



**INSTITUTO FEDERAL**  
Sul-rio-grandense

Campus  
Gravataí

# Lógica de Programação e Algoritmos

Prof. Me. Hunder E. Correa Junior



Curso Superior de Tecnologia em  
**Análise e Desenvolvimento**  
de Sistemas

# Uso da Informação



# Dados

- A elaboração de qualquer algoritmo seria impossível se desconsiderasse a estrutura de dados envolvida, pois qualquer computador processa informações que são inseridas em sua memória, no formato de **dados**.

# Exemplo de Dados

- Você tem um nome (muitas pessoas tem um nome e um sobrenome), uma data de nascimento, peso, altura, nacionalidade e etc. Tudo isso são **dados**.
- **Nome**
- **Telefone**
- **CEP**
- **CPF**



# Tipos de Dados

- **Tipos de Dados Primitivos**
- Toda variável pertence a um tipo de dado que define o conjunto de valores que ela pode receber (armazenar), assim como o conjunto de operadores que podem trabalhar sobre a mesma.
- 
- Tipos primitivos são os tipos de dados predefinidos pela linguagem de programação que se está utilizando. Iremos assumir que nossa linguagem trabalha com os seguintes tipos de dados: Inteiro, Real, Caracter e lógico.

- **Tipos Numéricos**
- **Inteiro (integer):** qualquer valor pertencente ao conjunto dos números inteiros (são os valores exatos, sem casas decimais). Exemplo: 1; 5; 0; -3
- **Real:** qualquer valor pertencente ao conjunto dos números reais (são os valores fracionários). Exemplos: -2.3; 5.6; 4.0
- **String:** seqüência de caracteres contendo letras, números e caracteres especiais. São sempre delimitados entre apóstrofos ( ' ' ). Este tipo de dado pode armazenar de 1 até 255 caracteres. Se a definição do tipo utilizado for **String**, será reservado na memória um espaço de 255 caracteres. Desejando reservar um espaço 10 caracteres, a referência deverá ser feita da seguinte forma: **String[10]**.
- **Lógico (Boolean):** comporta apenas dois tipos de valores: verdadeiro (*true*) ou falso (*false*).

# Exemplos de Tipos de Dados

- **Tipo *inteiro***
- Ex.: 1200; -12; 0; etc
- **Tipo *real***
- Ex.: 1200,00; -12,88; 0; etc
- **Tipo *Literal***
- Ex.: "MARIA JOSE"; "Recife"; "123";  
"NOMES"



# Identificadores

Os nomes escolhidos para rotular as variáveis devem obedecer as seguintes regras:

- Primeiro caractere deve ser uma letra
- Nomes devem ser formados por caracteres pertencentes ao seguinte conjunto: {A,B,...,Z, 0,1,...,9, \_ }
- Os nomes escolhidos devem explicitar seu conteúdo
- Não podem ser utilizados símbolos, acentos, espaços, etc.



# EXERCÍCIOS

Assinale com um X os identificadores válidos:

- ☐ !temp
- ☐ "ALUNO"
- ☐ #55
- ☐ KM/L
- ☐ CADILAC
- ☐ AB\*C
- ☐ maior
- ☐ nome
- ☐ B52
- ☐ max num
- ☐ soma\_total
- ☐ nota
- ☐ begin
- ☐ 2cont
- ☐ média



# ***EXERCÍCIOS:***

Diga o tipo de dado:

- 2.0
- -3
- 'Bom Dia'
- -52.5
- $4+5 = 9?$
- 174
- 'Olá!'
- hoje é terça-feira?
- "Brasil"
- '26.8'
- $3+2 = 6?$
- '142'



# Declaração de Variáveis

- Declarar uma variável significa reservar uma "gaveta" na memória, dar-lhe um rótulo. O tipo de dados com o qual a variável é declarada serve para especificar qual é o conjunto de valores que ela pode armazenar.

Exemplo:

DECLARAÇÃO DE VARIÁVEIS  
total: INTEIRO

# Declaração de Constantes

- Uma constante é determinado valor fixo que não se modifica ao longo do tempo, durante a execução do algoritmo. Por exemplo, se tivermos quatro avaliações durante o curso, a média aritmética das notas será dada pela expressão:  $(n1+n2+n3+n4)/4$
- As notas são variáveis, dependem do desempenho do aluno, mas o número de avaliações é constante.

Exemplo:

DECLARAÇÃO DE CONSTANTES

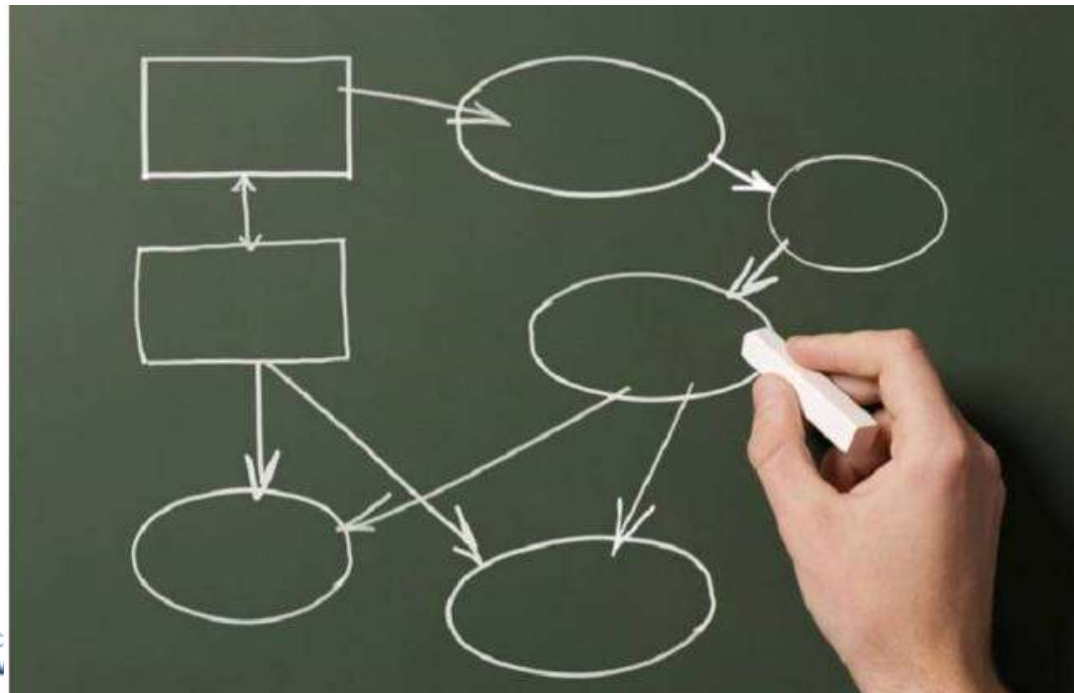
Num\_avaliacoes = 4;

# Escrevendo Algoritmos



# O que é um Algoritmo ?

“Conjunto das regras e procedimentos lógicos perfeitamente definidos que levam à solução de um problema em um número finito de etapas.”



# Exemplo de um Algoritmo:

## Algoritmo "Trabalhar pela manhã"

1. Acordar
2. Tomar banho
3. Vestir-se
4. Tomar café
5. Tirar o carro da garagem
6. Ir para o trabalho



# Regras para Construção de Algoritmos

- Ao montar um algoritmo é preciso dividir o problema apresentado em três fases fundamentais:





- Onde temos:
- **ENTRADA**: são os dados de entrada do algoritmo.
- **PROCESSAMENTO**: são os procedimentos utilizados para chegar ao resultado final (operações, comparações, etc.).
- **SAÍDA**: são os dados já processados (resultado).



# Regras para Construção de Algoritmos

Além disso, temos de lembrar que:

- 1º) a ordem das instruções é importante;
- 2º) não existe processamento sem entrada de dados;
- 3º) tem de haver saída de dados;
- 4º) o número de passos (instruções) é finito e **sempre** chega ao fim.

# Exemplo

- Imagine o seguinte problema: **calcular a média final dos alunos da 3ª Série. Os alunos realizarão quatro provas: P1, P2, P3 e P4.**
- Onde:
- **Média Final = (P1 + P2 + P3 + P4)/4**
- 1) Quais são os dados de entrada?
- *Resposta:*
- 2) Qual será o processamento a ser utilizado?
- *Resposta:*
- 3) Quais serão os dados de saída?

# Método para construção de algoritmos

1. Ler atentamente o enunciado
2. Retirar do enunciado a relação das entradas de dados
3. Retirar do enunciado a relação das saídas de dados
4. Determinar o que deve ser feito para transformar as entradas especificadas nas saídas determinadas
5. Construir o algoritmo
6. Executar o algoritmo (teste de mesa)

# Exemplo Algoritmo

Dados 2 números digitados via teclado, faça um algoritmo capaz de exibir a soma deles no vídeo.

**Algoritmo** soma;

**Declare**       X:Inteiro;  
                  Y:Inteiro;  
                  SOMA:Inteiro;

**Inicio**

**Leia** X;

**Leia** y;

SOMA := X + Y;

**Escreva** SOMA;

**Fim**

# Teste

- Você tem 8 esferas. Todas, **exceto uma**, têm o mesmo peso e você precisa descobrir qual a mais pesada. Para isto você pode utilizar uma balança, mas somente duas vezes.
- Resposta ???????

Dividindo as bolas em 3 grupos, põe 3 bolas em um prato e 3 em outro prato, se der desnível, pega o lado mais pesado e pesa uma bola em cada prato, se não der desnível, o que ficou de fora é o mais pesado.

Se na tentativa de pesar as 3 bolas não desnivelar, pega as 2 bolas que estavam de fora e pesa, algum dos 2 vai ser o mais pesado.

# Operadores Aritméticos

- Chamamos de operadores aritméticos o conjunto de símbolos que representa as operações básicas da matemática.

Operador	Operação
<b>sqrt(n)</b>	<b>Radiciação</b>
<b>pow(num,potencia)</b>	<b>Potenciação</b>
<b>*</b>	<b>Multiplicação</b>
<b>/</b>	<b>Divisão</b>
<b>+</b>	<b>Adição</b>
<b>-</b>	<b>Subtração</b>
<b>%</b>	<b>Resto da divisão</b>

# Operadores Relacionais

Operador	Relação testada
<	Primeiro operando menor que o segundo operando
>	Primeiro operando maior que o segundo operando
<=	Primeiro operando menor ou igual ao segundo operando
>=	Primeiro operando maior ou igual ao segundo operando
==	Primeiro operando igual ao segundo operando
!=	Primeiro operando não é igual ao segundo operando





# Operadores Lógicos

- O que caracteriza uma expressão é o seu resultado. Uma expressão aritmética tem como resultado um número. Uma expressão lógica tem como resultado um valor booleano (**verdadeiro** ou **falso**).
- Em outras palavras, uma expressão lógica é uma pergunta à qual a resposta tem de ser ou verdadeiro ou falso.
- As expressões lógicas são utilizadas quando nossos programas exigem tomadas de decisões.

- AND: Uma ação somente será executada se todas as condições mencionadas forem verdadeiras.

Cond1	Op	Cond2	Resultado
V	AND	V	V
V	AND	F	F
F	AND	V	F
F	AND	F	F



- **OR: Uma ação será executada se pelo menos uma das ações for verdadeira**

Cond1	Op	Cond2	Resultado
V	OR	V	V
V	OR	F	V
F	OR	V	V
F	OR	F	F

- **NOT: Uma ação não será executada se determinada condição não for verdadeira.**

Operador	Cond	Resultado
NOT	V	F
NOT	F	V

- **XOR: Uma ação será executada se uma e somente uma das condições for verdadeira.**

Cond1	Op	Cond2	Resultado
V	XOR	V	F
V	XOR	F	V
F	XOR	V	V
F	XOR	F	F



# Operadores Lógicos no C

&&	<i>E lógico (And)</i>
	<i>Ou lógico (OR)</i>
!	<i>Não (Not)</i>



# Expressões Lógicas

- Em algoritmos existem situações nas quais a execução de um comando depende de uma expressão lógica:
- Ex.  $15 == 14$  ?????? **F**
- $2 == 2$  ??????? **V**
- $3 < 2$  ???????? **F**



- $5 > 3 \ \&\& \ 10 < 1 \ \text{F}$
- $5 > 3 \ || \ 10 < 1 \ \text{V}$





# Prioridades entre operadores lógicos

1. NOT
2. AND , OR
3. XOR

# EXERCÍCIOS

1) Preencha as tabelas verdades:

		OR	AND	NOT	NOT		
X1	X2	S1	S2	X1	X2	S1 AND S2	S1 OR S2
F	F	F	F	V	V	F	F
F	V	V	F	V	F	F	V
V	F	V	F	F	V	F	V
V	V	V	V	F	F	V	V

2) Marque V ou F para as questões abaixo:

$$2 == (1+1)$$

$$5 != 5$$

$$7 > (4+3)$$

$$27 < 35$$

$$9 >= 9$$

$$21 <= 12 \quad || \quad 9 == 3 * 3$$

$$100 != 10 \quad \&\& \quad 10 >= 1$$

# Estruturas de Controle

- Ao se escrever um algoritmos, devemos descrever os passos necessários a solução, para isto utilizamos **estruturas de controle**.
- Estas estruturas refere-se à ordem em que instruções, expressões e chamadas de função são executadas ou avaliadas em programas de computador. Em outras palavras, controlam o fluxo de execução do algoritmo ou programa.

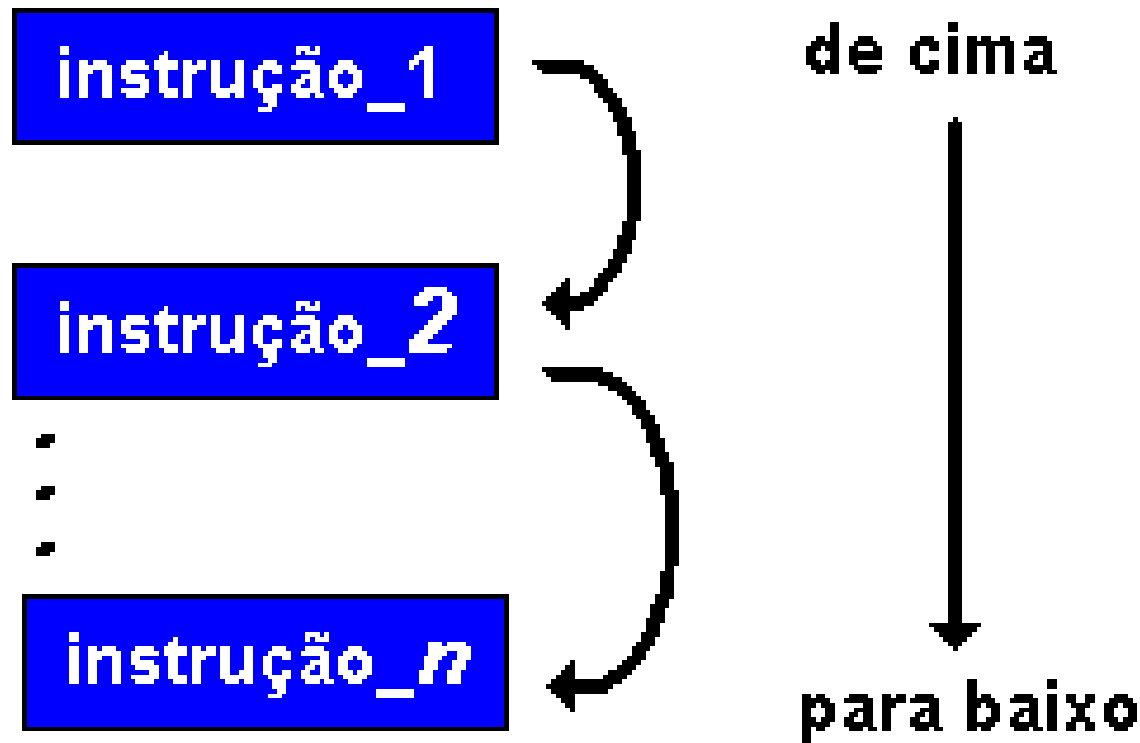
# Estruturas de Controle

- A principais estrutura de Controle são:
- Estruturas Sequenciais
- Estruturas de Seleção (simples e composta)
- Estruturas de Repetição



# Estruturas Sequenciais

## Seqüencial



# Sequência Simples

- Sequência simples é um conjunto de ações primitivas que serão executadas numa sequência linear de cima para baixo, ou seja, na ordem em que foram escritas.
- As ações primitivas sempre devem ser encerradas por um ponto e vírgula (;).

# Exemplo: Sequência

ALGORITMO

DECLARAÇÃO DE CONSTANTES

DECLARAÇÃO DE VARIÁVEIS

INÍCIO

Ação primitiva 1;

Ação primitiva 2;

FIM;

FIM.



# Conceito de Bloco

Um bloco pode ser definido como uma sequência de ações primitivas, que como um todo, possui uma função bem definida, neste caso um algoritmo pode ser visto como um bloco.

# Exemplo Bloco

Algoritmo <nome> { Nome do algoritmo em desenvolvimento}

Declaração de variáveis e constantes

INICIO {Início do algoritmo}

Inicio {inicio do bloco}

Sequência de ações primitivas

Fim {Fim do bloco}

FIM {Fim do algortmo}



# Formas de Organizar um programa – **Indentação**

- A **Indentação** consiste em recuar para a direita determinadas linhas do programa, através da tecla espaço ou tabulação.
- Este espaçamento é necessário para uma boa visualização e compreensão do programa, principalmente quando se utiliza comandos encadeados, pois a correta indentação faz com que os eles não fiquem misturados.

# Exemplo Indentação

**Algoritmo** CESGRANRIO

**Var** A, R: **real**

NOME: **literal**[32]

**Início**

**Leia** A, NOME

**Escolha**

**Caso** NOME = "João"

R  $\leftarrow$  5 \* A

**Caso** NOME = "Maria"

R  $\leftarrow$  2 \* A

**Senão**

R  $\leftarrow$  A

**Fim\_escolha**

A  $\leftarrow$  3 + A

**Escreva** Resultado: ",R

**Fim**

Indentação



# Comandos Básicos de um Algoritmo

- **LEIA X;** (recebe um valor em uma variável X)
- **Y=(x\*2);** (atribuição de um cálculo à uma variável Y)
- **Escreva Y;** (Exibe o valor da variável Y)

# Exercícios

1. Faça um algoritmo que leia 3 números inteiros e escreva o produto desses números.
2. Faça um algoritmo que leia a idade e o nome de três alunos e calcule a media de idades deles, no final escreva a media.
3. Faça um algoritmos que leia a altura e o comprimento de um retângulo. No final, escreva a área deste retângulo.
4. Faça um algoritmo que leia o nome de um aluno, as notas da etapa 1 e etapa 2 de um aluno e calcule a media, sendo que a etapa 2 tem peso 2.
5. Um motorista deseja colocar no seu tanque X reais de gasolina. Escreva um algoritmo para ler o preço do litro da gasolina e o valor do pagamento, e exibir quantos litros ele conseguiu colocar no tanque.



# Respostas

1)

Algoritmo um

Var n1,n2,n3,produto :int

Inicio

Leia n1,n2,n3

produto=n1\*n2\*n3

Escreva produto

FIM

# Respostas

2)

Algoritmo dois

Var idade1,idade2,idade3,media :int

Var nom1,nom2,nom3 :char

Inicio

Leia idade1,idade2,idade3

Leia nom1,nom2,nom3

media=(idade1+idade2+idade3)/3

Escreva media

FIM



# Respostas

3)

Algoritmo tres

Var altura,comprimento,area :real

Inicio

Leia altura,comprimento

area=altura\*comprimento

Escreva area

FIM

# Respostas

4)

Algoritmo quatro

Var nome :char

Var etapa1,etapa2,media :real

Inicio

    Leia nome

    Leia etapa1,etapa2

$media = (etapa1 + (etapa2 * 2)) / 3$

    Escreva media

FIM

# Respostas

5)

Algoritmo cinco

Var preco, valor :real

Var litros :int

Inicio

    Leia preco,valor

    litros=valor/preco

    Escreva litros

FIM