

# Algoritmos de Programação

## Instruções de Entrada e Saída

## Operadores Relacionais

Análise e Desenvolvimento de Sistemas/Presencial  
Professora Ma. Joyce Siqueira  
Email: [joycitta@gmail.com](mailto:joycitta@gmail.com)

# Descrição de Algoritmo

- **Português Estruturado (Portugol) → utilizada no curso**
  - pseudolinguagem de programação
  - “pensar” no problema e não na máquina mas não ficar tão longe dela
  - sintaxe definida e forma aceita como padrão
  - **diferencia letras maiúsculas e minúsculas** (Case Sensitive)
- **Diagrama de Chapin (*Nassi e Schneiderman*)**
- **Fluxograma → utilizada no curso**

## TIPOS DE DADOS

Tipo de dados	Características	Exemplos
Inteiro	Contempla todos os números inteiros relativos (positivos, negativos e nulos).	12, -56, 852, 146698
Real	Contempla todos os números reais (positivos, negativos e nulos).	23.5, 85.4, -354.8, -74.0, 3.141618
Caractere	Contempla todos os caracteres alfanuméricos e especiais.	0 - 9, A - Z, a - z, #, %, \$, &, *, @
Lógico ou Booleano	Assumem apenas a situação de verdadeiro ou falso.	0 ou 1, verdadeiro ou falso, sim ou não.

## TIPOS DE DADOS

Identifique os tipos primitivos presentes nas sentenças abaixo:

- A placa “Não Estacione” tinha 1 furo.
- A escada em T possui 10 degraus.
- Eu gastei 1,5 horas na fila.
- Ele comprou o adesivo escrito “Análise e Desenvolvimento de Sistemas - UCB” por R\$ 12,50.
- Ele conseguiu 55,5 segundos nos 100 metros rasos.

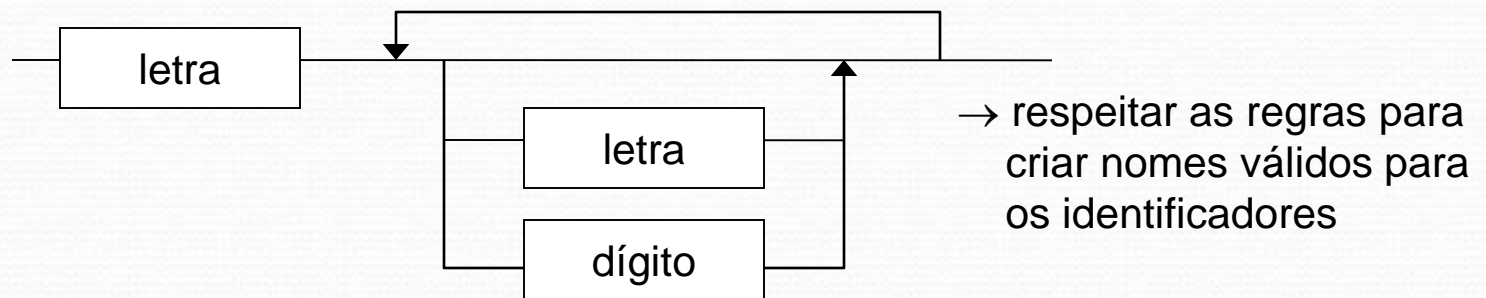


## RESPOSTAS

- A placa “Não Estacione” tinha 1 furo.
  - Texto: “Não Estacione”
  - Inteiro: 1 (furo)
- A escada em T possui 10 degraus.
  - Inteiro: 10 (degraus)
  - Caractere : 'T'
- Eu gastei 1,5 horas na fila.
  - Real: 1,5 (hora)
- Ele comprou o adesivo escrito “Análise e Desenvolvimento de Sistemas - UCB” por R\$ 12,50.
  - Texto : “Ciência da Computação - UCB”
  - Real: 12,50
- Ele conseguiu 55,5 segundos nos 100 metros rasos.
  - Inteiro: 100 (metros)
  - Real: 55,5

# VARIÁVEL

- Se caracterizam por poder sofrer alteração em algum instante durante a execução do algoritmo.
- Representa um nome de um local onde se pode colocar um valor ou conjunto de valores.
- O nome da variável é um **identificador** (deve ser **significativo** - **cuidado com letras maiúsculas e minúsculas**), cuja sintaxe é :



Exemplo: nome, ~~1a~~, ~~#55~~, ~~o{2}~~, k4, achou, ~~“aio”~~, y1

# Identificadores

- As constantes ou variáveis possuem identificadores, ou seja, um nome que representará o dado, como uma etiqueta.
- Para formação dos identificadores é preciso seguir 3 regras:
  - Devem começar com um caractere alfabético;
  - Podem ser seguidos por mais caracteres alfanuméricos ou numéricos.
  - Não devem ser usados caracteres especiais.
- Exemplos de identificadores válidos: Notas, X, K7.
- Exemplos de identificadores inválidos: 7X, E(13), A:B.



# Exercícios

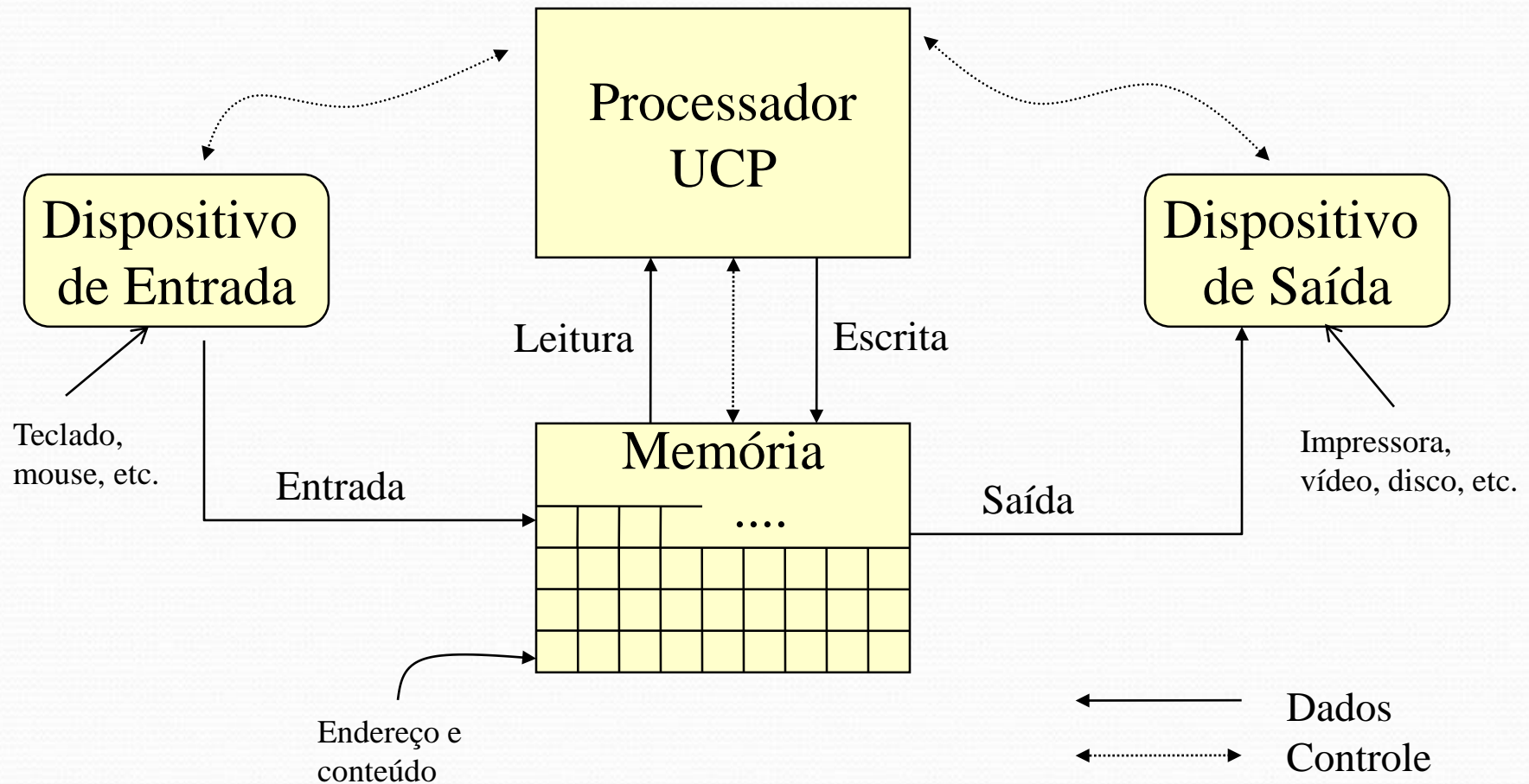
**Quais dos identificadores abaixo são válidos:**

**a) (X)   b) U2   c) Rua   d) #55   e) CEP   f) 5B**

**Supondo que as variáveis NR, NA, NMat e SX sejam utilizadas para armazenar a nota, o nome, o número de matrícula e o sexo do aluno, declare corretamente, associando o tipo primitivo e respeitando a sintaxe.**

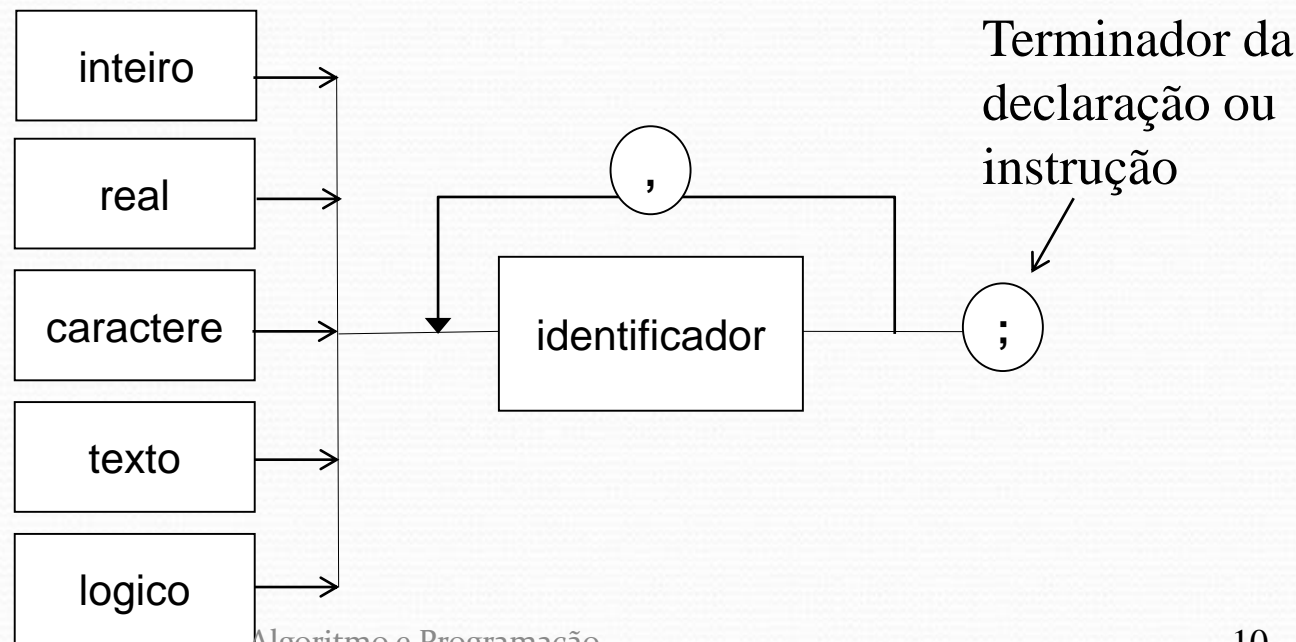


# Componentes de um sistema de computação



# DECLARAÇÕES DE VARIÁVEIS

- Todo dado a ser colocado na memória deve ser previamente identificado (**usar nomes significativos**).
  - primeiro saber qual o seu tipo e depois fazer o seu armazenamento.
- A declaração de variável corresponde a criação de locais na memória com o nome da variável (identificador) marcado com o tipo que pode assumir
- Sintaxe:



## Exemplos:

inteiro x1, numCorreto, conta;

real valor;

texto nome, frase, enderecoCliente;

caracter resposta;

logico achou;

## Logo:

- **x1** é um local da memória onde só pode ser armazenado números inteiros
- **nome** é um local de memória que só pode ser armazenado caracteres alfanuméricos
- palavras sublinhadas são palavras-reservadas

# OPERADORES

- **ARITMÉTICOS**

$+$ ,  $-$  (binário ou unário),  $*$ ,  $/$  ( resultado da divisão real)  
 $\backslash$  (resultado da divisão inteira),  
 $\%$  ou **mod** (resto da divisão inteira)

- **RELACIONAIS**

$==$ ,  $!=$ ,  $>$ ,  $>=$ ,  $<$ ,  $<=$

- **LÓGICOS**

**E** para conjunção  
**OU** para disjunção  
**! ou NAO** para negação

- **CARACTERES**

$+$  (concatenação)  
(este operador só pode  
ser aplicado sobre textos)



# Operadores Matemáticos

Operações	Operador
Adição	+
Subtração	-
Multiplicação	*
Divisão	/

# Linearização de Fórmulas

$$y = \frac{2x}{5x+7} + 4x$$

$$y = ((2*x)/(5*x + 7)) + (4*x)$$

# Operações Complexas

Operações	Operador	Exemplo
Divisão inteira	\ ou div	Retorna o valor inteiro
Módulo	%	Resto da divisão de a por b
Inversão de sinal	-	-a (-(-a))

- **Uso apenas de variáveis numéricas, operadores aritméticos e funções embutidas(definidas pela linguagem)**
  - raizQuadrada(x) calcula a raiz quadrada de x
  - abs(x) obtém o valor absoluto(módulo) de x
  - exp(b,e) obtém o valor de b elevado a e
  - comparaTexto(end1,end2) compara o texto end1 com end2 (retorna 0, >0 ou <0)
  - maiusculo(nome1) retorna o maiúsculo do texto nome1
  - minusculo(nome1) retorna o minúsculo do texto nome1
  - maiusculoCaracter(c1) retorna o maiúsculo do caractere c1
  - minusculoCaracter(c1) retorna o minúsculo do caractere c1
  - tamanhoTexto(texto1) retorna o tamanho do texto (quantidade de caracteres)
  - limpaTela(), limpa toda a tela de execução
  - resto da divisão de inteiros  $m$  por  $i \rightarrow m \bmod i \rightarrow m \% i$
  - quociente inteiro de inteiros  $m$  por  $i \rightarrow m \setminus i$



## Exercício

Faça a linearização da fórmula de Báskara.

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

- As expressões são calculadas da esquerda para direita obedecendo a precedência dos operadores.
- Precedência dos operadores:
  - + , - (unitários)
  - \* , / , \ , % (mod)
  - + , - (binários)

- **Cuidado:**

$$2+3*5 \neq (2+3)*5$$

$$2+3/5 \neq (2+3)/5$$

**Dica:** faça uso de parênteses no caso de dúvidas

- ✓ Uso de operadores relacionais
- ✓ Utilizado em comparações (relações)
- ✓ Precedência dos operadores:

$<$  ,  $<=$

$==$

$>$  ,  $>=$  ,

$!=$

**Dica:** faça uso de parênteses no caso dúvidas



- Uso de operadores lógicos
- Semântica pela **Tabela Verdade**:
  - Conjunto de todas as possibilidades combinatórias entre valores de diversas variáveis lógicas, que se encontram em duas situações e um conjunto de operadores lógicos

A	B	A <u>e</u> B	A <u>ou</u> B	<u>não</u> A
F	F	F	F	V
F	V	F	V	V
V	F	F	V	F
V	V	V	V	F

Precedência dos operadores: nao, e, ou

- Exemplo
  - $5 > 2 \text{ e } 100/2.0 == 50$  (V)
  - se é Sábado e nao é feriado, entao tem aula (Hoje tem aula) (V)
  - nao verdadeiro (F)



# PRECEDÊNCIA GERAL

**Primeiro:** parênteses e funções

**Segundo:** expressões aritméticas

+, - (unitários)

\*, /, \, % (**mod**)

+, - (binários)

**Terceiro:** comparações

<, <=, ==, >=, >, !=

**Quarto:** não, !

**Quinto:** e

**Sexto :** ou

# Expressões Lógicas: Negação

## Exemplos:

- nao (verdadeiro)  $\rightarrow$  falso
- nao (a e b)  $\rightarrow$  nao a ou nao b  $\rightarrow$  !a ou !b
- nao (a ou b)  $\rightarrow$  nao a e nao b
- nao ( a == b)  $\rightarrow$  a != b
- nao (a > b)  $\rightarrow$  a <= b
- ... (entre outras propriedades)

# COMANDOS BÁSICOS

- **Comando de atribuição:** atribuir um valor a uma variável
- Sintaxe:



- Exemplo:  
aux = 2;  
soma = aux + 100;  
b = (5 == 3);



# Exercício Proposto

- O que está errado nos seguintes comandos de atribuição:

logico a, b;

real c, d;

inteiro x;

c = a + 100;

c = a == b;

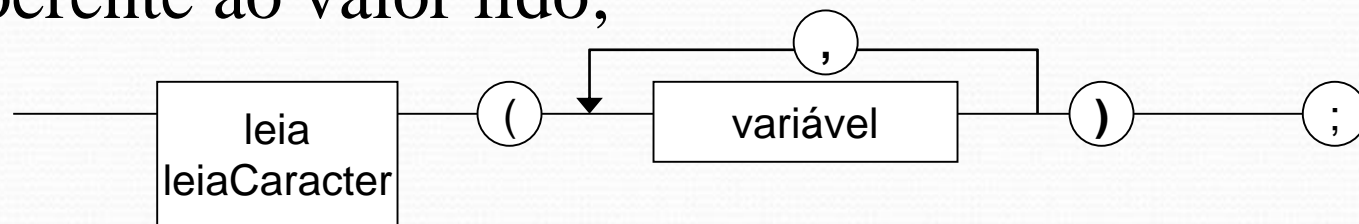
c e d = 24.5;

x = a;



# COMANDOS BÁSICOS

- **Comando de leitura de dispositivos de entrada:** ler um valor e atribuir a uma variável com tipo de dado coerente ao valor lido;



- Exemplo

- leia (idade);
- leia (nome);
- leiaCaracter(resposta);

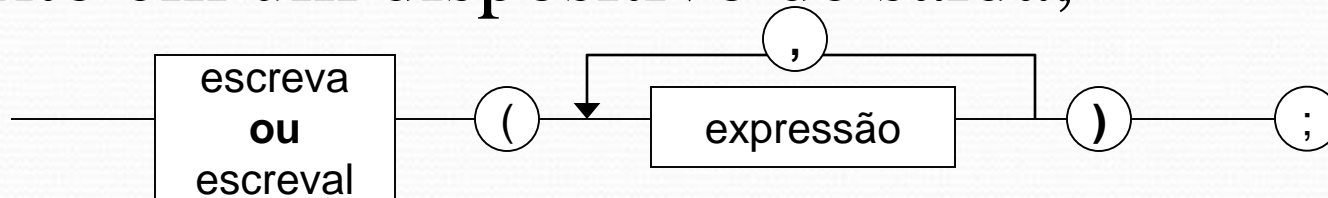
**leia** → utilizado para ler variáveis inteiras, reais e textos

**leiaCaracter** → utilizado para ler variáveis caracteres

- Observação: a sintaxe em alguns livros considera **ler** no lugar de **leia**

# COMANDOS BÁSICOS

- **Comando de saída para dispositivos de saída:** escreve o valor de uma variável ou texto em um dispositivo de saída;



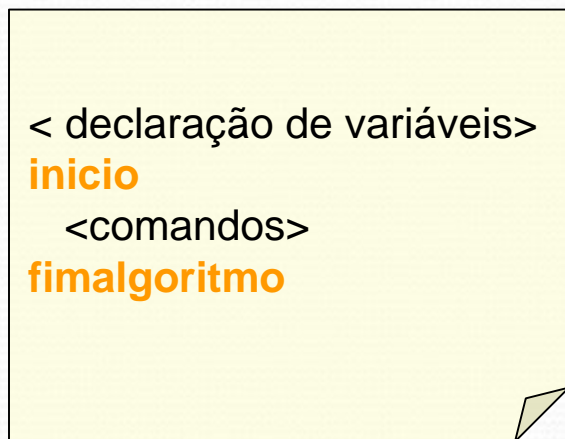
## Exemplo:

**escreval** → escreve e salta para a próxima linha na tela de execução

- escreva (idade);
- escreva ("Você pesa", peso , "quilos");
- escreval (x + 3);
- Observação: a sintaxe em alguns livros considera **imprima** ou **escrever** no lugar de **escreva**

# BLOCOS

- Consiste em um conjunto de comandos (ou instruções) com uma função bem definida;
- Serve para definir os limites onde as variáveis declaradas em seu interior são conhecidas;



Português Estruturado

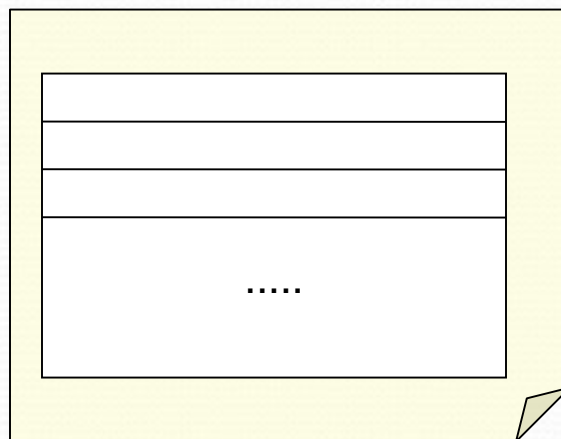
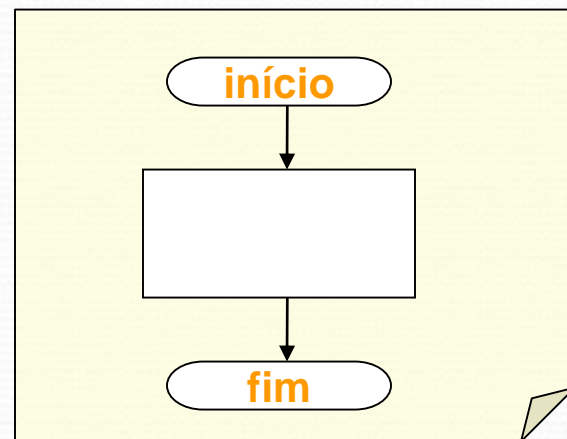


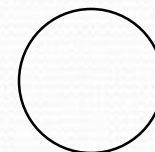
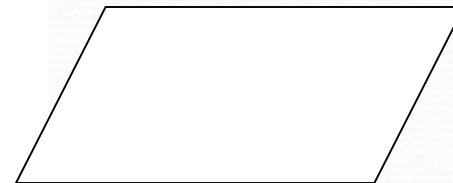
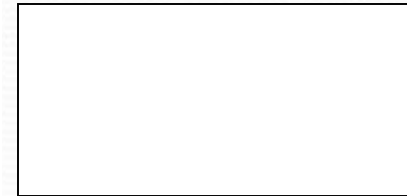
Diagrama de Chapin



Fluxograma

# Fluxograma: símbolos para estrutura sequencial

- Início e Fim
- Comandos de processamento
- Comandos de entrada de dados
- Comando de saída de dados
- Conector





- Seqüencial
  - conjunto de comandos que serão executados em uma seqüência linear de cima para baixo (linha por linha)
- Seleção (ou condicional)
  - uma ação ou um conjunto de ações a ser selecionada para execução, dependendo do resultado de uma avaliação condicional (lógica convencional – verdadeiro ou falso)
- Repetição (ou *looping*)
  - conjunto de ações que é executada repetidamente enquanto uma condição permanece válida ou não, de acordo com o comando utilizado

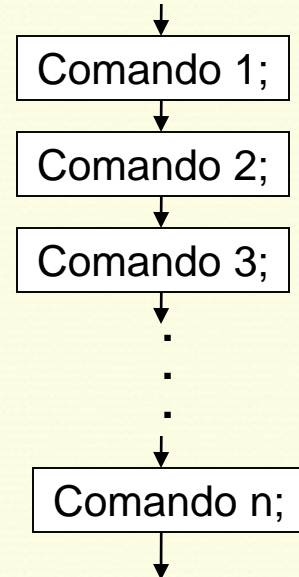
# ESTRUTURA SEQUENCIAL

Comando 1;  
Comando 2;  
Comando 3;  
.  
.  
.  
Comando n;

Português Estruturado

Comando 1;
Comando 2;
Comando 3;
·
·
·
Comando n;

Diagrama de Chapin



Fluxograma

- O nome dado a uma constante também é um **identificador**
- As constantes podem ser:

**Numéricas**: representadas por valores reais ou inteiros, onde o ponto final '.' separa a parte inteira da decimal (fracionária).

Exemplo: -0.59                      2.0                      -34.597                      -100

**Lógicas**: representadas pelas palavras **verdadeiro** ou **falso**, sendo denominadas constantes *booleanas* (lógica convencional).

**Literais**: formadas por um caractere ou uma sequência de caracteres aceitos na simbologia da linguagem (algarismos especiais, letras e números). Elas são representadas pelos caracteres correspondentes colocados entre aspas simples(') para um caractere ou aspas duplas (") para um ou mais caracteres, exemplo:

'N', 'S', 'n', "algoritmo ", "José ", "Sala B-110"



# Exercícios Proposto

- Construa um algoritmo que leia o nome do usuário e responda sempre BOM DIA para esse usuário.

## ➤ Organizando o algoritmo de solução

Síntese do  
problema

**algoritmo** saudacao;

// **Síntese**

// Objetivo: saudação de bom dia

// Entrada : nome do usuário

// Saída : mensagem de bom dia com nome

**principal**

Bloco de criação  
ou declaração

// **Declarações**

texto nome, msg; // msg é uma constante

Bloco de  
instruções

// **Instruções**

msg = "Bom dia " ;

escreva("Informe o seu nome: ");

leia(nome);

escreval(msg , nome);

**fimPrincipal**



# Metodologia de desenvolvimento de algoritmo

- **Passo 1**
  - leia cuidadosamente toda a especificação do problema
  - faça anotações
- **Passo 2**
  - se não entender, repita o **passo 1** até entender (tire dúvidas com quem especificou ou conhece bem o problema)
- **Passo 3**
  - levantar todas as saídas exigidas na especificação do problema
- **Passo 4**
  - levantar todas as entradas exigidas na especificação do problema ou necessárias para sua solução

# Metodologia de desenvolvimento de algoritmo

- **Passo 5**
  - verificar a necessidade de geração de valores intermediários
  - fazer comentários
- **Passo 6**
  - levantar todas as transformações necessárias para a partir da entrada produzir a(s) saída(s)
  - fazer comentários
- **Passo 7**
  - testar cada passo do algoritmo usando valores no teste
- **Passo 8**
  - fazer uma reavaliação geral, elaborando o algoritmo

# OPERAÇÃO DE DIVISÃO

A operação aritmética de divisão pode ocorrer de duas formas distintas, de acordo com a necessidade que se deseje como resultado final (quociente).

## Divisão Inteira

- resto:  $5 \bmod 2 \rightarrow 1$
- quociente:  $5 \setminus 2 \rightarrow 2$

## Divisão Aritmética Padrão (comum)

- não existe a manipulação direta do resto
- quociente real para a operação solicitada

$$5 / 2 = 2.5 \rightarrow \text{resultado será sempre REAL}$$

# Referências Bibliográficas

## Básica:

- EVARISTO, J. **Aprendendo a programar: Programando em C.** Book Express, 2001.
- FARRER, H. etall. **Algoritmos Estruturados.** 3ª ed. LTC, 1999.
- MANZANO, J.; OLIVEIRA, J. **Algoritmos: Lógica para Desenvolvimento de Programação.** 6ª ed. São Paulo: Ética, 2000.



# Referências Bibliográficas

## Complementar:

- FORBELLONE, A. L. V. **Lógica de Programação: A construção de algoritmos e estrutura de dados.** Makron Books, 1993.
- GUIMARÃES, A.; LAGES, N. **Algoritmos e Estrutura de Dados.** LTC, 1994.
- MIZRAHI, V. V. **Treinamento em linguagem C: Módulo 2.** São Paulo: Makron Books, 1990.
- SALVETTI, D. D; BARBOSA, L. M. **Algoritmos.** São Paulo: Makron Books, 1998.
- SCHILDT, H. **C: Completo e total.** 3ª ed. São Paulo: Makron Books, 1997.