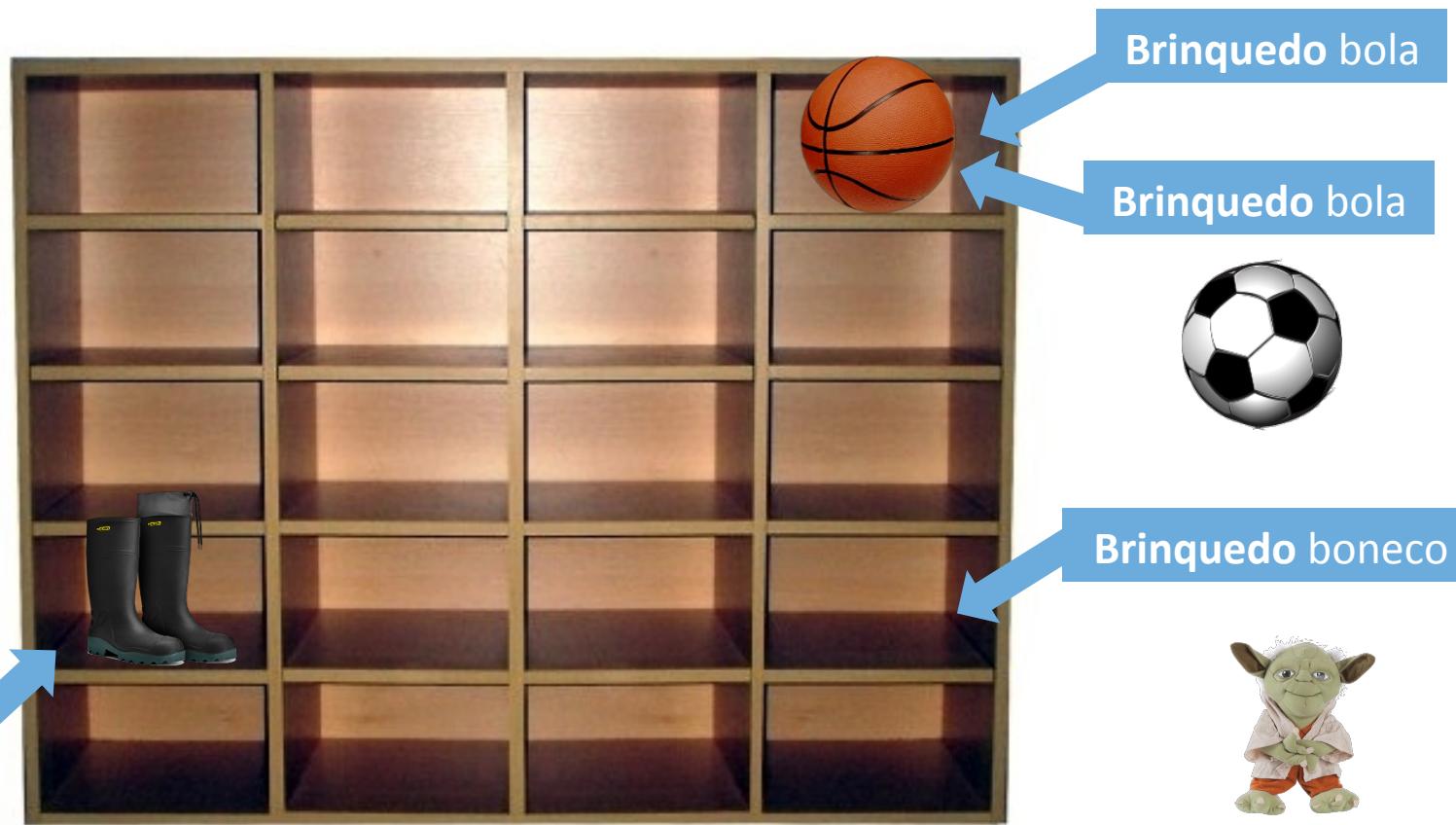
Variáveis



- Cada espaço de memória tem um endereço

Similar à memória do computador

Variáveis



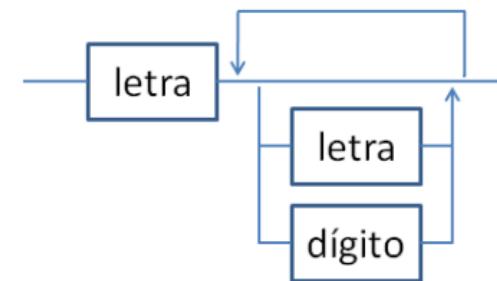
Variáveis

Tipo Identificador

Calçado botas



Brinquedo boneco



Identificadores:

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

Tipo Identificador

Calçado botas



Brinquedo boneco



Tipos:

Inteiro (int): ex.: 1 3 -5 198 0

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

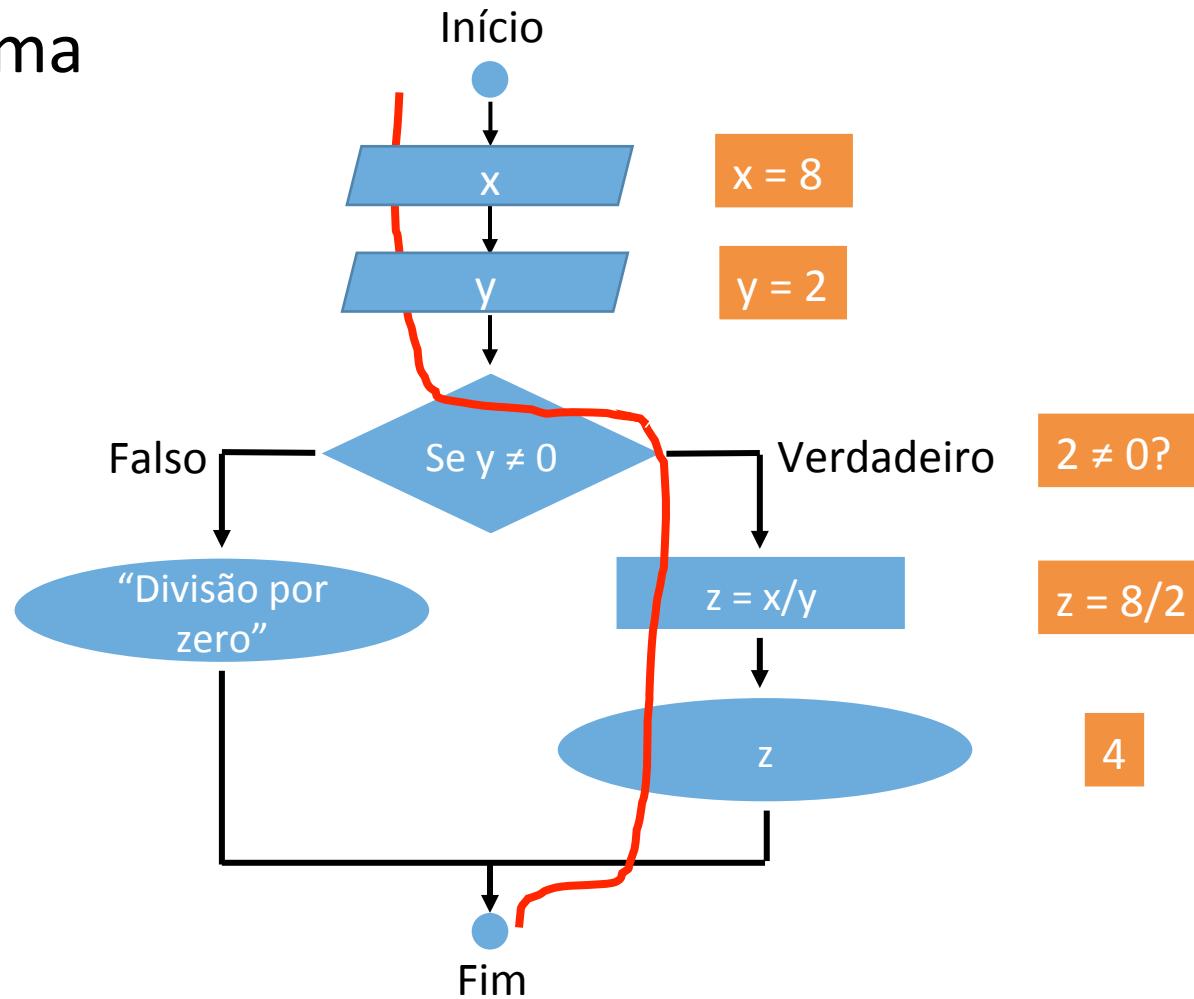
Caractere (char - character): ex.: “programa” ‘c’ “algoritmo”
“123”

Lógico (bool - booleano): ex.: true false

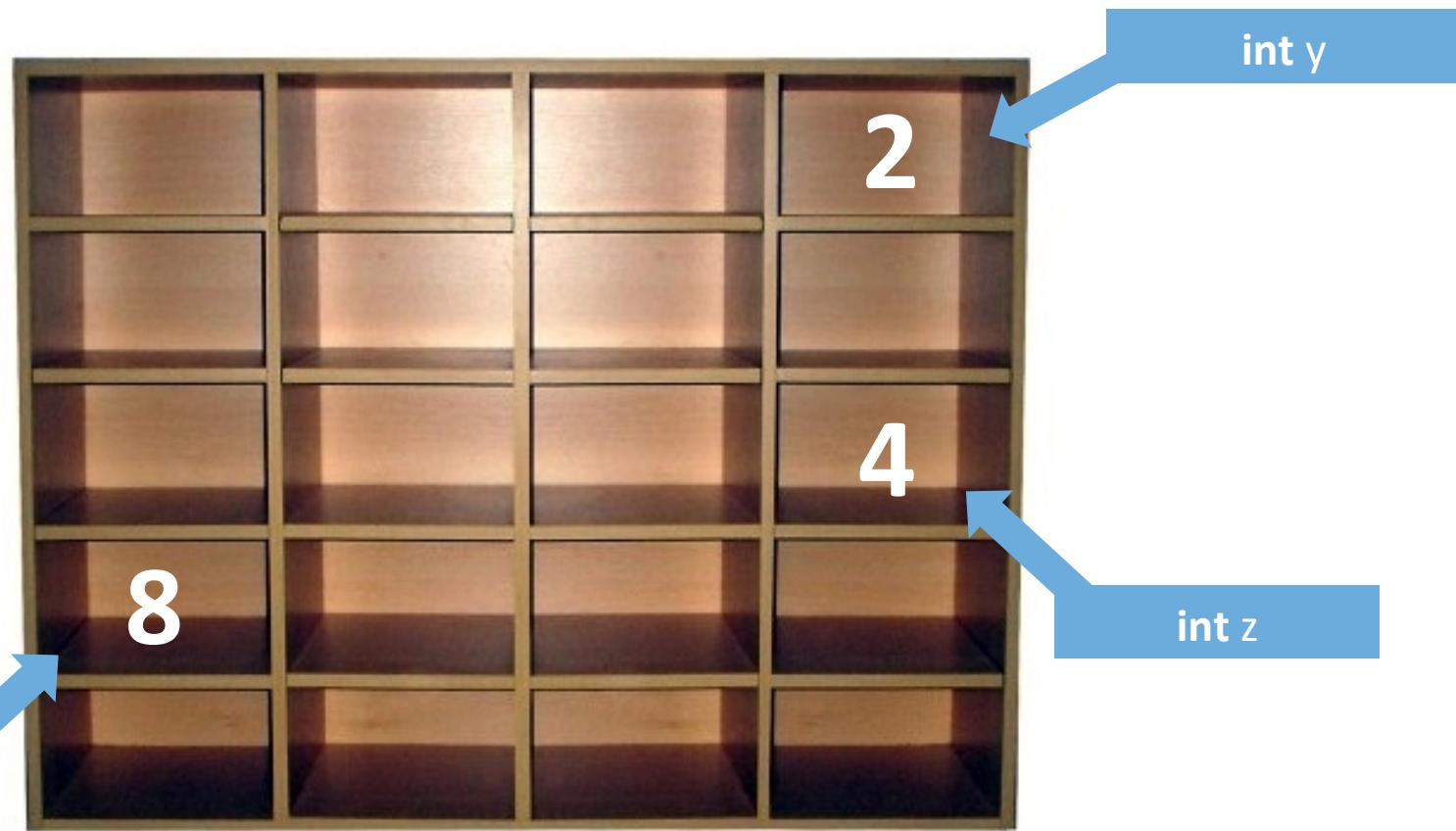
Caracteres simples usam aspas simples

Lógica de Programação

- Fluxograma

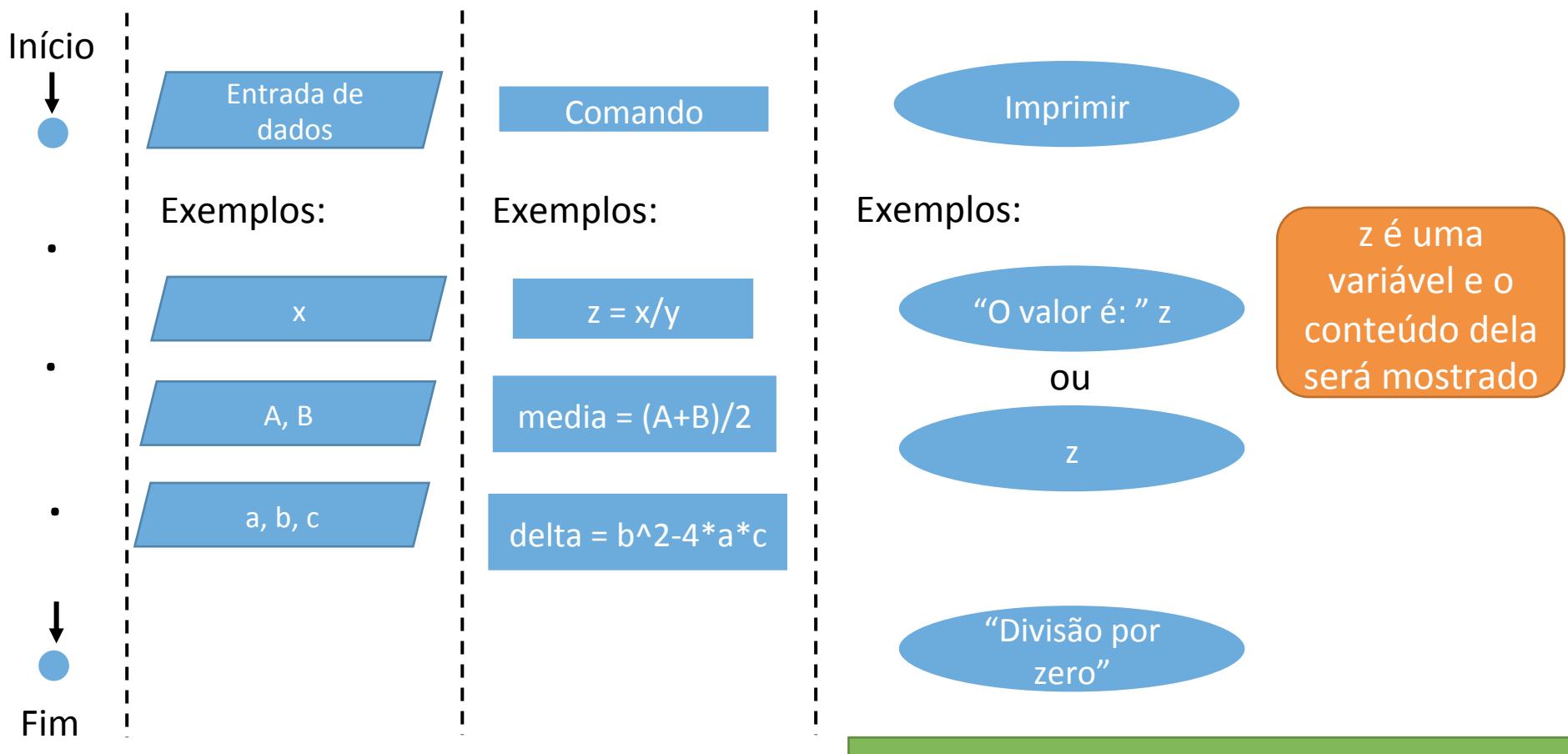


Lógica de Programação

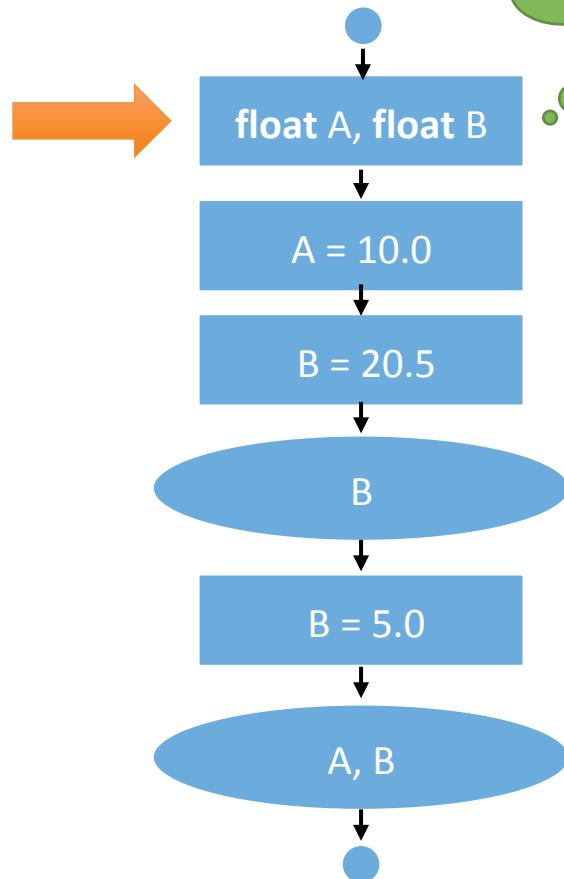


Lógica de Programação

- Fluxograma

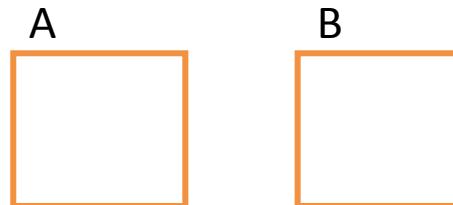


Exercícios



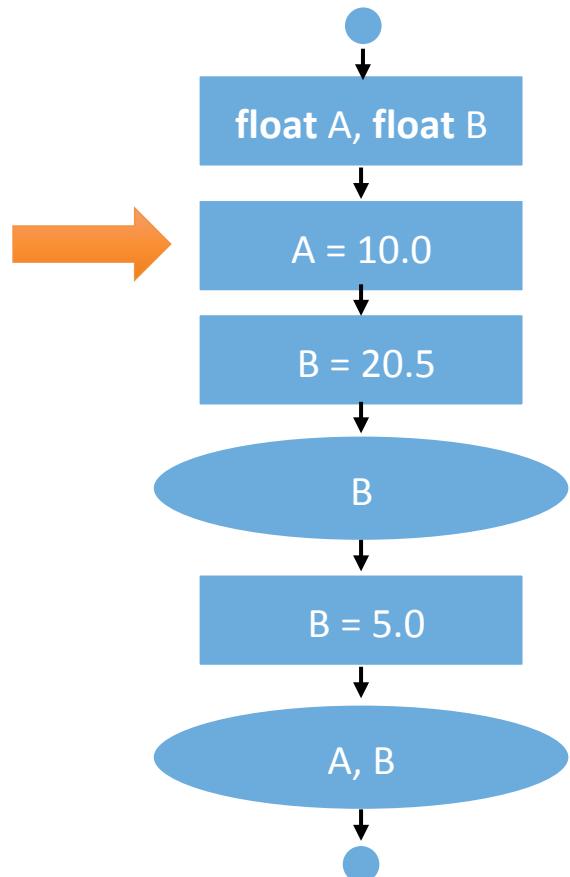
Uma boa prática é definir as variáveis que serão usadas

Memória

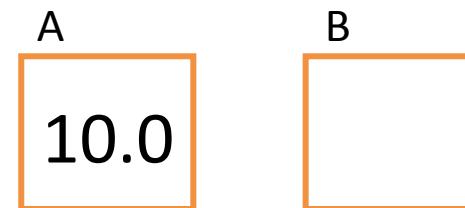


O que o usuário vê

Exercícios

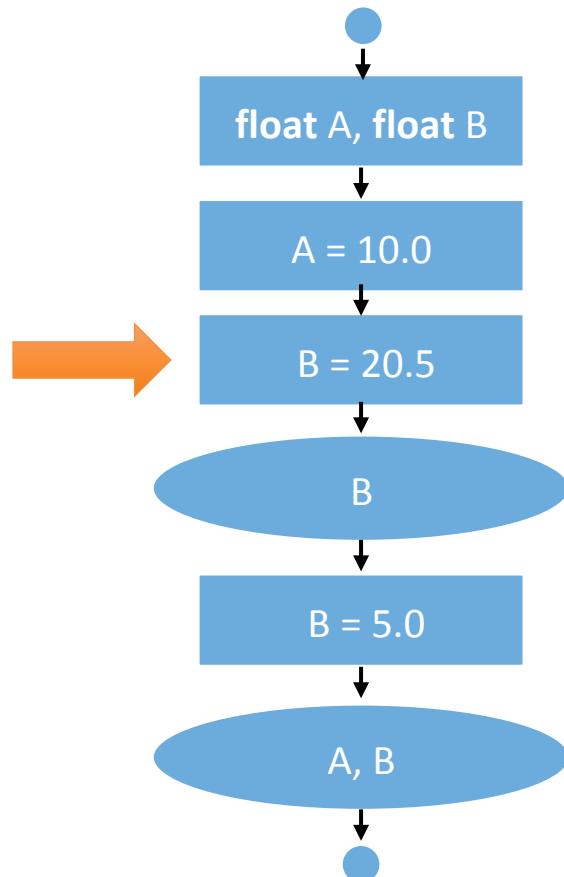


Memória

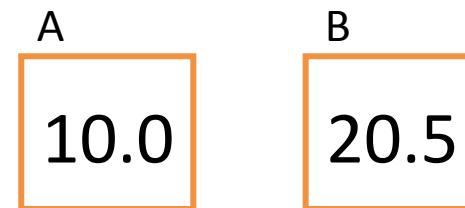


O que o usuário vê

Exercícios

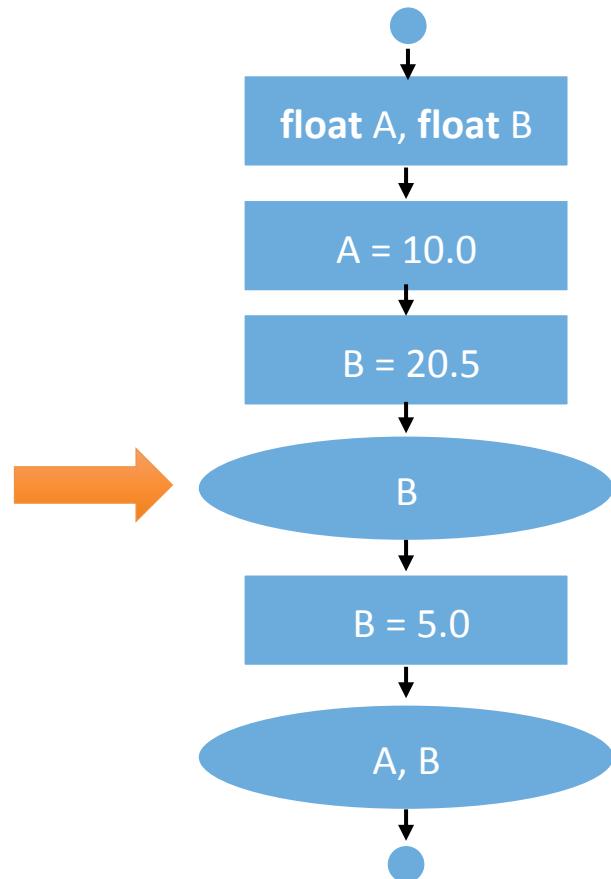


Memória

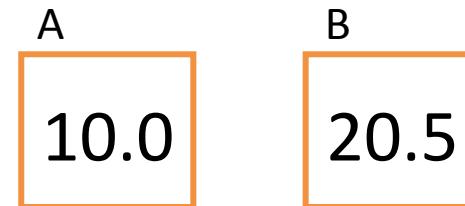


O que o usuário vê

Exercícios



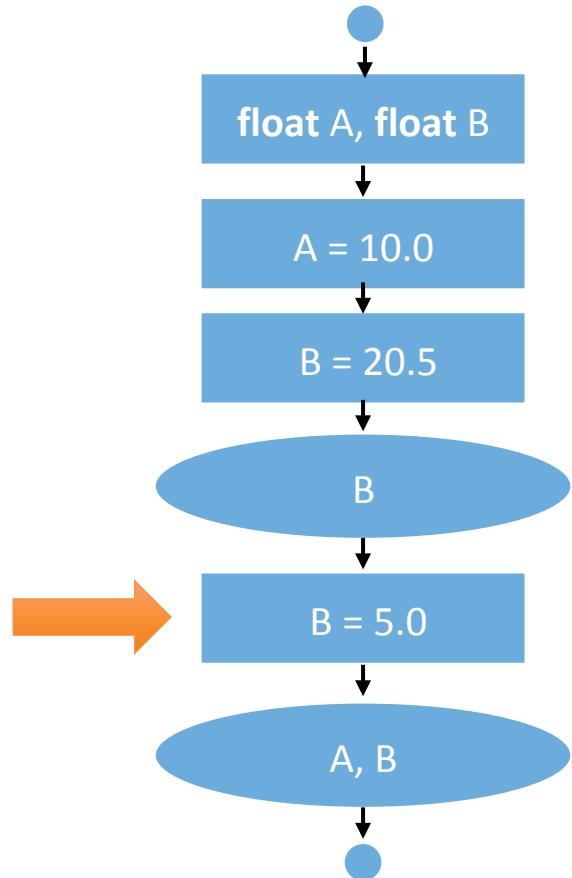
Memória



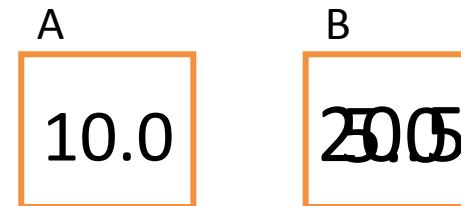
O que o usuário vê

20.5

Exercícios



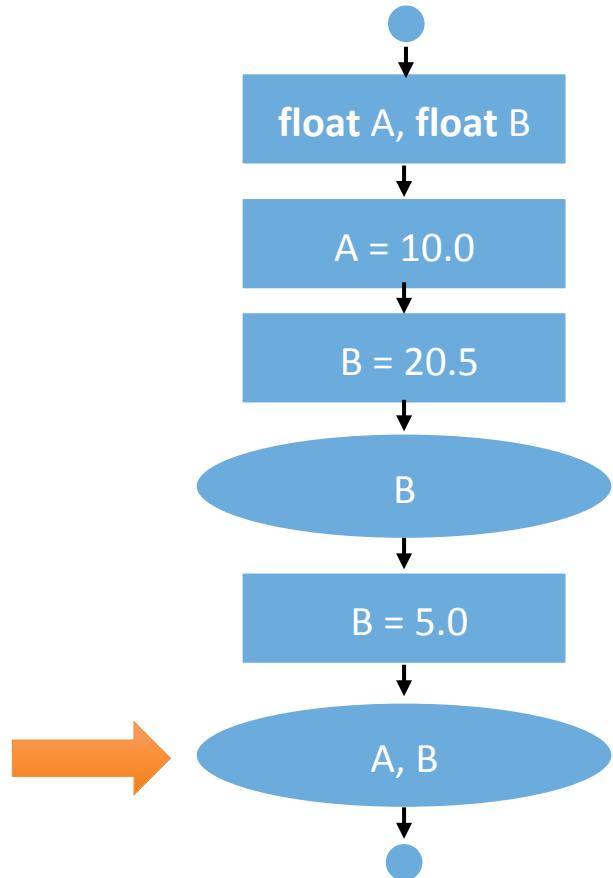
Memória



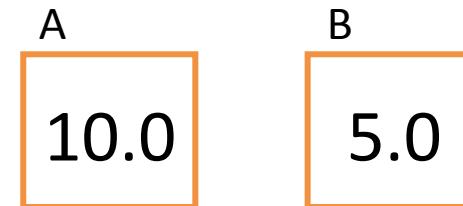
O que o usuário vê

20.5

Exercícios



Memória



O que o usuário vê

20.5
10.0 5.0

Exercício

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
```

b)

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

c)

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

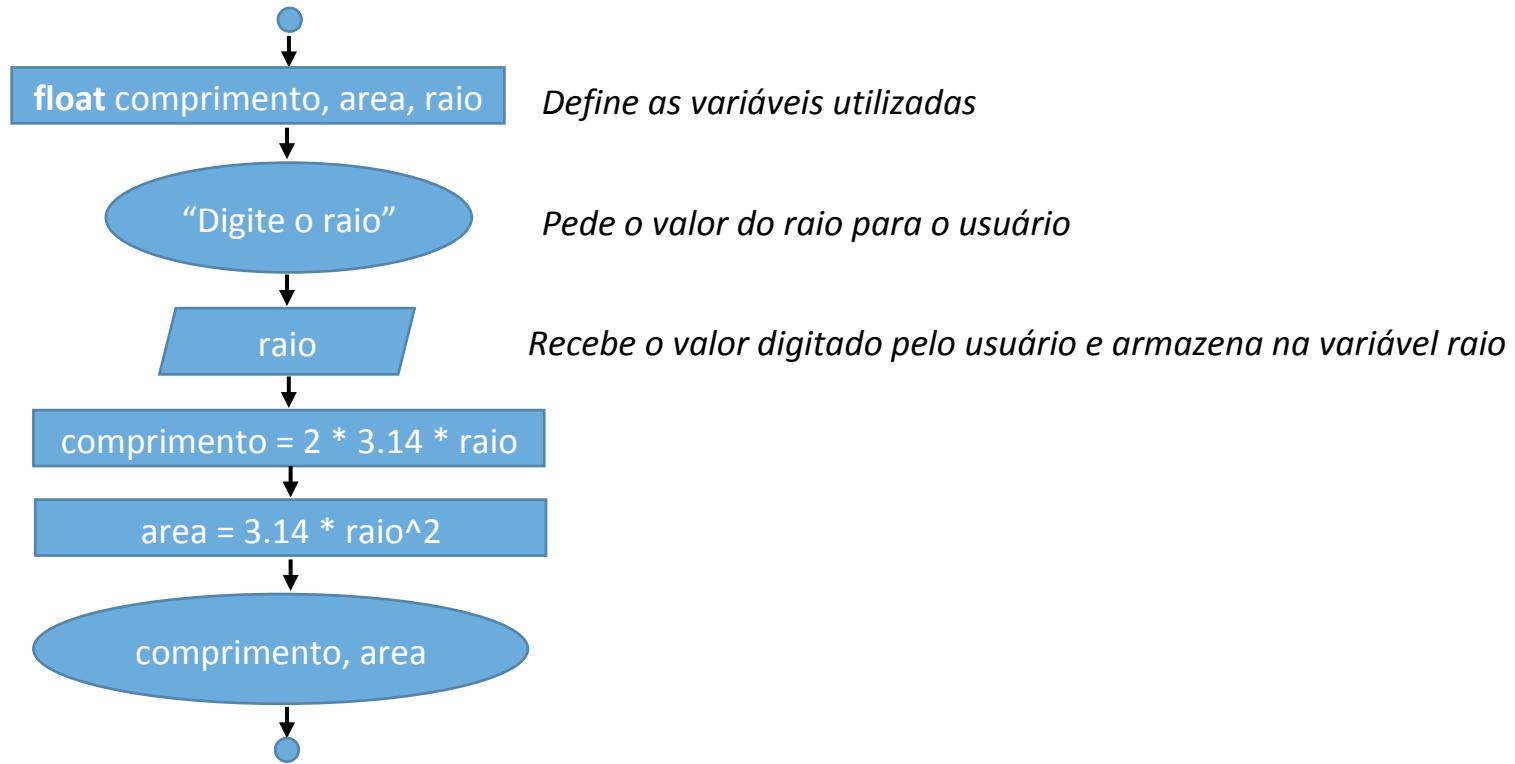
50
10 50
30 10 40

10 10 10

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

$$4 \% 2 = 0$$

$$5 \% 2 = 1$$

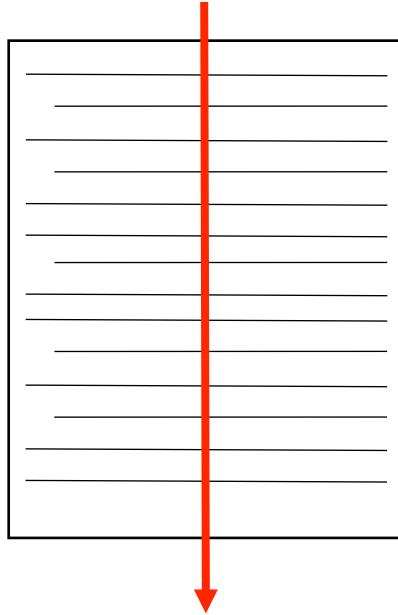
$$17 \% 3 = 2$$

$$100 \% 50 = 0$$

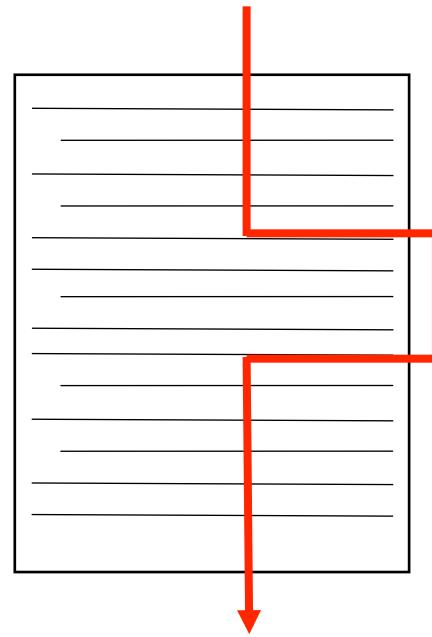
Estrutura de Controle

- 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

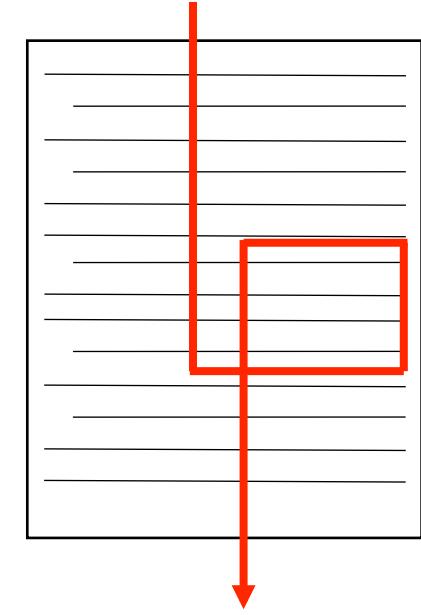
Estrutura de Controle



Fluxo de execução
Sequencial



Fluxo de execução
com desvio



Fluxo de execução
repetitivo

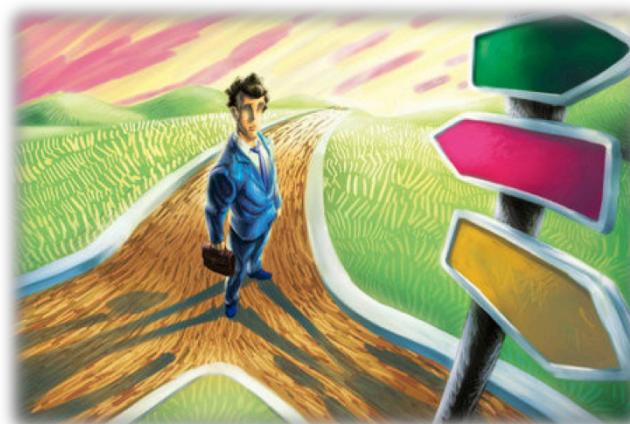
Comandos são executados um após o outro

Comandos são executados dependendo do valor de uma condição

Comandos são executados de forma repetida

Estrutura de Controle

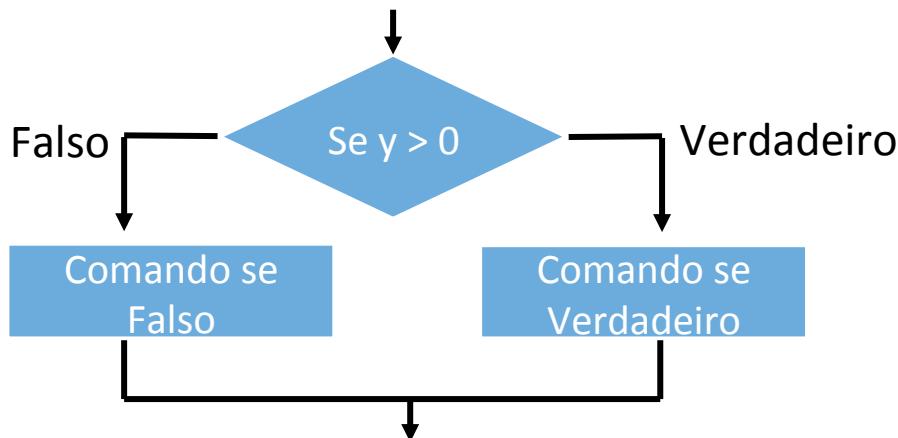
- As estruturas de controle dividem-se em:
 - Estruturas de decisão (*condicionais*);
 - Estruturas de repetição (*loops de repetição*).
- As estruturas de controle estão vinculadas às **condições** que determinam se instruções serão ou não executadas;
- Uma **condição de controle** está relacionada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



Estrutura de Controle

Condisional

Verifica se uma condição é satisfeita

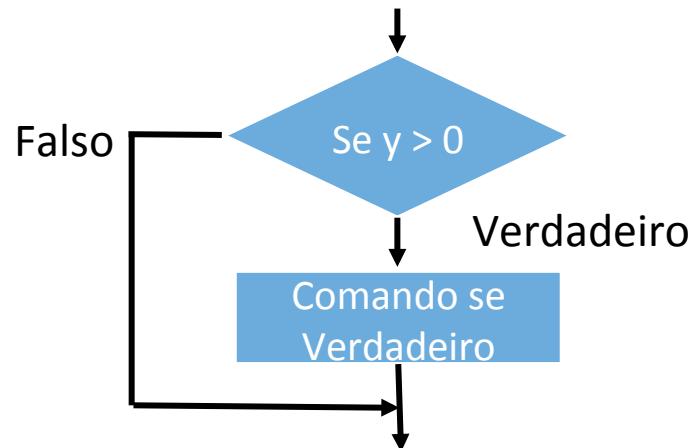


Se y é maior que zero

Executa Comando se Verdadeiro

Senão

Executa Comando se falso



Se y é maior que zero

Executa Comando se Verdadeiro

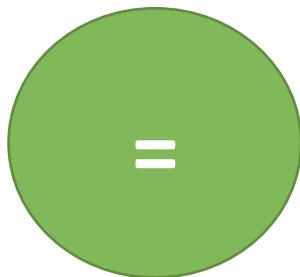
NUNCA executa o verdadeiro e falso simultaneamente

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

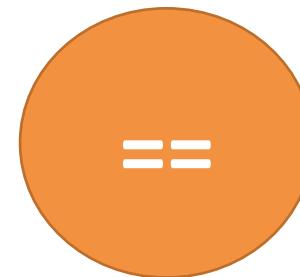
Operadores Relacionais



Operador de Atribuição

$x = 5$

x recebe o valor 5



Operador Relacional

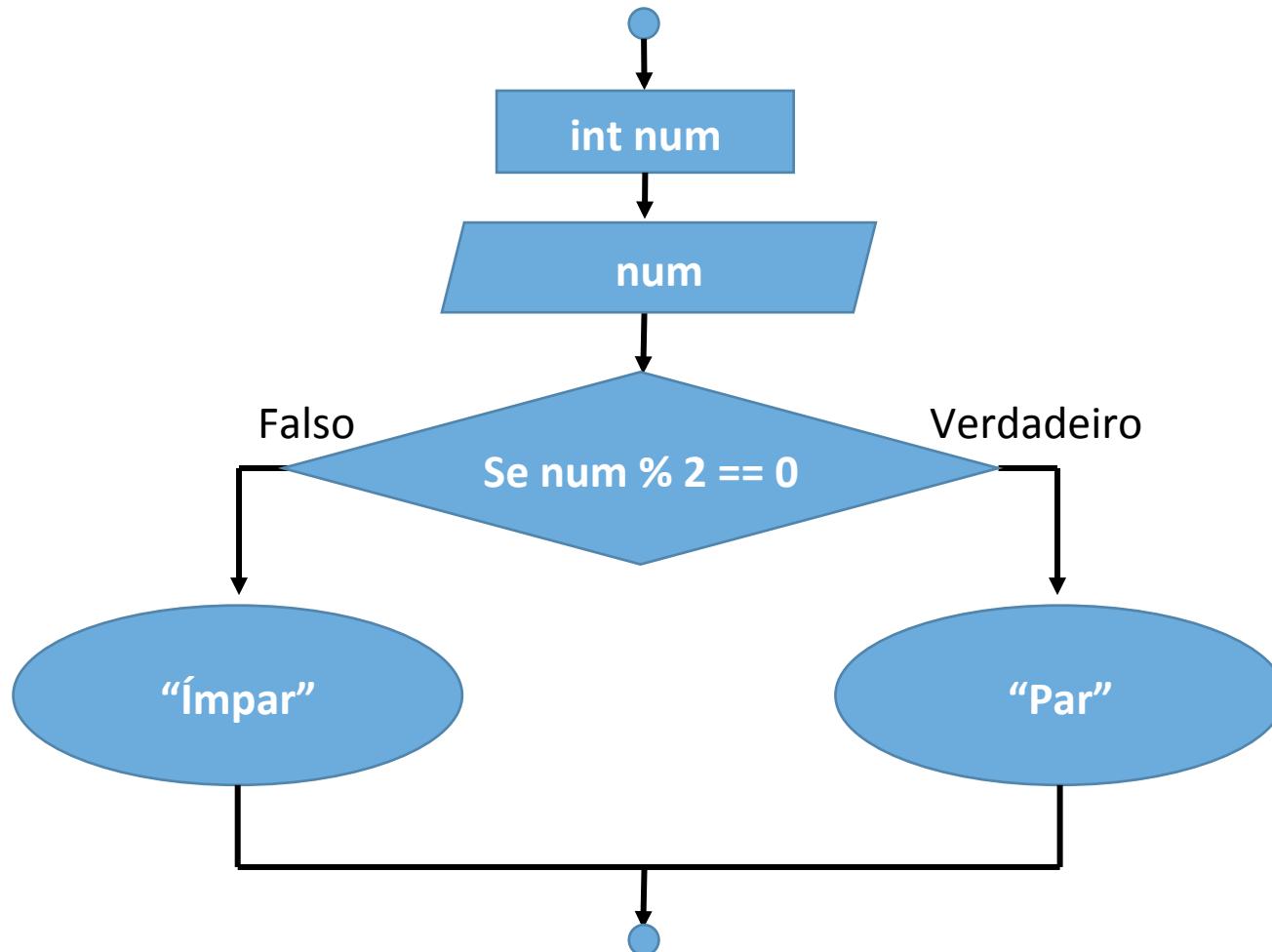
$x == 5$

O valor armazenado
em x é igual a 5?

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

Exemplo

Programa que indica se um número é ímpar ou par



Exercício

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 até 5.9	Exame
De 6.0 até 10.0	Aprovado

Exercício

3. Faça um programa que leia **dois números quaisquer**, e imprima o resultado do cálculo do **maior dividido pelo menor**.
4. Faça um programa que leia um código (um valor inteiro):
 - **1** – sinal vermelho;
 - **2** – sinal verde.Baseado no código digitado diga para o pedestre que está esperando no sinal: “**SIGA**” ou “**ESPERE**”.
5. Desenvolver a lógica para um programa que efetue o cálculo do reajuste de salário de um funcionário.
 - Salário ≤ 500 , reajuste de 15%
 - Salário > 500 , mas Salario ≤ 1000 , reajuste será de 10%
 - Salário > 1000 , reajuste será de 5%

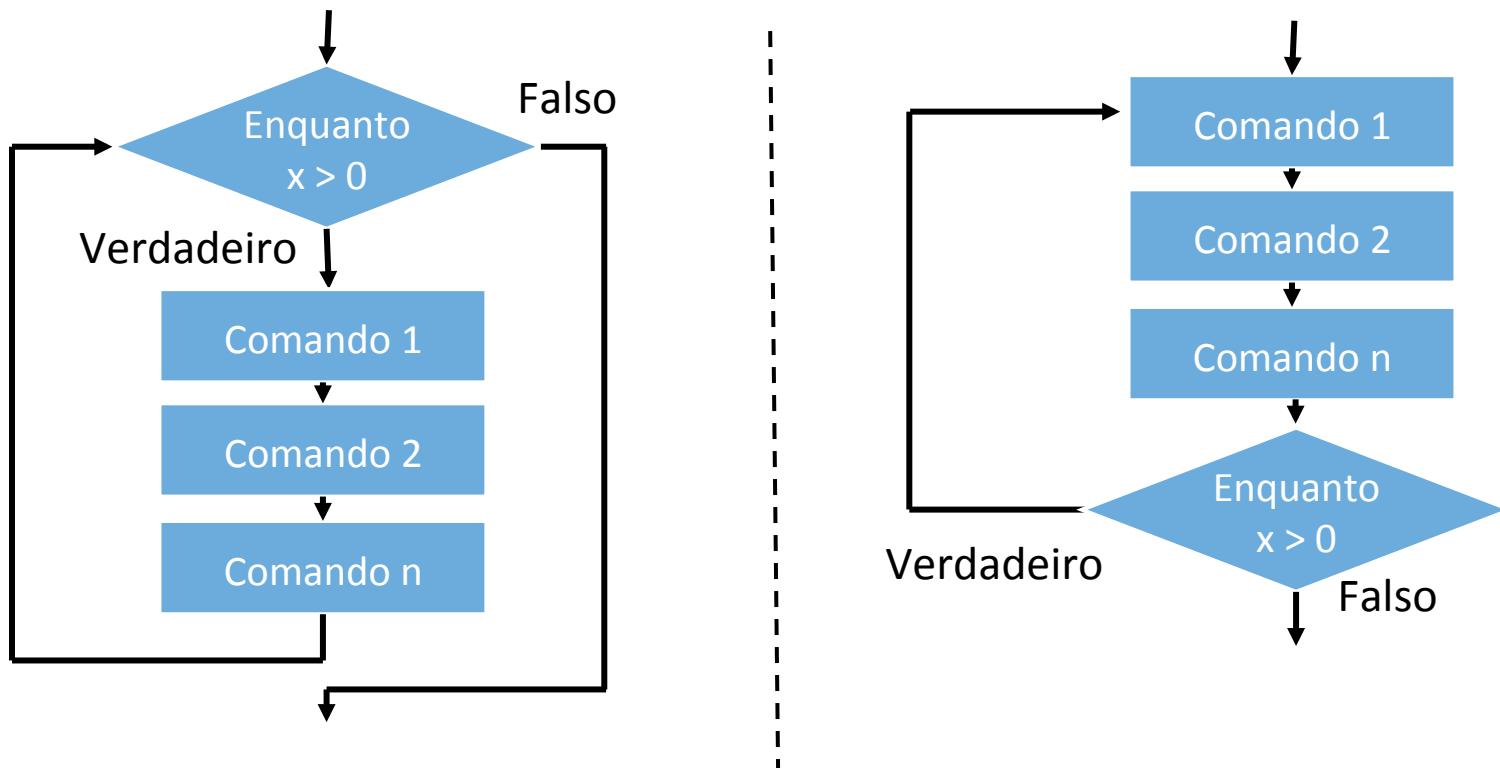
Exercício

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.: $\text{total} = \text{total} + \text{valor}$.
- **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.: $\text{total_de_itens} = \text{total_de_itens} + 1$

Exemplos

1. Faça um algoritmo que leia 10 valores e escreva no final a soma dos valores lidos.
2. 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.

Exercício

1. Faça um algoritmo que imprima os números pares de 0 até 20;
2. Imprima a tabuada do 7;
3. 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;
4. 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;
5. Calcule o resultado da série:
 - $S = 1/1 + 3/2 + 5/3 + 7/4 + \dots + 99/50$

Exercício

6. 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.

7. Faça um algoritmo para calcular $n!$
Lembrando que: $n! = n * (n-1) * (n-2) * \dots * 1$
 $0! = 1$, por definição

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	Equivale
!	Negação	Não
&&	Conjunção	E
	Disjunção	OU

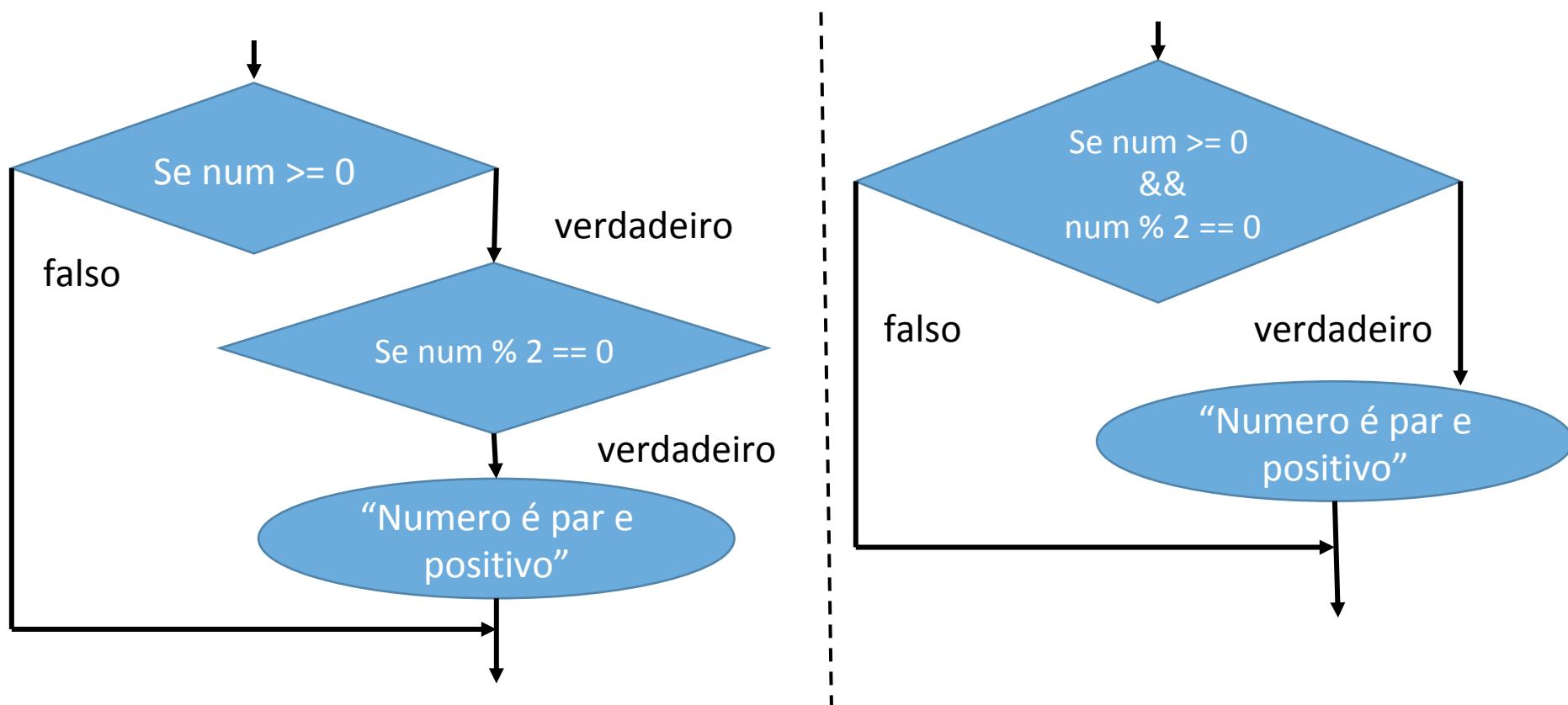
Operadores Lógicos

- 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 Lógicos

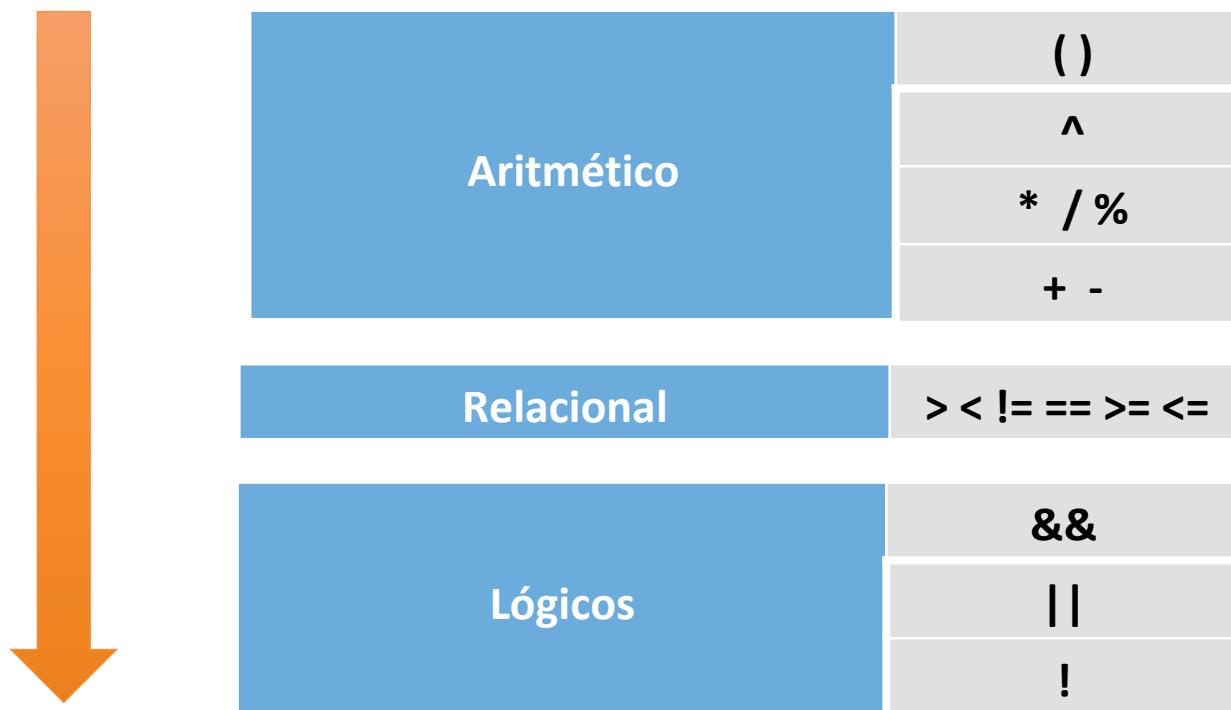
- Se quisermos testar se um número **num** é positivo e par



Operadores Lógicos

- Condições simples podem ser combinadas para formar condições complexas
- Exemplos:
 - $3 > 2 \ \&\& \ 7 > 3 \rightarrow$ resulta em Verdadeiro
 - $3 < 2 \ \&\& \ 7 > 3 \rightarrow$ resulta em Falso
 - $3 < 2 \ \|\ \ 7 > 3 \rightarrow$ resulta em Verdadeiro
 - $!(3 < 2 \ \&\& \ 7 > 3) \rightarrow$ resulta em Verdadeiro

Ordem de precedência

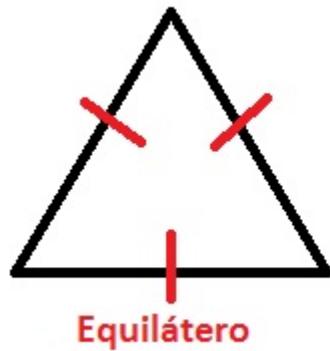


Exercícios

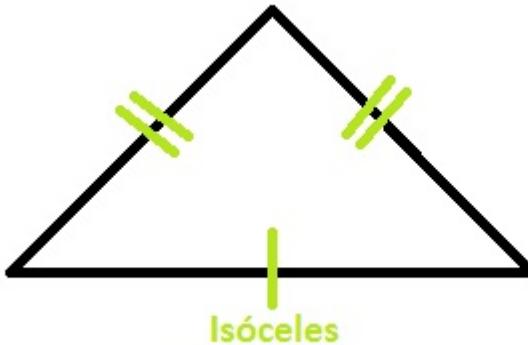
1. Sabendo que $A = 5$, $B = 4$, $C = 3$ e $D = 6$, informe se as expressões abaixo são verdadeiras ou falsas
 - $(A > C) \&\& (C \leq D)$
 - $(A + B) > 10 \mid\mid (A + B) == (C + D)$
 - $(A \geq C) \&\& (D \geq C)$

Exercícios

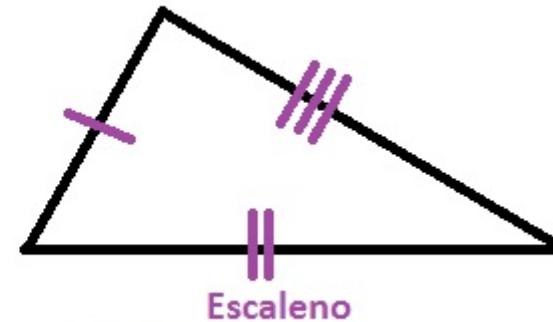
2. Leia 3 valores (L_1 , L_2 e L_3) que correspondem aos lados de um triângulo. Verifique se o triângulo é:
- Equilátero (3 lados com mesma medida);
 - Escaleno (nenhum dos lados tem a mesma medida)



Equilátero



Isóceles

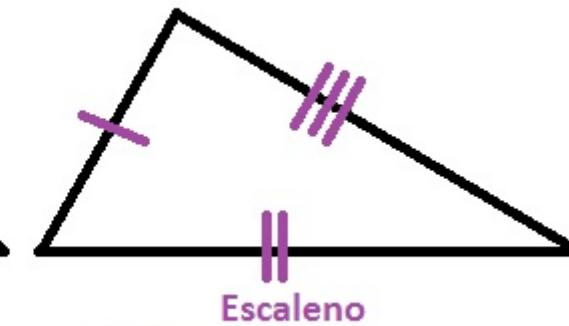
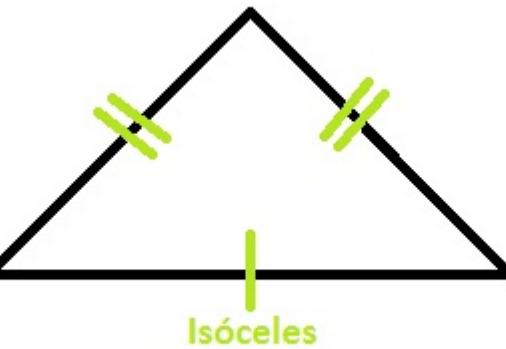
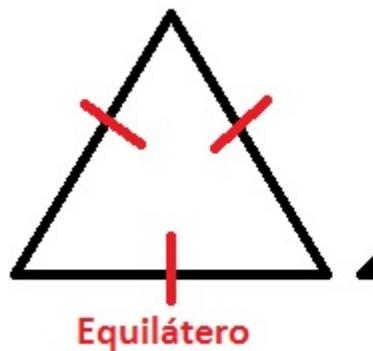


Escaleno

Exercícios

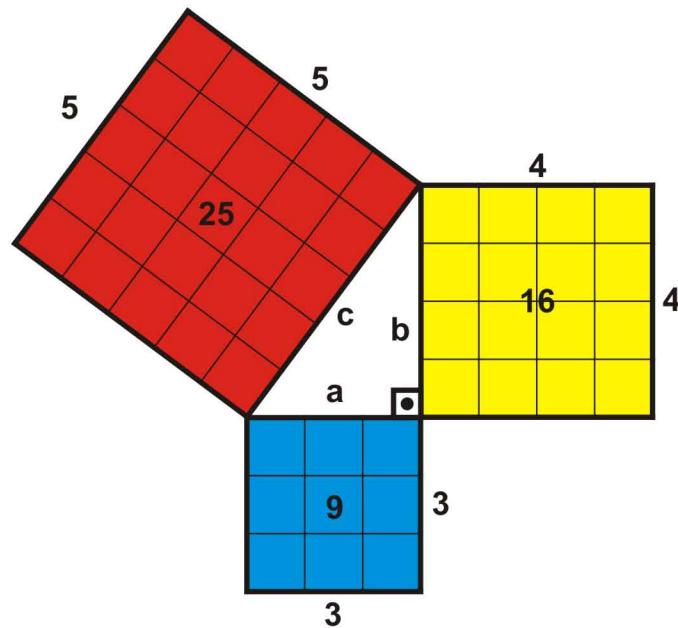
3. As medidas só podem formar um triângulo caso cada um dos lados seja menor que a soma dos outros dois lados. Faça o programa anterior verificar se as medidas passadas podem formar um triângulo.

- Se não formar, imprima “Não é um triângulo” e encerre o algoritmo.
- Se for um triângulo, imprimir se é escaleno ou equilátero.



Exercícios

4. O quadrado de um dos lados do triângulo deve ser igual a soma dos quadrados dos outros lados (Teorema de Pitágoras) para determinar se o triângulo é do tipo retângulo. Crie um algoritmo para verificar se um triângulo é retângulo. Considere que o usuário sempre fornece valores positivos e maiores que zero para cada um dos lados.



Exercícios Extras

1. 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.
2. 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).

Exercícios Extras

3. Faça um algoritmo que receba 10 números e que calcule e mostre a quantidade de números entre 30 e 90.
4. Faça um algoritmo que receba 10 idades, pesos e alturas e que calcule e mostre:
 - a) A média das idades das pessoas;
 - b) A quantidade de pessoas com peso superior a 90kg e altura inferior a 1,5m.
 - c) A porcentagem de pessoas com idade entre 10 e 30 anos que medem mais de 1,8m.
5. 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.