



ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES

AULA 2

Fundamentos de Algoritmo



Vandor Roberto Vilardi Rissoli



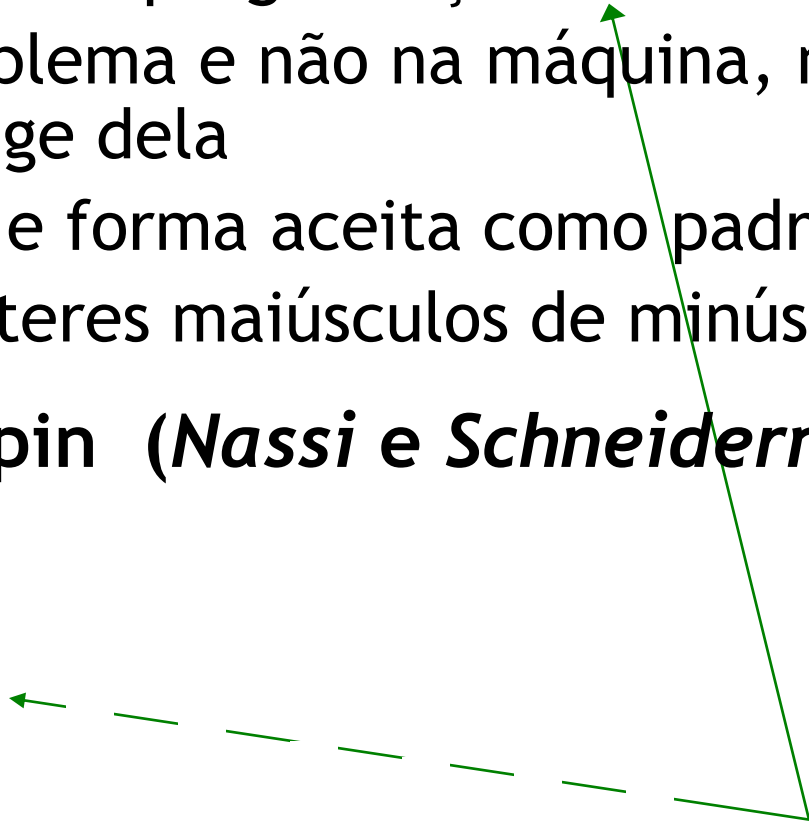
APRESENTAÇÃO

- Fundamentos de Algoritmo
- Formas de Representação
- Comandos Básicos
- Fluxograma
- Metodologia de Desenvolvimento
- Referências



Fundamentos de Algoritmo

- **Português Estruturado (Portugol)**
 - pseudolinguagem de programação
 - “pensar” no problema e não na máquina, mas não ficar tão longe dela
 - sintaxe definida e forma aceita como padrão
 - diferencia caracteres maiúsculos de minúsculos
- **Diagrama de Chapin (*Nassi e Schneiderman*)**
- **Fluxograma**



Usado na Disciplina



Fundamentos de Algoritmo

- Início da construção da parte (bloco) mais importante do algoritmo em Português Estruturado

Síntese do
problema

```
algoritmo Analisa_idade;  
// Síntese  
// Objetivo: analisar idade da pessoa  
// Entrada: idade  
// Saída: classifica em idoso ou jovem
```

- Para elaboração desta síntese é necessário o estudo do problema aprofundado para que seja possível propor uma **solução** adequada e completa
- O “problema” é seu maior aliado no sucesso



Fundamentos de Algoritmo

Elabore a síntese dos problemas apresentados abaixo, procurando entendê-los e construindo o bloco Síntese de seu futuro algoritmo completo.

- A. Uma oficina mecânica de automóveis deseja fazer um cadastro de seus clientes a fim de estreitar seu relacionamento e gerar benefícios e possíveis descontos para seus principais clientes. Faça somente a Síntese para que o proprietário desta oficina possa ter os dados de cada cliente e de seu(s) automóvel(is).
- B. Faça somente a Síntese para uma farmácia que deseja ter controle sobre seus estoques de medicamentos, principalmente para que NÃO se tenha prejuízo com seu armazenamento.



Fundamentos de Algoritmo

TIPOS DE DADOS

Tipos primitivos ou escalares processados pelo computador:

- **inteiro**: toda informação numérica inteira (não fracionária) negativa ou positiva, por exemplo: -3, 0, 100 (**limite de valores**)
- **real**: toda informação numérica pertencente ao conjunto dos números reais (inteiros ou fracionários), (negativa ou positiva), por exemplo: -5; 0; 12; -1.7; 101.5; 4.0 (**usa ponto**)
- **caracter**: sequência contendo letras, símbolos especiais ou numéricos (caracteres alfanuméricos), por exemplo: 'B', '3', '\$', 'd'
 - sequência deve ser indicada entre apostrofe (aspa simples)
- **texto**: sequência de caracteres entre aspas duplas (*string*), por exemplo: "Taguatinga", "3356-9025", "Desconto 10%"
- **logico**: tipo booleano e possui só os valores *falso* ou *verdadeiro*
 - esse tipo só aceita um entre esses valores (excludentes)

→ A grafia correta destes tipos devem ser **idênticas** as mostradas anteriormente.

Fundamentos de Algoritmo

Identifique os tipos primitivos presentes nas sentenças abaixo:

- A placa "Não Estacione" tinha um furo
- A escada possui dez degraus
- Eu gastei uma hora e meia na fila
- Ela é do sexo 'F' e comprou adesivo da "UnB FGA" por R\$ 15.20
- Ele conseguiu 81.75 segundos nos 200 metros rasos

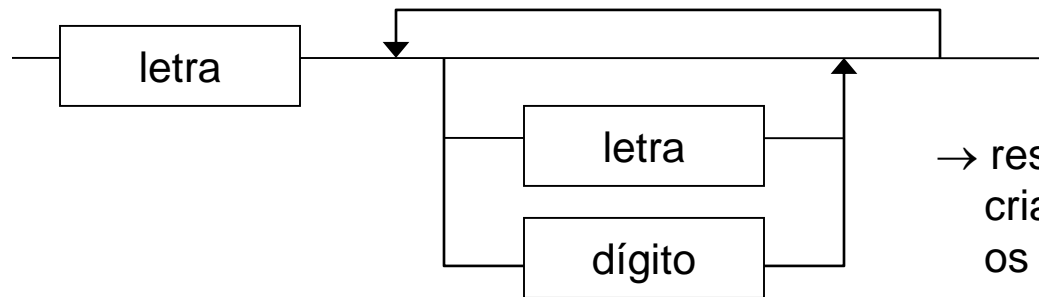
- texto: "Não Estacione"
- inteiro: 1 (furo)
- inteiro: 10 (degraus)
- real: 1.5 (hora)
- texto: "UnB FGA"
- caracter: 'F'
- real: 15.20
- inteiro: 200 (metros)
- real: 81.75



Fundamentos de Algoritmo

VARIÁVEL

- Tudo aquilo que é sujeito a variações, que é incerto, instável ou inconstante
- Representa o nome (*case sensitive*) de um local onde se pode guardar um valor ou um conjunto
- O nome da variável é um **IDENTIFICADOR**, cuja sintaxe respeita as regras a seguir:



→ respeitar as regras para criar nomes válidos para os **identificadores**

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

Fundamentos de Algoritmo

IDENTIFICADOR COMPOSTO

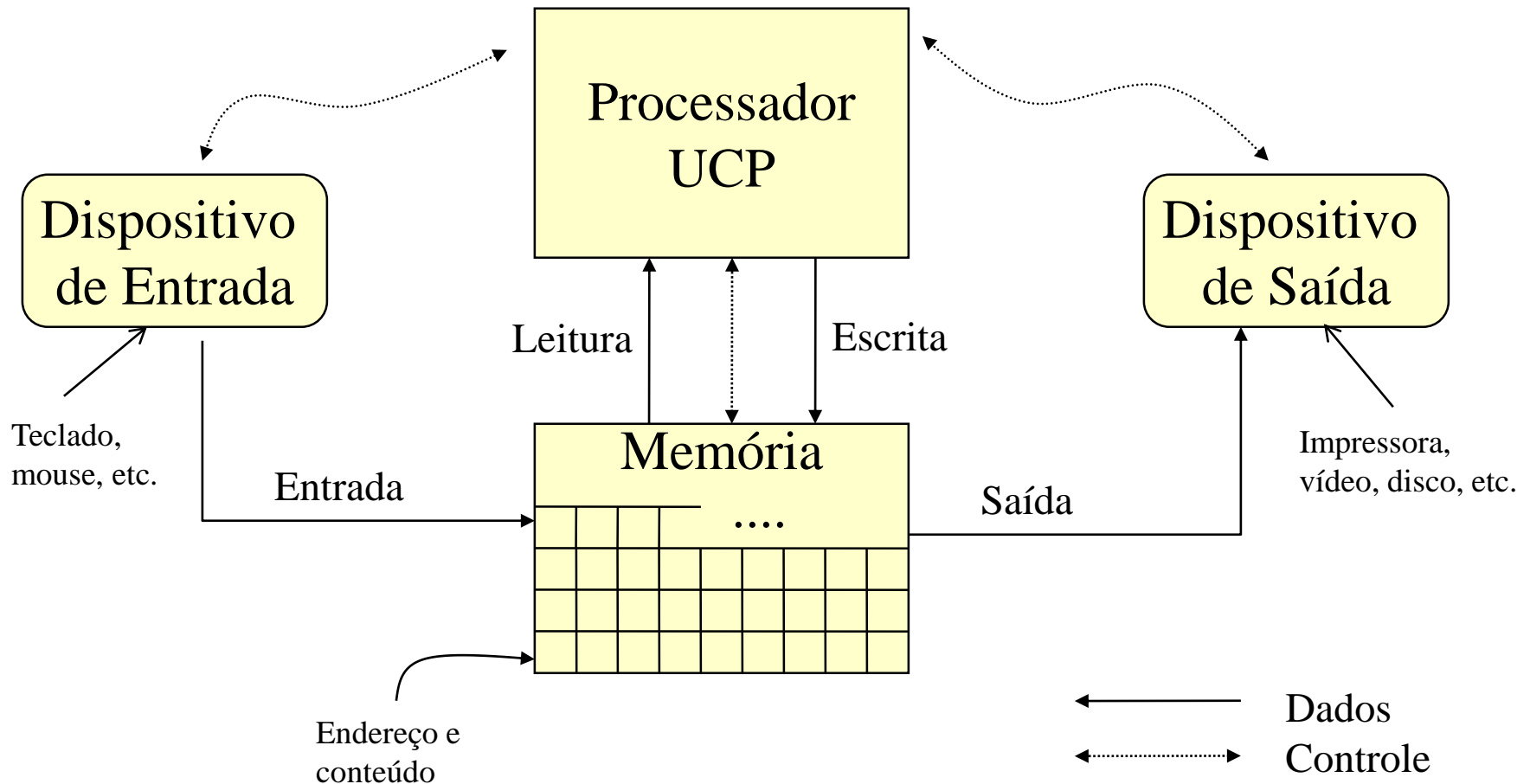
- A definição de identificador SEMPRE emprega nomes significativos ao que será guardado;
- O emprego de nomes significativos deve seguir as regras de formação corretas para fácil elaboração e manutenção de algoritmos mais complexos, podendo ser necessário o uso de nomes compostos;

Exemplo: primeiroNome, mediaAritmetica
 ou valorTotalFinal.

- Veja que o segundo (ou terceiro ou ...) nome sempre vem seguido (concatenado) do primeiro nome, sem nenhum espaço, porém com a primeira letra do segundo (terceiro, ...) nome em maiúsculo.

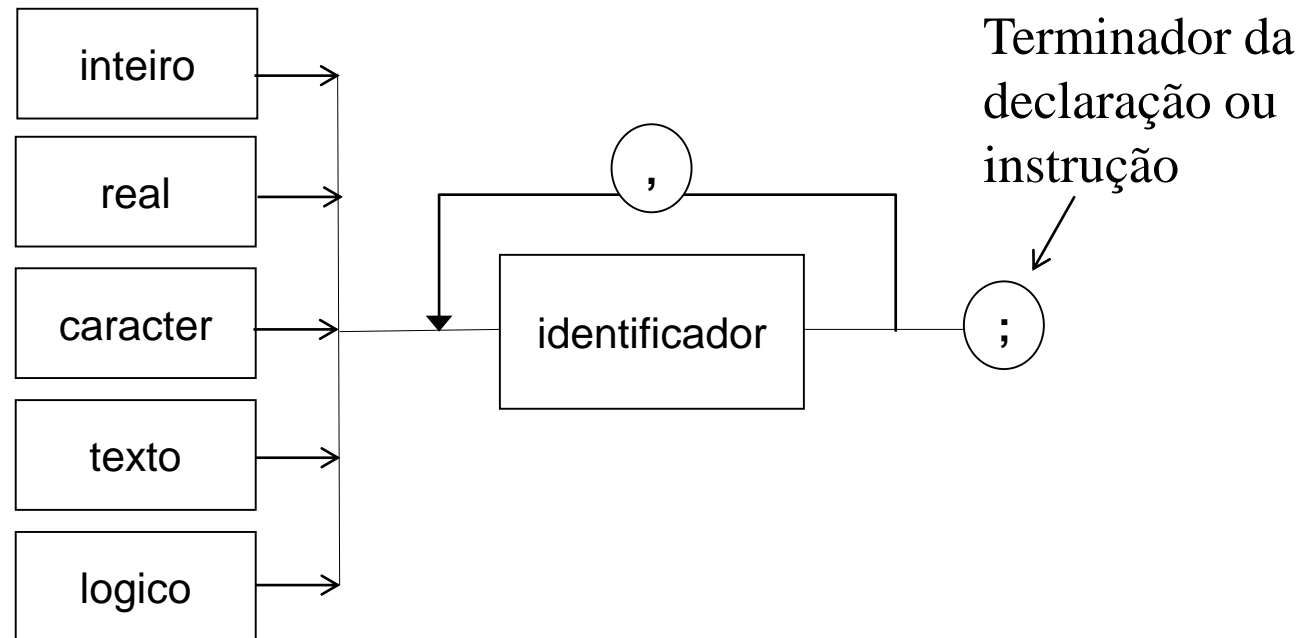
Fundamentos de Algoritmo

Principais Componentes do Sistema de Computação



Fundamentos de Algoritmo

- Todo dado a ser colocado na memória deve ser previamente identificado com valor **significativo**
 - primeiro reconhecer qual o seu tipo de dado adequado e depois fazer o seu armazenamento
- A **declaração de variável** corresponde a definição do lugar na memória que receberá o nome ou apelido da variável (identificador) marcada com o seu tipo
- Sintaxe:



Fundamentos de Algoritmo

Exemplos:

inteiro *x1*, *num_Correto*, *valorConta*;

real *preco*; ↪ caractere especial aceito, mas evitar

texto *nome*, *enderecoCliente*;

caracter *sexo*;

logico *achou*, *resposta*;

Logo:

- *x1* é um local de memória onde só pode ser armazenado números inteiros por esta declaração
- *nome* é um local de memória que pode armazenar um conjunto alfanumérico, de acordo com a declaração anterior
- na representação acima as expressões em **negrito** correspondem a palavras reservadas em algoritmo

Fundamentos de Algoritmo

OPERADORES

- **ARITMÉTICOS**

+, - (binário ou unário), *, / (/ resultado sempre real)
\ (divisão inteira conhecida como operação **div**)
% ou **mod** (resto da divisão inteira)

- **RELACIONAIS**

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

- **LÓGICOS**

E para conjunção
OU para disjunção
! ou NÃO para negação

- **CARACTERES**

+ (concatenação:
operador só aplicado
sobre texto)



Fundamentos de Algoritmo

- Uso apenas de variáveis numéricas, operadores aritméticos e funções embutidas (definidas na linguagem)
 - **raizQuadrada(x)** → calcula a raiz quadrada de x
 - **abs(x)** → obtém o valor absoluto ou o módulo de x
 - **exp(bs, exp)** → obtém o valor da base elevado ao expoente
 - $\text{valor} = \text{nro1} \setminus \text{nro2}$ → quociente inteiro da divisão
 - $\text{valor} = \text{nro1} \bmod \text{nro2}$ → resto da divisão de inteiros
 - $\text{valor} = \text{nro1} \% \text{nro2}$ → resto da divisão de inteiros
- Precedência dos operadores:
 - + , - (unitários)
 - * , / , \ , % , **mod**
 - + , - (binários)

Cuidado:

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

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

Dica: use sempre parênteses

Fundamentos de Algoritmo

- Uso de funções embutidas na manipulação de símbolos (caracteres ou textos)
 - **limpaTela()** → limpa toda a janela de execução
 - **tamanhoTexto(str)** → retorna a quantidade de caracteres que o texto possui (tamanho da string)
 - **comparaTexto(str1,str2)** → compara textos entre str1 e str2 retornando 0 (zero) para iguais e diferente **>0** ou **<0**
 - **maiusculo(str)** retorna o maiúsculo do texto str
 - **minusculo(str)** retorna o minúsculo do texto str
 - **maiusculoCaracter(y)** retorna o maiúsculo do caractere
 - **minusculoCaracter(y)** retorna o minúsculo do caractere



Fundamentos de Algoritmo

EXPRESSÕES RELACIONAIS

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

$<$, $<=$

$=$

$>$, $>=$,

$<>$

Dica: faça uso de parênteses na elaboração destas expressões



Fundamentos de Algoritmo

EXPRESSÕES LÓGICAS

- 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>nao</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 \Rightarrow nao > e > ou

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

Fundamentos de Algoritmo

PRECEDÊNCIA GERAL

Primeiro: parênteses e funções

Segundo: expressões aritméticas

+, - (unitários)

*, /, \, %, **mod**

+, - (binários)

Terceiro: comparações

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

Quarto: nao, !

Quinto: e

Sexto: ou

Fundamentos de Algoritmo

EXPRESSÕES LÓGICAS DE 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 provenientes da matemática)



Comandos Básicos

ATRIBUIÇÃO

- **Instrução de atribuição:** atribuir um valor a uma variável ou constante
- Sintaxe:



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



Exercício Proposto

- O que está errado nos seguintes comandos de atribuição:

// Declarações

logico aux, beta;

real salario, dote;

inteiro minuto;

// Instruções

salario = aux + 100;

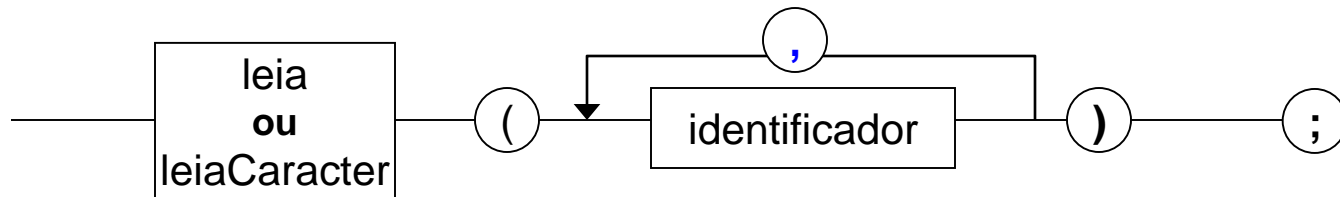
salario = aux == beta;

salario e dote = 24.5;

minuto \leftarrow aux;

Comandos Básicos

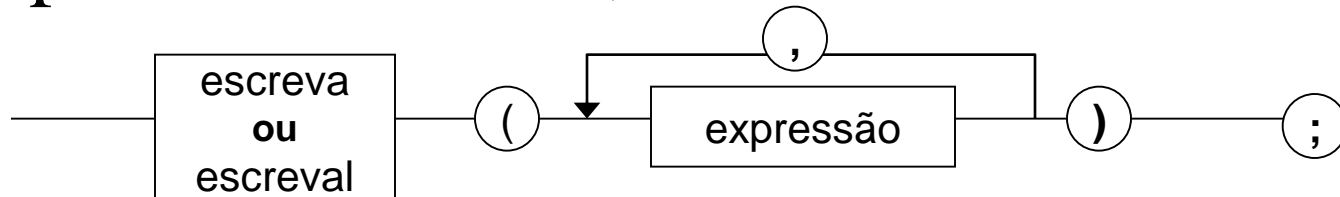
- **Instrução de leitura de dispositivos de entrada:** ler um valor e atribuir a uma variável com tipo de dado coerente ao valor lido;



- Exemplo
 - leia (idade); → faz leitura de variáveis numéricas e texto
 - leia (nome, notaFinal); → várias leituras de variáveis
 - leiaCaracter(sexo); → leitura de variáveis caracter
- Observação: a sintaxe em alguns livros considera a expressão **ler** como comando (ou instrução)

Comandos Básicos

- **Instrução de saída para dispositivos de saída:** escreve o valor de uma variável, constante ou texto orientador em um dispositivo de saída;



Exemplo:

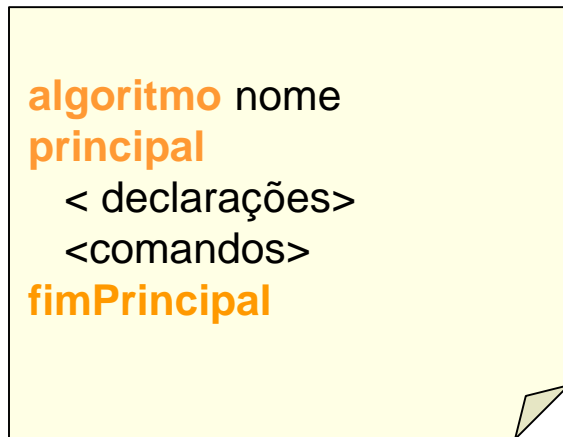
- escreva (idade);
- escreva ("Você pesa", peso , "quilos");
- escreval (x + 3); \Rightarrow este comando escreve e salta para próxima linha na tela de execução

- Observação: a sintaxe em alguns livros adota a instrução imprima ou escrever

Comandos Básicos

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 e usadas;



Português Estruturado

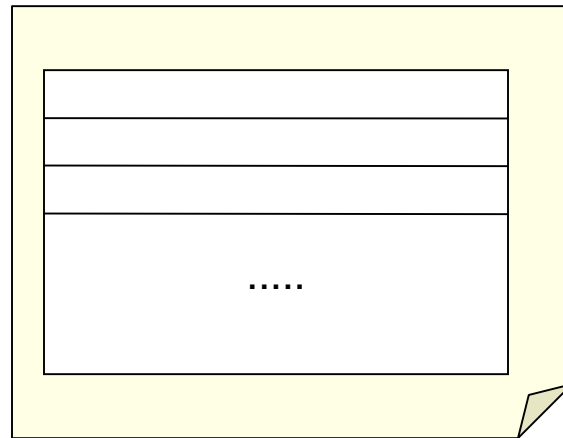
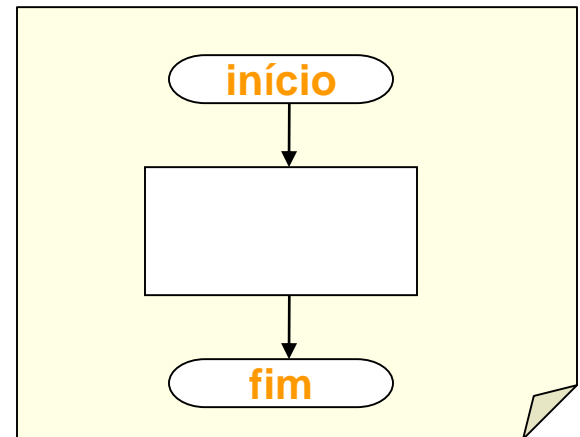


Diagrama de Chapin

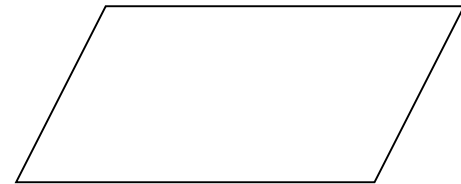
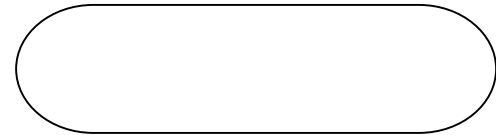


Fluxograma

Fluxograma

FIGURAS PARA ESTRUTURA SEQUENCIAL

- Início e Fim
- Instruções de processamento
- Instruções de entrada de dados
- Instruções de saída de dados
- Conector (sequência do diagrama)



Comandos Básicos

LÓGICA DAS ESTRUTURAS DE CONTROLE

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

Comandos Básicos

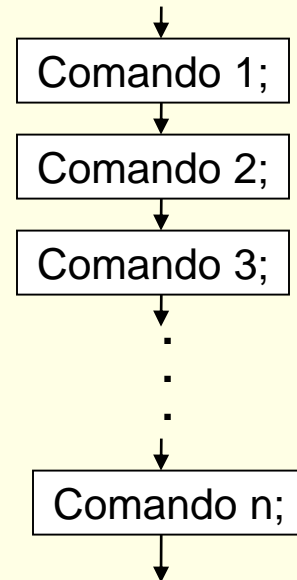
ESTRUTURA SEQUENCIAL

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

Português Estruturado

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

Diagrama de Chapin



Fluxograma

Comandos Básicos

CONSTANTES

- Objeto que não está sujeito a variações, ou seja, é sempre constante e não sofre alteração no seu conteúdo
- O nome dado a uma constante também é um **identificador**
- As constantes respeitam as definições dos tipos de dados com as variáveis: **inteiro**, **real**, **logico**, **caracter**, **texto**.

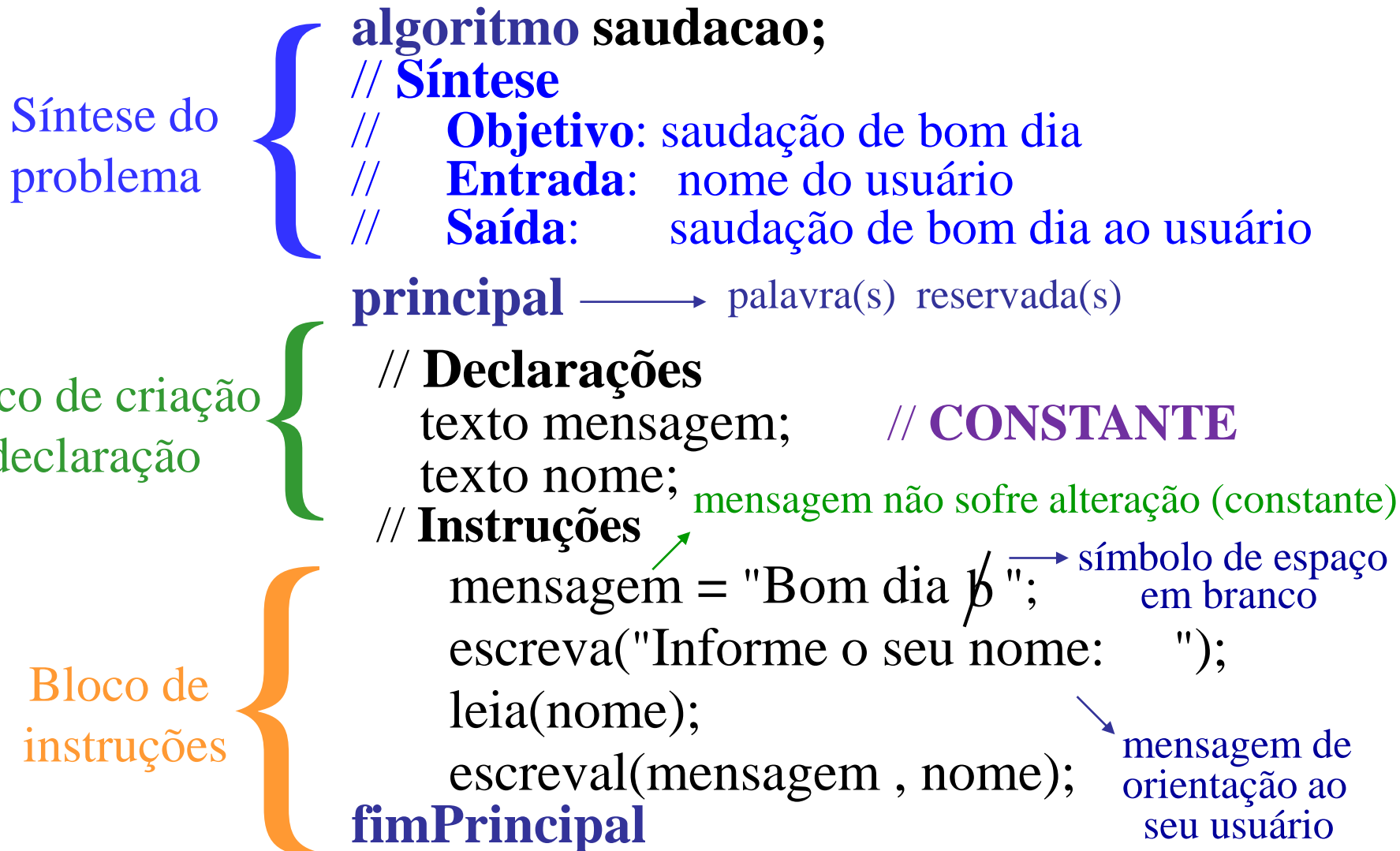
Exemplos: **Numéricas:** valores **reais** ou **inteiros**, onde o ponto final ‘.’ separa a parte inteira da decimal (fracionária). Exemplo: -0.59 ou 2.0 ou -34.597 ou -100

Lógicas: as palavras **verdadeiro** ou **falso**, sendo denominadas constantes *booleanas* do tipo de dado **logico**

Literais: **caracter** ou **texto** com símbolos aceitos na linguagem (algarismos especiais, letras e números). São definidas por apostrofes (caracter) ou aspas (texto):
"algoritmo" ou 'M' ou "José" ou "Sala B-110"

Exercício Proposto

- Construa um algoritmo que obtenha o nome do usuário e o desejo BOM DIA.



Metodologia de Desenvolvimento

Método para elaboração 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** necessárias na especificação do problema para sua solução

Metodologia de Desenvolvimento

- 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



Exercícios de Fixação

1. Construa um algoritmo que leia um número inteiro de horas e mostre ao usuário a quantos minutos e quantos segundos estas horas correspondem.
2. Elabore um algoritmo que calcule a área de uma circunferência cujo raio é fornecido pelo usuário (formula desta área = $3.14 \times \text{raio}^2$)
3. Faça um algoritmo em Português Estruturado que calcule a média aritmética de 3 números quaisquer fornecidos pelo usuário e também seu **Fluxograma**.



Metodologia de Desenvolvimento

TESTE DE ALGORITMO - chinês

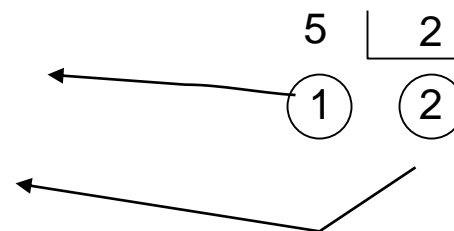
- ✓ Verificar a execução do algoritmo por meio da atribuição de valores de entrada (**valores supostos**)
 - Conhecido também como o processo do **Chinês** (ou chinêsinho) e **Teste de Mesa**
- ✓ Verificação importante para constatar e analisar que o raciocínio do programador foi elaborado corretamente no algoritmo desenvolvido. Este método consiste no teste do algoritmo.
- O programador se coloca no lugar do computador e segue realizando todas as instruções do algoritmo, respeitando sua sequência e lógicas de funcionamento.

Exercícios de Fixação

4. Construa um algoritmo que a partir da leitura de dois números forneça o resto e o quociente da divisão do primeiro pelo segundo número.
- Lembre-se das expressões aritméticas:
 - $\text{mod } (\%) \rightarrow m \text{ mod } i$: resto da divisão de m por i
 - $\backslash \rightarrow m \text{ div } i$: quociente da divisão de m por i

Exemplo: $5 \% 2$ (ou $5 \text{ mod } 2$) = 1

$$5 \backslash 2 = 2$$



Este algoritmo terá duas formas de representação, ou seja, faça uma solução no **Portugues Estruturado** e **Fluxograma**.



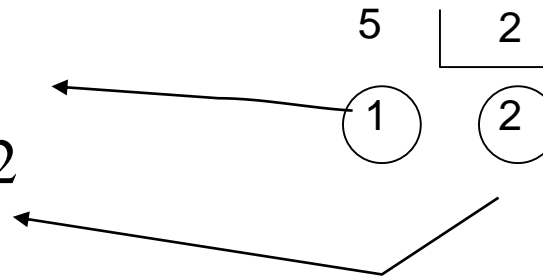
Comandos Básicos

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

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



Referências de Criação para Apoio ao Estudo

Material para Consulta e Apoio ao Conteúdo

- FARRER, H. *et al.* Algoritmos Estruturados, Editora LTC, 3ª . edição, 1999.
 - Capítulo 1
- MANZANO, J.; OLIVEIRA, J. Algoritmos, Lógica para desenvolvimento de programação, Editora Ética, 1996.
 - Capítulo 3
- Universidade de Brasília (UnB Gama)
 - <https://sae.unb.br/cae/conteudo/unbfga>
(escolha a disciplina **Algoritmo Prog Computadores**)

