

Disciplina

# R para Ciência de Dados

## Aula 1

INTRODUÇÃO À LÓGICA DE PROGRAMAÇÃO E ALGORITMOS

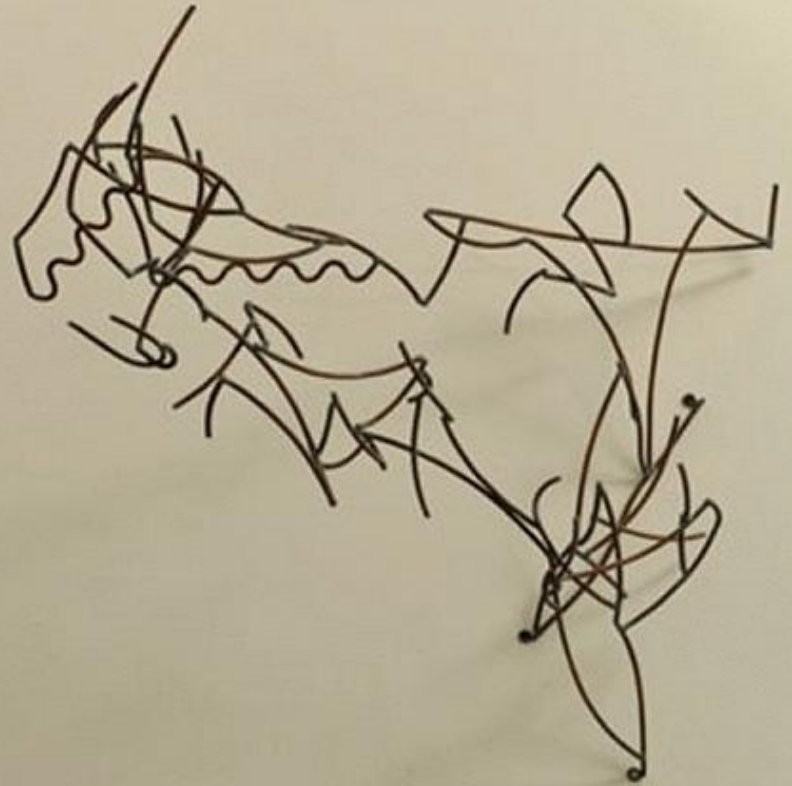
Alan Rodrigo Panosso

Departamento de Ciências Exatas

[alan.panosso@unesp.br](mailto:alan.panosso@unesp.br)



# Noções de Lógica



Então,  
O que é lógica...?

## O que é Lógica

é a análise das **formas e leis do pensamento**, mas não se preocupa com a produção do pensamento, não se preocupa com o conteúdo do pensamento, mas sim com **a maneira pela qual os pensamentos são organizados e apresentados**, possibilitando que cheguemos a uma conclusão por meio do encadeamento dos argumentos.

**A lógica é a ciência que estuda as leis do raciocínio.**

**Correção/validação do pensamento.**

**Encadeamento/ordem de ideias.**

**Arte de bem pensar.**

## Exemplos de aplicação da lógica

O quarto está fechado e meu livro está no quarto.

**Então...**

**...preciso primeiro abrir o quarto para pegar o livro.**

Rosa é mãe de Ana, Paula é filha de Rosa, Júlia é filha de Ana.

**Então...**

**...Júlia é neta de Rosa e sobrinha de Paula.**

Todo mamífero é animal e todo cavalo é mamífero.

**Então...**

**...todo cavalo é animal.**

Todo mamífero bebe leite e o homem bebe leite.

**Então...**

**...todo homem é mamífero e animal (mas não é um cavalo).**

## Resolva os seguintes problemas de lógica:

**P1** – Uma lesma deve subir um poste de 10m de altura. De dia sobe 2m e à noite desce 1m. Em quantos dias atingirá o topo do poste?

**P2** - Três gatos comem três ratos em três minutos. Cem gatos comem cem ratos em quantos minutos?

**P3** - O pai do padre é filho do meu pai. O que eu sou do Padre?

**P4** - Se um bezerro pesa 75 kg mais meio bezerro, quanto pesa um bezerro inteiro?

**P5** – Qual o próximo número da sequência 7, 8, 10, 13, 17?

## Resolva os seguintes problemas de lógica:

**P1** – Uma lesma deve subir um poste de 10m de altura. De dia sobe 2m e à noite desce 1m. Em quantos dias atingirá o topo do poste?

**R1** - 9(nove) dias

**P2** - Três gatos comem três ratos em três minutos. Cem gatos comem cem ratos em quantos minutos?

**R2** – 3 (três) minutos.

**P3** - O pai do padre é filho do meu pai. O que eu sou do Padre?

**R3** – Tio.

**P4** - Se um bezerro pesa 75 kg mais meio bezerro, quanto pesa um bezerro inteiro?

**R4** – 150 (cento e cinquenta) kg.

**P5** – Qual o próximo número da sequência 7, 8, 10, 13, 17?

**R5** – 22.



**P6** – Três senhoras – Dona Branca, Dona Rosa e Dona Violeta – passeavam pelo parque quando Dona Rosa disse:

- Não é curioso que estejamos usando vestidos de cores branca, rosa e violeta, embora nenhuma de nós esteja usando um vestido de cor igual ao seu próprio nome?
- É uma simples coincidência, respondeu a senhora com o vestido violeta.  
Qual a cor do vestido de cada senhora?

		Vestidos		
		Branco	Rosa	Violeta
Dona	Branca			
	Rosa			
	Violeta			

Utilizamos o **F** para indicar  
**FALSO**

Utilizamos o **V** para indicar  
**VERDADEIRO**

**P6** – Três senhoras – Dona Branca, Dona Rosa e Dona Violeta – passeavam pelo parque quando Dona Rosa disse:

- Não é curioso que estejamos usando vestidos de cores branca, rosa e violeta, embora nenhuma de nós esteja usando um vestido de cor igual ao seu próprio nome?
- É uma simples coincidência, respondeu a senhora com o vestido violeta.  
Qual a cor do vestido de cada senhora?

		Vestidos		
		Branco	Rosa	Violeta
Dona	Branca	F	F	V
	Rosa	V	F	F
	Violeta	F	V	F

Utilizamos o **F** para indicar  
**FALSO**

Utilizamos o **V** para indicar  
**VERDADEIRO**

Em Lógica um conceito importante  
é o de "Proposição"

Você sabe o que é uma  
**PROPOSIÇÃO?**

- **Proposição:** é um enunciado verbal, ao qual **deve ser atribuído, sem ambiguidade, um valor lógico verdadeiro (V) ou falso (F).**
  - Exemplos de proposições:
    - Robson Fidalgo é Professor. (V)
    - $3 + 5 = 10$  (F)
    - $5 < 8$  (V)
  - Contra-exemplos de Proposições:
    - Onde você vai ?
    - $3 + 5$
    - Os estudantes jogam vôlei. (Quais ?)

**A realidade é complexa  
e rica em detalhes!**



O que você **abstrai** dessa realidade?

O que é abstração?

## **Abstração** **=**

Operação mental que  
**observa a realidade** e  
captura apenas os  
**aspectos relevantes**  
para um contexto



# Abstração

- Passe alguns segundos olhando para a figura abaixo e diga o que você consegue abstrair dela.



# Abstração

- Conseguiu ver uma jovem mulher?
- Ou um pássaro?

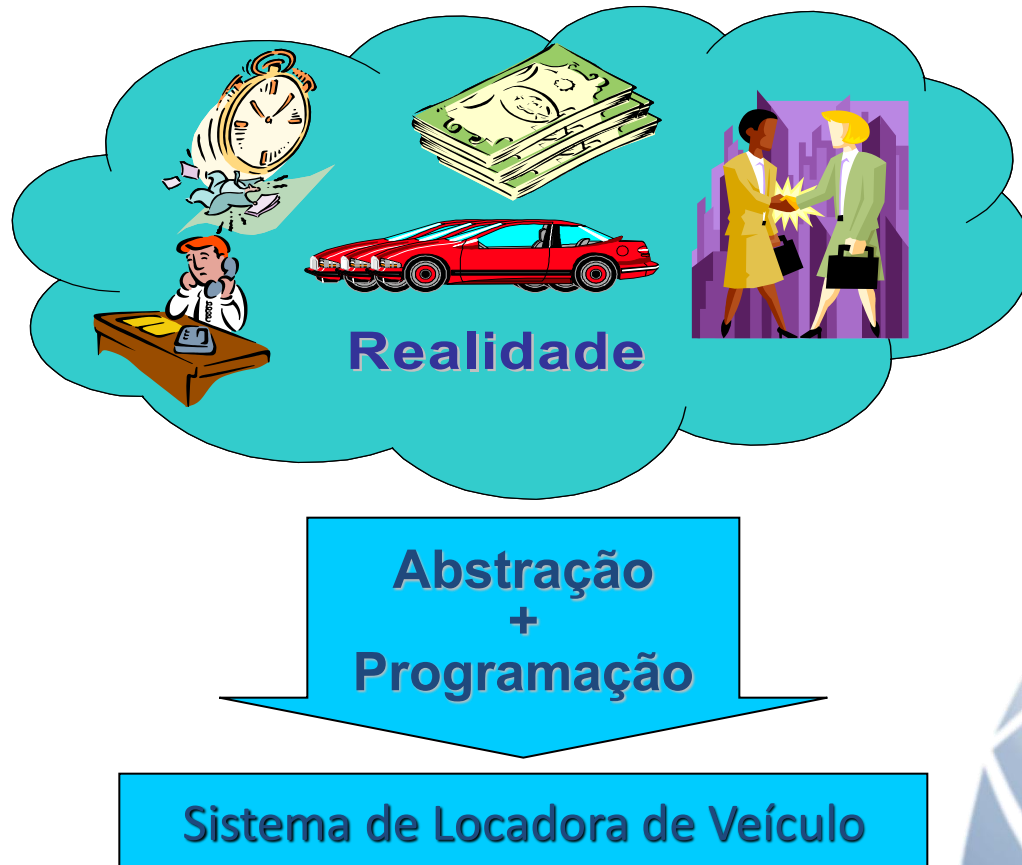


- Perceba que a realidade é a mesma.
  - Isto é, uma figura em preto e branco.
- Mas, dependendo da **observação** da realidade, você pode **ter abstrações diferentes**.
- Por isso, a abstração **depende mais do observador** do que da realidade observada.

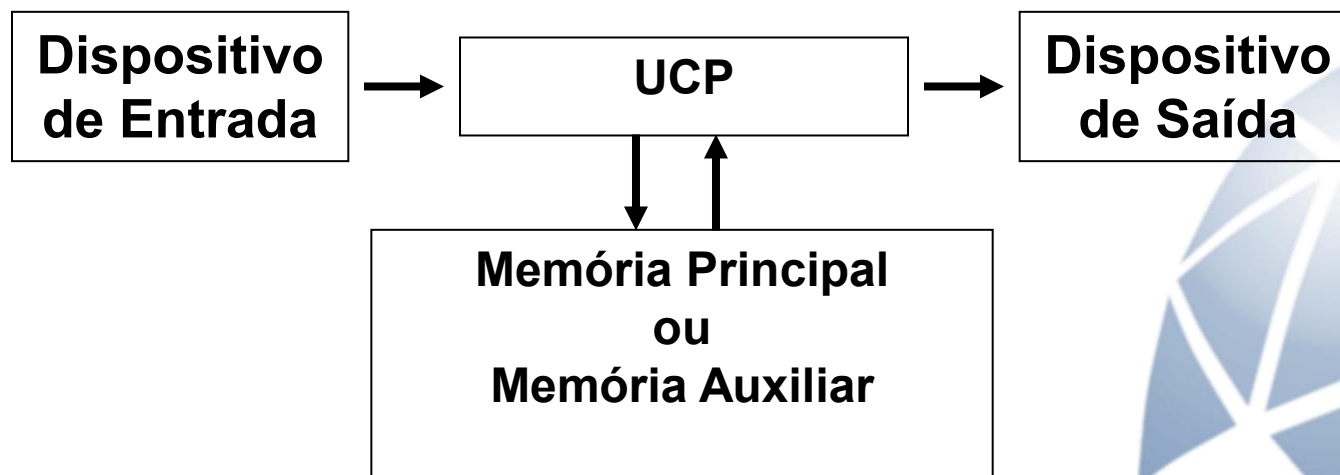


# Abstração

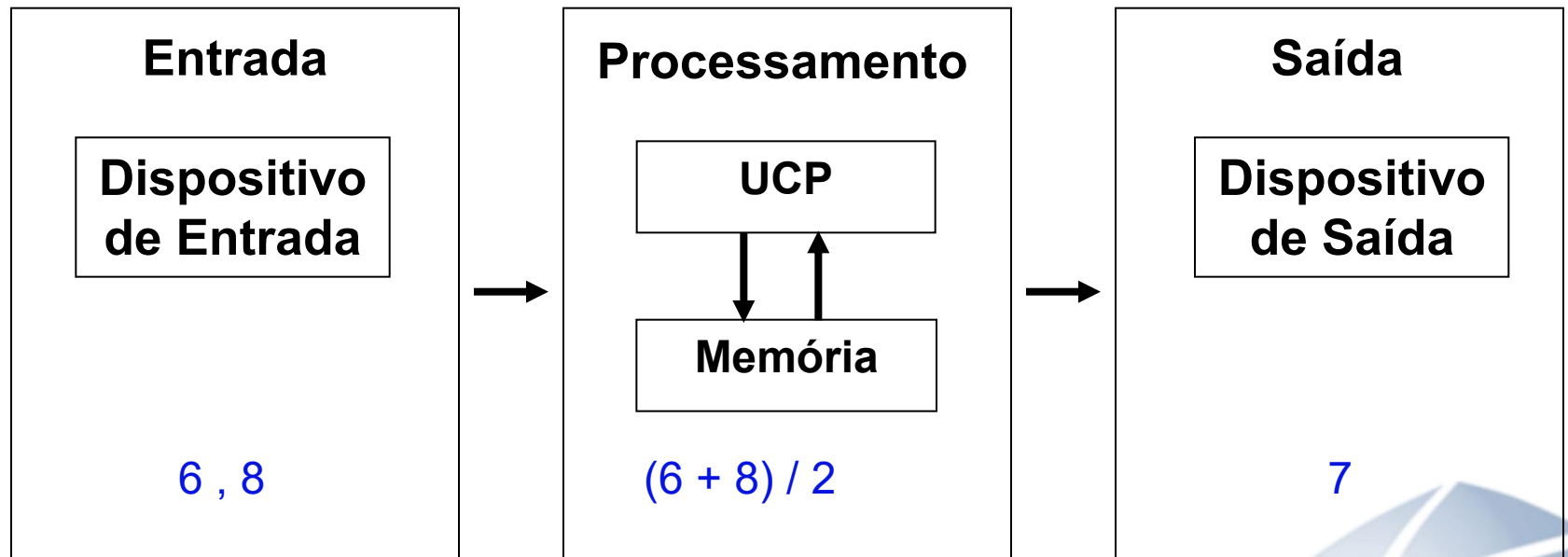
- A tarefa de programar sistemas computacionais envolve o exercício constante da abstração da realidade e sua codificação em uma linguagem de programação.



- O **hardware** de um sistema computacional pode ser agrupado nas **seguintes categorias básicas**:
  - Dispositivo de entrada de dados
  - Dispositivo de saída
  - Unidade Central de Processamento – UCP
  - Memória Principal
  - Memória Auxiliar



- Exemplo 1 – Exibir a média de dois números



O que é uma  
Linguagem de  
Programação?

**Linguagem**  
**=**  
**Meio de Comunicação**



Linguagem de programação

=

conjunto de palavras e regras que  
permitem **comunicar** ao computador  
o que este deve **executar**

Em computação, uma linguagem de programação é a ferramenta de comunicação entre o programador que visa resolver um problema e o computador que irá ajudá-lo a resolver.

## • Tipos de Linguagens de Programação

### – 1 – Linguagem de Máquina

- Possuía uma notação binária (zeros e uns), a qual tornava a programação trabalhosa, cansativa e fortemente sujeita a erros.

### – 2 – Linguagem Assembly

- Surgiu para minimizar as dificuldades da programação em notação binária.
- Códigos de operação e endereços binários foram substituídos por mnemônicos.

### – 3 – Linguagem de Alto Nível

- Aproxima-se das linguagens utilizadas por humanos para expressar problemas.
- Cada declaração numa linguagem de alto nível equivale a várias declarações numa linguagem de baixo nível.

Baixo Nível

## • Tipos de Linguagens de Programação

- **1** - Totalmente codificadas em binário (0's e 1's).
- **2** - Usa instruções simbólicas para representar os 0's e 1's.
- **3** - Voltadas para facilitar o raciocínio humano.

Baixo Nível		Alto Nível
<b>Linguagem de Máquina</b>	<b>Linguagem Assembly (<i>Mnemônica</i>)</b>	<b>Linguagem de Alto Nível</b>
0010 0001 1110	LOAD R1, val1	val2 = val1+val2
0010 0010 1111	LOAD R2, val2	
0001 0001 0010	ADD R1, R2	
0011 0001 1111	STORE R1, val2	
(1)	(2)	(3)

Se o computador só entende linguagem de máquina, o que deve ser feito para que ele entenda programas em linguagem assembly ou de alto nível?

Usar um tradutor!

- Tradutores no contexto de linguagens de programação são programas que recebem como entrada um programa em linguagem assembly ou de alto nível (dita linguagem fonte) e produzem como saída as instruções deste programa traduzidas para linguagem de máquina.
- Existem basicamente três tipos de tradutores:
  - Compilador
  - Interpretador
  - Montadores

## • Tipos de Tradutores

- **1) COMPILADOR:** traduz de uma vez só todo o programa escrito em linguagem de alto nível (**código-fonte**) para um programa equivalente escrito em linguagem de máquina (**código-objeto**).
- **2) INTERPRETADOR:** **traduz** (sem gerar código-objeto) e em seguida executa, uma-a-uma, as instruções de um programa em linguagem de alto nível (**código-fonte**).
- **3) MONTADOR (ou Assembler):** faz a **tradução direta** das instruções Assembly para um programa equivalente escrito em linguagem de máquina.



- **Tipos de Tradutores**

- **COMPILADOR X INTERPRETADOR**

- O **código objeto** (tradução para a linguagem de máquina) que é produzido por um compilador **pode ser executado várias vezes** sem a necessidade de re-compilação. Essa só é necessária se o código-fonte for alterado.

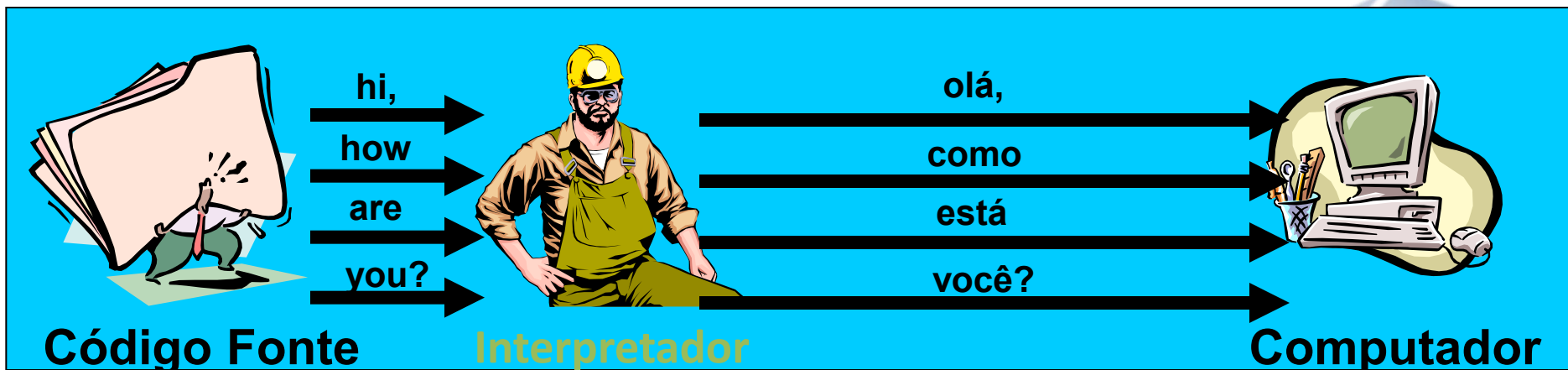


- Todo programa **interpretado** deve **re-executar** o processo de interpretação, independentemente de ter havido ou não modificações no seu código-fonte.



## ■ Tipos de Tradutores

### ■ COMPILADOR X INTERPRETADOR



# Tópicos Iniciais

# Identificadores e Palavras Reservadas

- Identificadores:
  - São **nomes únicos** definidos pelos programadores para identificar/distinguir os elementos de um programa.
- Palavras Reservadas
  - São **instruções primitivas** que têm significados pré-determinados e fazem parte da estrutura de qualquer linguagem de programação.

- Algumas regras para os nomes de Identificadores:
  - 1) Devem **começar** por um **caractere alfabético**.
  - 2) Podem ser seguidos por mais caracteres alfabéticos e/ou numéricos e/ou ponto “.”
  - 3) **Não é permitido** o uso de espaço em branco ou de caracteres especiais, como: @, #, &, \*, +, ?, \$ (exceto o \_).
  - 4) Não poderá ser uma palavra reservada a uma instrução do algoritmo.
  - 5) Devem ser significativos.
  - 6) Não podem ser repetidos dentro de um mesmo programa

# Atividade

- Identifique os erros e reescreva os identificadores abaixo:
  - vm
  - 13salário
  - salário\$
  - salario\_minimo
  - salario+reajuste
  - novoSalario
  - fumante?
  - preço medio
  - %desconto
  - km/h

# Atividade - Respostas

- Identifique os erros e reescreva os identificadores abaixo:
  - vm → **sem significado** → valor\_médio ou valor.medio
  - 13salário → **não começa com caractere alfabético** → salario13
  - salário\$ → **usa caractere especial** → salario ou salário
  - salario\_minimo → **correto**
  - salario+reajuste → **usa caractere especial** → salário\_reajustado
  - novoSalario → **correto**
  - fumante? → **usa caractere especial** → fumante
  - preço medio → **tem espaço em branco** → preço\_medio
  - %desconto → **não começa com caractere alfabético** → percentual\_desconto
  - km/h - **usa caractere especial** → km\_por\_hora

# Tipos de Dados

- As fases de Entrada, Processamento e Saída podem manipular vários tipos primitivos de dados, a saber:

Tipo Primitivo	Descrição
Inteiro	Representa o conjunto de números inteiros
Real	Representa o conjunto de números reais
Caracter	Representa um ou mais caracteres do teclado
Lógico	Representa um valor lógico (V ou F).

- Obs: Um Caractere **SEMPRE** deve estar entre " "
- EX: "A", "Fone 3333-33333", "1",



# Atividade

- Classifique os dados de acordo com o seu tipo, sendo (I = Inteiro, R = Real, C = Caractere e L = Lógico):

a( ) 0      b( ) + 36      c( ) 0,3257      d( ) F

e( ) 1      f( ) "F"      g( ) "+3257"      h( ) -1

i( ) 0,0      j( ) - 0,001      k( ) "-0,0"      l( ) ".F."

m( ) "o"      n( ) + 0,05      o( ) ".V."      p( ) 7/2

q( ) 32      r( ) + 3257      s( ) V      t( ) -32

u( ) "A"      v( ) "abc"      x( ) -1,9E123      z( ) "0"

# Atividade - Respostas

- Classifique os dados de acordo com o seu tipo, sendo  
(I = Inteiro, R = Real, C = Caractere e L = Lógico):

a( I ) 0      b( I ) + 36      c(R) 0,3257      d(L) F

e( I ) 1      f(C) "F"      g(C) "+3257"      h( I ) -1

i (R) 0,0      j(R) - 0,001      k(C) "-0,0"      l (C) ".F."

m(C) "o"      n(R) + 0,05      o(C) ".V."      p(R) 7/2

q( I ) 32      r( I ) + 3257      s(L) V      t ( I ) -32

u(C) "A"      v(C) "abc"      x(R) -1,9E123      z(C) "0"

# Constante e Variável

- **Constante** é um identificador que representa **valores constantes**, ou seja, que **não variam no decorrer do programa**.
  - Seu uso **poupa tempo** quando tem que alterar o seu valor no programa.
- Ao trocar o valor de uma constante, todas as instruções que a usam irão manipular, automaticamente, o novo valor.

# Variável e Constante

- **Variável** é um endereço físico da memória principal, que é representado por um identificador que, ao longo do seu tempo de existência, pode armazenar vários conteúdos de um único tipo pré-determinado.

Endereço Físico	Identificador	Conteúdo	Tipo
1000:2000	Nome	“João”	Caracter
2001:3000	RG	12345	Inteiro
3001:4000	Salário	999,99	Real
4001:5000	Fumante	F	Lógico

# Variável e Constante

- Simplificando...

- Considere que a memória principal do seu computador é um armário, onde cada gaveta é a uma variável.



# RESOLVER A

## LISTA 01

# Expressões

- Uma **expressão** é uma **fórmula** para processamento de **um valor**.
- As principais expressões são as seguintes:
  - **Aritméticas**: Retornam um valor numérico (inteiro ou real).
    - **EX**:  $10+(3+1)/2$
  - **Lógicas**: Retornam um valor lógico V ou F.
    - **EX**:  $(3=2+1)$  e  $(3>2)$
- Obs: Ter atenção com as prioridades dos operadores !
  - $3+2*2 = 7$
  - $(3+2)*2 = 10$

Use corretamente os parênteses!

# Operadores

- Os **operadores** podem ser classificados em:
  - **Binários**: atuam sobre dois operandos.
    - Ex: operadores aritméticos básicos (+ - \* /)
  - **Unários**: atuam sobre um único operando.
    - Ex.: o sinal de (-) na frente de um número para inverter seu sinal.
- Tipos de operadores da nossa linguagem:
  - Atribuição
  - Aritméticos
  - Relacionais
  - Lógicos ou Booleanos





# Operadores

- **Atribuição**: serve para atribuir um valor a uma variável.
- Operador de atribuição “  $\leftarrow$  ”, "<-" ou "="
  - EX:
    - Nome <- “Um nome”;
    - Idade <- 18;
    - Casado <- F;
    - Salário <- 500.50;
  - A expressão do lado **direito** do operador **é avaliada** e seu resultado é **armazenado** na variável **à esquerda**.
    - Obs: A expressão deve retornar o mesmo tipo da variável, ENTRETANTO, A **ATRIBUIÇÃO PODE SER FEITA NO SENTIDO CONTRÁRIO (50 -> S)**

# Operadores

- **Aritméticos:** são as operações aritméticas básicas

Operador	Tipo	Operação	Prioridade
-	Unário	Inversão do Sinal	1
+	Unário	Manutenção do Sinal	1
RAD(x)	Binário	Radiciação	2
POT(x,y)	Binário	Potenciação	2
DIV	Binário	Quociente da Divisão Inteira	3
MOD	Binário	Resto da Divisão Inteira	3
/	Binário	Divisão	3
*	Binário	Multiplicação	3
-	Binário	Subtração	4
+	Binário	Adição	4

Diagrama de seta indicando a ordem de prioridade:

↑  
Ordem prioridade  
↓

Maior  
Menor

# Operadores

- Exemplos:

Operador	Operação	Exemplo	Resultado
<b>+</b>	<b>Adição</b>	<b>4+3</b>	<b>7</b>
<b>-</b>	<b>Subtração</b>	<b>4-3</b>	<b>1</b>
<b>*</b>	<b>Multiplicação</b>	<b>4*3</b>	<b>12</b>
<b>/</b>	<b>Divisão</b>	<b>4/3</b>	<b>1,33</b>
<b>MOD</b>	<b>Resto da Divisão Inteira</b>	<b>4 %% 3</b>	<b>1</b>
<b>DIV</b>	<b>Quociente da Divisão Inteira</b>	<b>4 %/% 3</b>	<b>1</b>
<b>POT(x,y)</b>	<b>Potenciação</b>	<b>4^3</b>	<b>64</b>
<b>RAD(x)</b>	<b>Radiciação</b>	<b>sqrt(4)</b>	<b>2</b>
<b>+</b>	<b>Manutenção do Sinal</b>	<b>+- 4</b>	<b>- 4</b>
<b>-</b>	<b>Inversão do Sinal</b>	<b>-- 4</b>	<b>+4</b>

# Operadores

- **Relacionais:** são operadores binários (de mesma prioridade) que somente **retornam** os valores lógicos **V** ou **F**.

Operador	Comparação
>	maior que
<	menor que
>=	maior ou igual
<=	menor ou igual
=	igual
<>	diferente

- Estes somente são usados para **efetuar comparações**, as quais só podem ser feitas entre dados do **mesmo tipo**.
- O resultado de uma comparação é sempre um **valor lógico**.

# Operadores

- Exemplos:

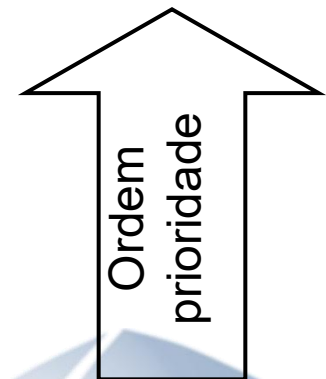
Operador	Comparação	Exemplo	Resultado
>	maior que	4>3	V
<	menor que	4<3	F
>=	maior ou igual	4>=3	V
<=	menor ou igual	4<=3	F
=	igual	4==3	F
<>	diferente	4!=3	V

# Operadores

- **Lógicos** ou **Booleanos**: são usados para combinar expressões relacionais e lógicas. Também retornam como resultado valores lógicos V ou F.

Operador	Tipo	Operação	Prioridade
NÃO	Unário	Negação	1
E	Binário	Conjunção	2
OU	Binário	Disjunção	3

Maior



Menor

# Operadores

- Exemplos:

Operador	Operação	Exemplo	Resultado
OU	Disjunção	V ou F	V
E	Conjunção	V e F	F
NÃO	Negação	Não V	F

- **Operações Lógicas:** são usadas para formar novas proposições a partir de proposições existentes.

Operação	Símbolo	Significado
Negação	!	Não
Conjunção	&	E
Disjunção		OU



- Exemplos de aplicação das operações lógicas

- Resumindo:

p	q	$\sim p$	$p \wedge q$	$p \vee q$
V	V	F	V	V
V	F	F	F	V
F	V	V	F	V
F	F	V	F	F

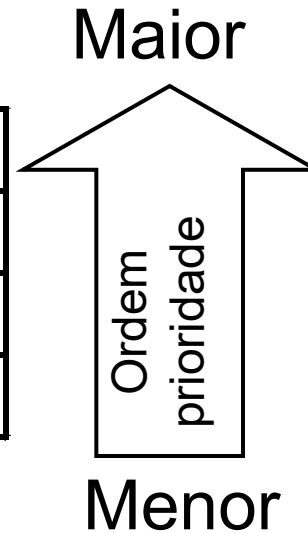
- Ou seja:

- Não ( $\sim$ ) troca o valor lógico. Se é F passa a ser V e vice-versa.
- E ( $\wedge$ ) só tem valor V quando as duas proposições forem V. Basta uma proposição ser F para o resultado ser F.
- OU ( $\vee$ ) só tem valor F quando as duas proposições forem F. Basta uma proposição ser V para o resultado ser V.

# Operadores

- Ordem de prioridades

Operador	Prioridade
Aritméticos	1
Relacional	2
Lógico	3



- Observações:
  - Operadores de igual prioridade, execução da esquerda para direita.
  - Para alterar a ordem de prioridade, utilizar parênteses.

# RESOLVER A

## LISTA 02