

# Algoritmos e Lógica de Programação II

## Revisão

Prof. MSc. Rafael Staiger Bressan  
[rafael.bressan@docentes.unicesumar.edu.br](mailto:rafael.bressan@docentes.unicesumar.edu.br)

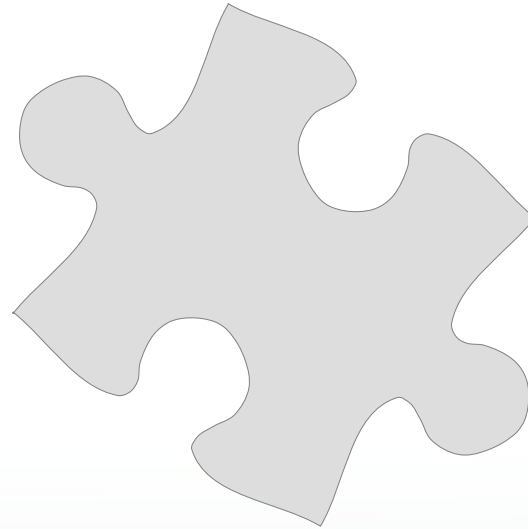


# Cronograma

- **Introdução**
- **Conceituando Algoritmos**
- **Como Construir Algoritmos**
- **Tipos de Algoritmos**
- **Estudando Variáveis**
- **Tipos de Variáveis**
- **Constante**
- **Expressões**
- **Funções Intrínsecas**
- **Atribuições**
- **Entrada de Dados**
- **Saída de Dados**
- **Construindo Algoritmos**
- **Considerações Finais**

# Introdução

## Algoritmo e Lógica de Programação





# Introdução

- O que é Informática?

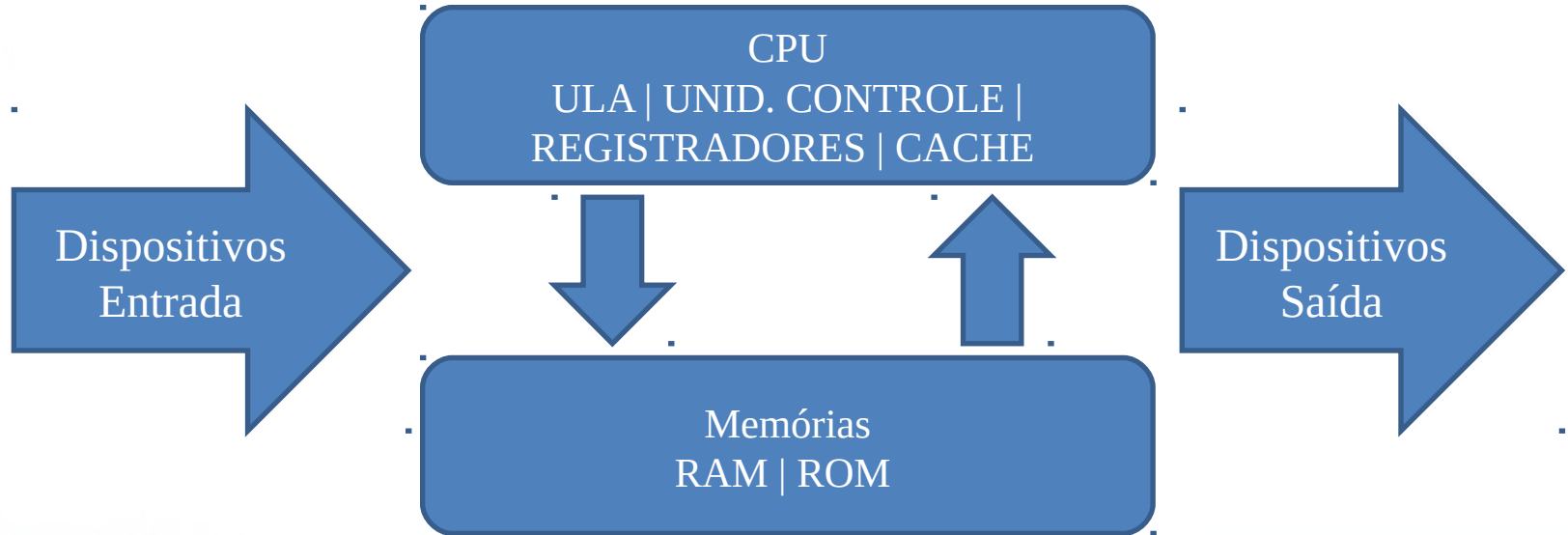
- Informação + Automática. (Século XXI)
- Processamento da informação por meios automáticos em simples impulsos **binários**.

- Computador

- Equipamento que faz o Tratamento **automático** de **informações** ou processamento de dados. (1550-1617, John Napier)
- Classificação:
  - Capacidade de armazenar informação durante o processamento e para uso posterior;
  - Capacidade de processamento das informações;

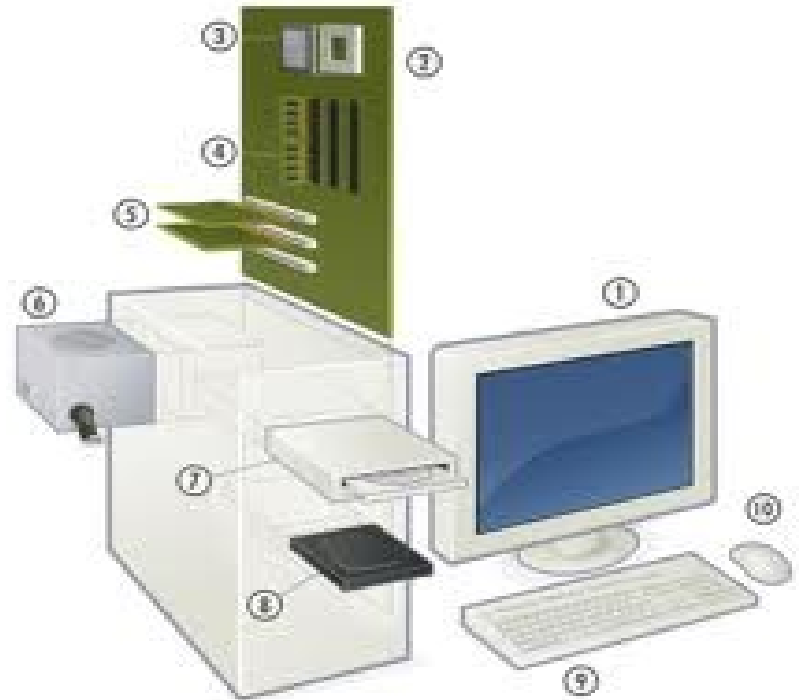
# Introdução

## Unidades Básicas do Computador



# Introdução

## Hardware | Software





# Introdução

## Hardware | **Software**

- Toda Hardware necessita de um **Software** que o gerencie.
  - Software é a parte lógica do computador.
    - “**Sequência de instruções a serem seguidas e/ou executadas**, na manipulação, redirecionamento ou modificação de um dado/informação ou acontecimento.”



# Introdução

## Tabela de bits / Bytes

1 bit	0 ou 1 (b)
8 bits	1 Byte (B)
1024 Bytes	1 Quilobyte (KB)
1024 Quilobyte	1 Megabytes (MB)
1024 Megabytes	1 Gigabytes (GB)
1024 Gigabytes	1 Terabyte (TB)
1024 Terabyte	1 Petabyte (PB)





# Introdução

## Conceitos Básicos

- Quando queremos criar ou **desenvolver um software** para realizar determinado tipo de processamento de dados, devemos escrever um programa ou vários programas interligados. No entanto, para que o computador compreenda e execute esse programa, devemos escrevê-lo usando uma linguagem que tanto o computador quanto o criador de software entendam.

# Introdução

## Existe Lógica no dia a dia?



Disponível em: <http://ospontosdevista2.blogspot.com/2018/02/imagens-do-facebook-preguica-e-mae-de.html>  
Acesso em: 09 jul. 2019



# Introdução

## Sequência Lógica

- Sequência Lógica
  - Sequência lógica são **passos executados até atingir um objetivo** ou solução de um problema.
  - Exemplos
    - Receita de um bolo
    - Manual montagem

# Desafio

## ”O Lobo e a Ovelha”



<https://rachacuca.com.br/jogos/o-lobo-e-a-ovelha/>

- O barquinho do camponês comporta apenas um item, além dele próprio. O barquinho pode levar e trazer itens
- Você deve ficar atento às seguintes regras:
  - **O lobo devora a ovelha se os dois ficarem sozinhos e;**
  - **A ovelha come o couve se ficar sozinha com ele.**

# Desafio

## ”Missionários e Canibais”



- Os missionários não podem ficar em menor número, em qualquer uma das margens, ou os canibais os devorarão.

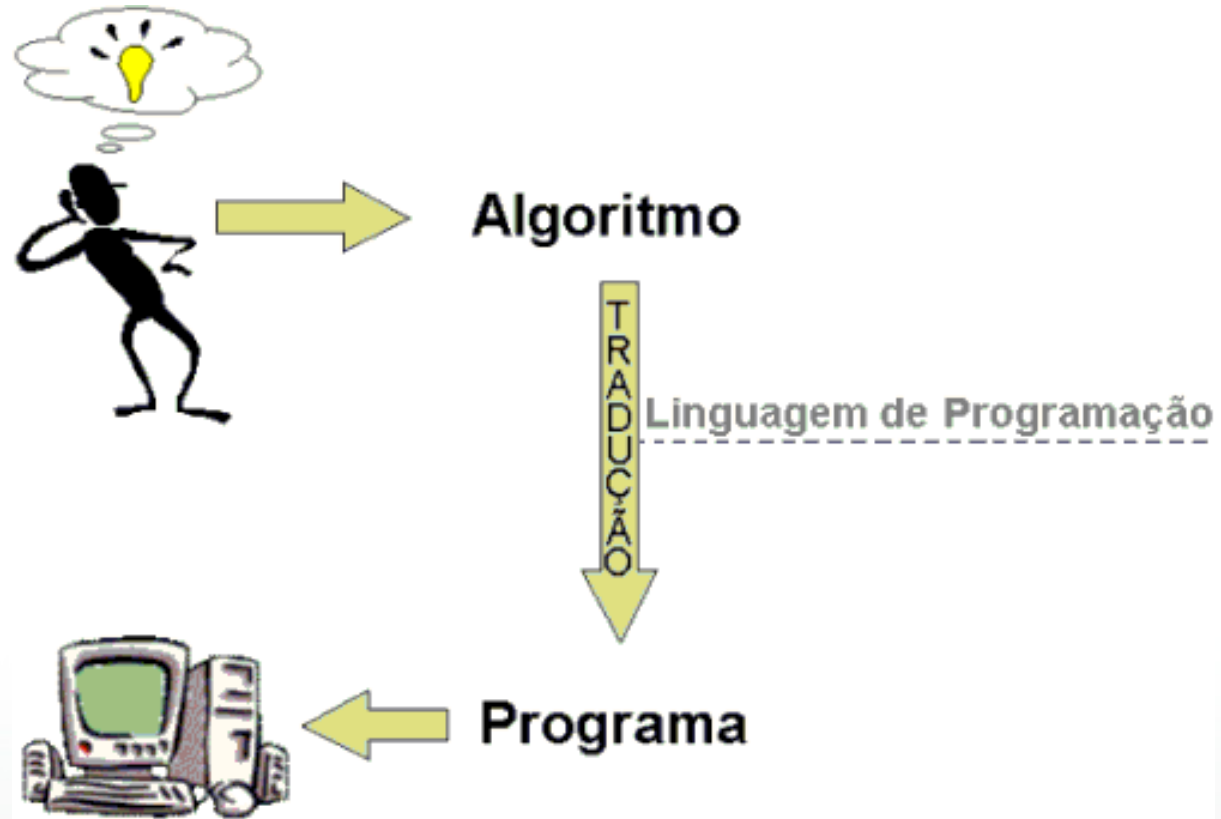
<https://rachacuca.com.br/jogos/missionarios-e-canibais/>



# Conceituando Algoritmos

- As etapas para o desenvolvimento de um programa são:
  - **Análise** –Estuda-se o enunciado do problema para definir os **dados de entrada**, o **processamento** e os **dados de saída**.
  - **Algoritmo** – Ferramentas do tipo descrição narrativa, **fluxograma** ou **português estruturado** são utilizadas para descrever o problema com suas soluções.
  - **Codificação** – O algoritmo é transformado em **códigos da linguagem de programação** escolhida para se trabalhar.

# Conceituando Algoritmos







# Conceituando Algoritmos

- “Algoritmo é uma sequência de passos que visa atingir um objetivo bem definido.” (FORBELLONE, 1999)
- “Algoritmo é uma sequência finita de instruções ou operações cuja execução, em tempo finito, resolve um problema computacional, qualquer que seja sua instância.” (SALVETTI, 1999)
- “Algoritmo são regras formais para a obtenção de um resultado ou da solução de um problema, englobando fórmulas de expressões aritméticas.” (MANZANO, 1997)





# Conceituando Algoritmos

- “Ação é um acontecimento que, a partir de um estado inicial, após um período de tempo finito, produz um estado final previsível e bem definido. Portanto, um algoritmo é a descrição de um conjunto de comandos que, obedecidos, resultam numa sucessão finita de ações.” (FARRER, 1999)
- ***“Informalmente, um algoritmo é qualquer procedimento computacional bem definido que toma algum valor ou conjunto de valores como entrada e produz algum valor ou conjunto de valores como saída. Portanto, um algoritmo é uma sequência de passos computacionais que transformam a entrada na saída.” (CORMEN, 2002)***



# Conceituando Algoritmos

- Analisando as definições percebemos que executamos no dia a dia vários algoritmos.

# Conceituando Algoritmos

## **Algoritmo 1 - Trocar uma lâmpada queimada**

- Passo 1 - Pegar uma escada;
- Passo 2 - Posicionar a escada debaixo da lâmpada;
- Passo 3 - Buscar uma lâmpada nova;
- Passo 4 - Subir na escada;
- Passo 5 - Retirar a lâmpada queimada;
- Passo 6 - Colocar lâmpada nova;





# Conceituando Algoritmos

## **Algoritmo 2 – Somar três números.**

- Passo 1 – Receber os três números;
- Passo 2 – Somar os três números;
- Passo 3 – Mostrar o resultado da soma.

# Conceituando Algoritmos

## **Algoritmo 3 – Fazer um lanche.**

- Passo 1 – Pegar o pão;
- Passo 2 – Cortar o pão ao meio;
- Passo 3 – Pegar a maionese;
- Passo 4 – Passar a maionese no pão;
- Passo 5 – Pegar e cortar alface e tomate;
- Passo 6 – Colocar alface e tomate no pão;
- Passo 7 – Pegar o hambúrguer;
- Passo 8 – Fritar o hambúrguer;
- Passo 9 – Colocar o hambúrguer no pão;



# Conceituando Algoritmos

## **Algoritmo 4 – Sacar dinheiro no banco.**

- Passo 1 – Ir até o banco 24h;
- Passo 2 – Colocar o cartão;
- Passo 3 – Digitar a senha;
- Passo 4 – Solicitar a quantia desejada;
- Passo 5 – Se o saldo for maior ou igual a quantia desejada, sacar; senão mostrar mensagem de impossibilidade de saque;
- Passo 6 – Retirar o cartão;
- Passo 7 – Sair do banco 24h.





# Conceituando Algoritmos

- Você pode estar pensando: **“Mas eu realizo essas atividades de maneira diferente!”**
  - Esse pensamento é correto, pois às vezes um problema pode ser resolvido de diversas maneiras, porém, gerando a mesma resposta, ou seja, podem existir vários algoritmos para solucionar o mesmo problema.





# Como Construir Algoritmos

- Para a construção de qualquer tipo de algoritmo, é necessário seguir estes passos:
- **Compreender completamente o problema** a ser resolvido, destacando os pontos mais importantes e os objetos que o compõem.
- Definir os **dados de entrada**, ou seja, quais dados serão fornecidos.
- **Definir o processamento**, ou seja, quais cálculos serão efetuados. O processamento é responsável pela transformação dos dados de entrada em dados de saída.
- Definir os **dados de saída**.
- Construir o algoritmo e testar o algoritmo realizando simulações.





# Tipos de Algoritmos

- Os três tipos mais utilizados de algoritmos são:
  - **Descrição Narrativa**
  - **Fluxograma**
  - **Pseudocódigo ou Portugol ou Português Estruturado**



# Tipos de Algoritmos

## **Descrição Narrativa:**

- Consiste em analisar o enunciado do problema e escrever, utilizando uma linguagem natural (por exemplo, a língua portuguesa), os passos a serem seguidos para sua resolução.
- **Vantagem:** Não é necessário aprender nenhum conceito novo, pois é a linguagem natural.
- **Desvantagem:** A língua natural abre espaço para várias interpretações;



# Tipos de Algoritmos

Descrição Narrativa - Exemplo: **Algoritmo 2 – Somar três números.**

- Passo 1 – Receber os três números;
- Passo 2 – Somar os três números;
- Passo 3 – Mostrar o resultado.









# Tipos de Algoritmos

## Fluxograma

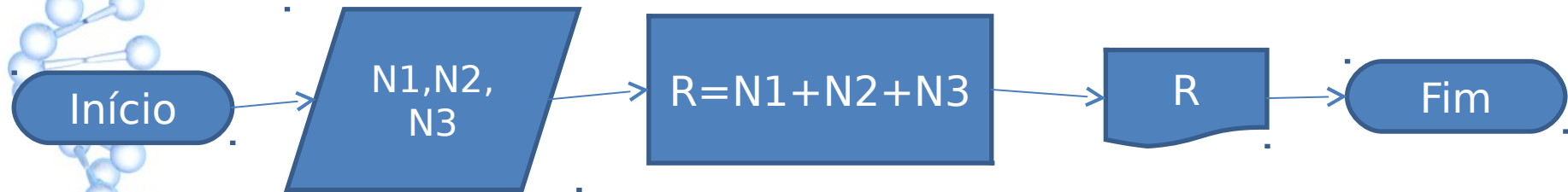
- **Utiliza símbolos gráficos** predefinidos para a resolução do problema.
- **Vantagem:** O entendimento de elementos gráficos é mais simples que o entendimento de texto;
- **Desvantagem:** É necessário aprender a simbologia dos fluxogramas e, além disso, o algoritmo resultante não apresenta muitos resultados, dificultando sua transcrição para o programa (Codificação)

# Tipos de Algoritmos

Símbolo	Descrição
	Utilizado para indicar o início e o fim do algoritmo
	Utilizado para representar a entrada de dados
	Utilizado para representar a saída de dados
	Utilizado para indicar os cálculos e atribuição de valores
	Utilizado para indicar a tomada de decisão e os possíveis desvios
	Utilizado para indicar o fluxo de dados, conectando os blocos existentes

# Tipos de Algoritmos

## Exemplo Fluxograma – Somar três números





# Tipos de Algoritmos

## Pseudocódigo

- Consiste em analisar o enunciado do problema e escrever, por meio de **regras predefinidas**, os passos a serem seguidos para sua resolução.
- **Vantagem:** A passagem do algoritmo para qualquer linguagem de programação é quase imediata, bastando conhecer as palavras reservadas dessa linguagem que serão utilizadas.
- **Desvantagem:** É necessário aprender as regras do pseudocódigo.





```
algoritmo "Soma"
```

```
// Função : Soma três números inteiros
```

```
// Autor : Prof. Rafael S. Bressan
```

```
// Data : 16/02/2014
```

```
// Seção de Declarações
```

```
var
```

```
    N1,N2,N3,RESULTADO:inteiro
```

```
inicio
```

```
// Seção de Comandos
```

```
    escreva("Digite o valor do N1 = ")
```

```
    leia(N1)
```

```
    escreva("Digite o valor do N2 = ")
```

```
    leia(N2)
```

```
    escreva("Digite o valor de N3 = ")
```

```
    leia(N3)
```

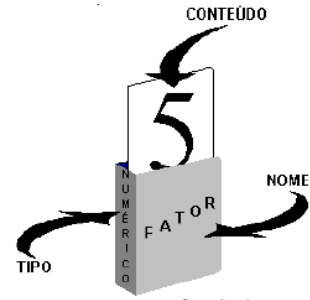
```
    RESULTADO:=N1+N2+N3
```

```
    escreva("Resultado          = ", RESULTADO)
```

```
fimalgoritmo
```



# Estudando Variáveis



- Um algoritmo e, posteriormente, um programa, recebe dados, que são **armazenados na memória**;
- Uma **variável representa uma posição de memória**. Possuindo um **nome** e **tipo**, cujo **conteúdo** pode variar ao longo do tempo, durante a execução de um programa.



# Estudando Variáveis

## Formação de Identificadores

- Os identificadores são os **nomes das variáveis**, dos programas, das constantes, das rotinas etc. As regras básicas são:
- Os caracteres que você pode utilizar são: os números, as letras maiúsculas, as letras minúsculas e o caractere sublinhado.
- O primeiro caractere deve ser sempre uma letra ou caractere sublinhado.
- Não são permitidos espaços em branco e caracteres especiais (@,\$,+,-,%,!).
- Não podemos usar palavras reservadas nos identificadores, ou seja, palavras que pertençam a uma linguagem de programação.



# Exemplos de Identificadores

## Válidos

A  
a  
nota  
numero\_1  
x4

## Não válidos

5b – por começar com número  
E 39 – por conter espaço em branco  
Case – por ser palavra reservada



# Tipos de Variáveis

## Numéricos:

- **Inteiro**

- Podem ser positivos ou negativos e não possuem parte fracionária
- {-23, 98 , 0, -3875, 230}

- **Real**

- Podem ser positivos ou negativos e possuem parte fracionária
- {23,56 | -35,2 | 0,0 | 1 | 52}



# Tipos de Variáveis

## Literais ou Caracteres

- São formados por um caractere ou por uma cadeia de caracteres. Esses caracteres podem ser letras minúsculas, maiúsculas e caracteres especiais. {"aluno" | "\$5nh" | "Universidade" | "1 + 2" | "A"}

## Lógico

- São também chamados de dados **booleanos** e podem assumir os valores FALSO ou VERDADEIRO.



```
algoritmo "Exemplo_Character"
```

```
// Função : Exemplo do uso de variáveis - caracteres
```

```
// Autor : Prof. Rafael S. Bressan - CCET UNOPAR
```

```
// Data : 2014
```

```
// Seção de Declarações
```

```
var
```

```
dias,idade:inteiro
```

```
nome:caracter
```

```
inicio
```

```
// Seção de Comandos
```

```
escreva("Digite seu nome = ")
```

```
leia(nome)
```

```
escreva("Digite sua idade = ")
```

```
leia(idade)
```

```
dias:=idade*365
```

```
escreva("Olá ",nome," , você viveu aproximadamente",dias," dias")
```

```
finalgoritmo
```



# Constantes

- Uma constante armazena dados que não variam com o tempo, ou seja, seu conteúdo é um valor fixo.
- Definição
  - CONST  
`<nome da constante> = <valor>`



# Expressões

Operação	Operador	Prioridade
Soma	+	1 (menor)
Subtração	-	1
Multiplicação	*	2
Divisão	/	2
Exponenciação	**	3 (maior)
Resto	mod	3
Divisão Inteira	div	3





# Expressões

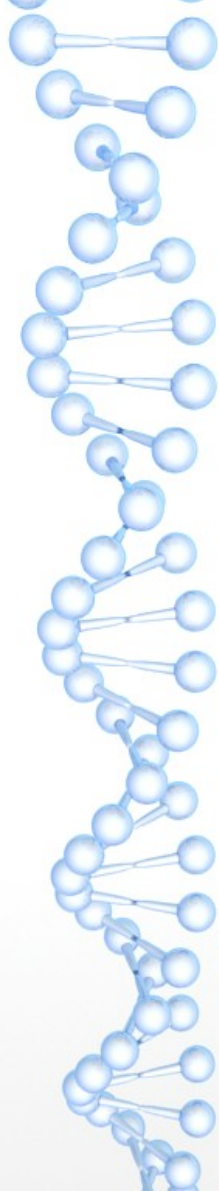
Operação	Operador	Significado
Igual	=	A=B ---- B=A
Diferente	<>	A <> B
Maior	>	A > 5
Menor	<	A < 5
Maior ou igual a	>=	A >= 5
Menor ou igual a	<=	A <= 5

# Expressões

OPERADORES LÓGICOS	PORTUGUÊS ESTRUTURADO	SIGNIFICADO
Multiplicação lógica	E	Resulta <b>VERDADEIRO</b> se ambas as partes forem verdadeiras.
Adição lógica	Ou	Resulta <b>VERDADEIRO</b> se uma das partes é verdadeira.
Negação	Nao	Nega uma afirmação, invertendo o seu valor lógico: se for <b>VERDADEIRO</b> torna-se <b>FALSO</b> , se for <b>FALSO</b> torna-se <b>VERDADEIRO</b> .

A tabela abaixo – chamada **tabela-verdade** – mostra os resultados das aplicações dos operadores lógicos conforme os valores dos operadores envolvidos.

A	B	A E B	A OU B	NÃO A	NÃO B
VERDADEIRO	VERDADEIRO	VERDADEIRO	VERDADEIRO	FALSO	FALSO
VERDADEIRO	FALSO	FALSO	VERDADEIRO	FALSO	VERDADEIRO
FALSO	VERDADEIRO	FALSO	VERDADEIRO	VERDADEIRO	FALSO
FALSO	FALSO	FALSO	FALSO	VERDADEIRO	VERDADEIRO



OPERADOR ARITMÉTICO	PRIORIDADE
Exponenciação	3 (maior)
Multiplicação	2
Divisão	2
Adição	1
Subtração	1 (menor)

OPERADOR LÓGICO	PRIORIDADE
<b>e</b>	3
<b>ou</b>	2
<b>nao</b>	1

OPERADOR	PRIORIDADE
Operadores aritméticos	3
Operadores relacionais	2
Operadores lógicos	1

# Linearização de Expressões

**Exemplo 3.1**

$$\left\{ \left[ \frac{2}{3} - (5-3) \right] + 1 \right\} .5$$

**Tradicional**

$$((2/3 - (5-3)) + 1) * 5$$

**Computacional**



# Funções Intrínsecas

Função	Objetivo
ABS(variável)	Valor Absoluto
ARCTAN(variável)	Arco tangente de um ângulo
COS(variável)	Valor do cosseno de um ângulo
EXP(variável)	Exponencial
FRAC(variável)	Parte Fracionária
LN(variável)	Logaritmo Natural
PI	Valor de PI
SIN(variável)	Seno de um ângulo
SQR(variável)	Elevado ao quadrado
SQRT(variável)	Raiz Quadrada



# Atribuições

- O símbolo utilizado para a atribuição é  $( \leftarrow )$  ou  $( := )$  e a representação é dada por:

Identificador  $\leftarrow$  expressão

Identificador  $:=$  expressão

...

Num  $\leftarrow$  20



# Entrada de Dados

- A entrada de dados (dispositivo de entrada – teclado) permite receber os dados digitados pelo usuário e é realizada por meio do comando (leia)

leia <variável>



# Saída de Dados

- A saída de dados (dispositivo de saída – monitor) permite mostrar os dados ao usuário e é realizada por meio do comando (escreva)

escreva <variável> ou <literal>





```
algoritmo "Exemplo_Character"
```

```
// Função : Exemplo do uso de variáveis - caracteres
```

```
// Autor : Prof. Rafael S. Bressan
```

```
// Data : 2014
```

```
// Seção de Declarações
```

```
var
```

```
dias,idade:inteiro
```

```
nome:caracter
```

```
inicio
```

```
// Seção de Comandos
```

```
escreva("Digite seu nome = ")
```

```
leia(nome)
```

```
escreva("Digite sua idade = ")
```

```
leia(idade)
```

```
dias:=idade*365
```

```
escreva("Olá ",nome,",", "você viveu aproximadamente",dias," dias")
```

```
finalgoritmo
```



# Exercícios

- Defina: **Algoritmo**
- Quais são as etapas para desenvolvimento de um software (Programa)?
- Descreva os três tipos de algoritmos mais usados.
- Descreva como funcionam as declarações de variáveis em algoritmo e de exemplos.

# Exercícios

- Faça a linearização das expressões a baixo.

$$3xy^2 - 2x^2y$$

$$2xy^{-2} - x^3 + 3$$

$$5x\sqrt{y^3} - 11$$

$$7xy^{\frac{2}{3}} - x^{\frac{1}{4}} - 2$$

$$\frac{3m}{x-y}$$

$$\frac{7}{a^{-3}}$$

$$4\left(xy^{\frac{2}{3}}\right)^3$$

$$\frac{x^3y^2 - 2x^4y^3}{\sqrt{2}}$$




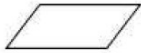




# Leitura Recomendada

- Capítulo 1 e 3
  - ASCENCIO, Ana Fernanda Gomes; VENERUCHI, Edilene Aparecida. **Fundamentos de programação de computadores: algoritmos, Pascal e C/C++**. São Paulo: Prentice Hall, 2002.

# Construindo Algoritmos

## Estrutura Sequencial

- Desenvolva um algoritmo em fluxograma e pseudocódigo que leia o nome e idade de uma pessoa, calcule e apresente na tela o nome digitado e a quantidade aproximada de dias vividos.

Símbolo	Descrição
	Utilizado para indicar o início e o fim do algoritmo
	Utilizado para representar a entrada de dados
	Utilizado para representar a saída de dados
	Utilizado para indicar os cálculos e atribuição de valores
	Utilizado para indicar a tomada de decisão e os possíveis desvios
	Utilizado para indicar o fluxo de dados, conectando os blocos existentes



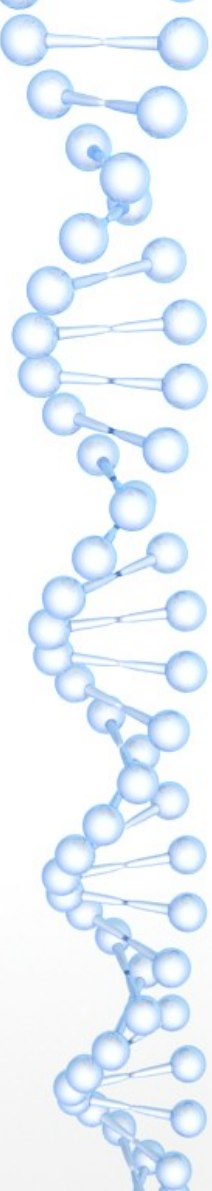
# Referencias

MANZANO, José Augusto NG. **Algoritmos lógica para desenvolvimento de programação de computadores**. Saraiva Educação SA, 2010.

ASCENCIO, Ana Fernandes Gomes; CAMPOS, Edilene Aparecida Veneruchi de. **Fundamentos da programação de computadores: algoritmos, Pascal e C/C++ e Java**. São Paulo: Pearson Prentice Hall, 2010

CORMEN, Thomas H. et al. **Algoritmos: teoria e prática**. Editora Campus, v. 2, p. 2, 2002.

XAVIER, Gley Fabiano Cardoso. **Lógica de programação**. Senac, 2018.



*“Só existem dois dias no ano que nada pode ser feito.  
Um se chama ontem e o outro se chama amanhã”*

Dalai Lama