CAPÍTULO I

INTRODUÇÃO

I. 1 – CONCEITOS PRELIMINARES

O leitor pode estar se perguntando: Porque estudar lógica? Ou O que é lógica?

A lógica faz parte do nosso cotidiano, fazemos uso da mesma em diversos momentos e já estamos tão habituados que nem percebemos. A simples tarefa de abrir uma porta requer um conjunto de decisões e ações lógicas, que executamos de maneira quase automática.

A palavra lógica vem do Grego, logike, e significava ciência do raciocínio. Atualmente a lógica está associada a idéia de coerência e ao emprego da razão.

O que se pretende com esse capítulo é apresentar os conceitos básicos da lógica formal, necessária ao aprendizado de qualquer linguagem de programação. O sentido da lógica, sua etimologia e demais questões filosóficas fogem completamente ao escopo dessa apostila e mais ainda, fogem inteiramente ao domínio desse que vos fala.

Embora não haja uma única definição para a lógica, termino dizendo que lógica é o encadeamento coerente do pensamento segundo certas regras e convenções pré-estabelecidas.

> **Lógica** é o encadeamento coerente do pensamento segundo certas regras e convenções pré-estabelecidas.

I.2 – LÓGICA DE PROGRAMAÇÃO

Se é importante que tenhamos uma compreensão geral de lógica, não é menos verdade que precisamos entender de maneira clara e precisa a idéia da lógica de programação.

O significado da lógica de programação não é muito diferente daquele apresentado no bloco anterior, porém agora daremos maior ênfase a três coisas: regras, seqüência e objetivo.

A seguir é dado um exemplo que ilustra como os três conceitos citados acima se unem para criar um tipo de lógica que falaremos até o final dessa apostila e por fim daremos uma definição a Lógica de Programação.

No primeiro bloco falamos de abrir uma porta, como então descrever passo-a-passo todas as etapas envolvidas nesse processo?

Antes de responder duas definições são necessárias:

Essa é uma porta normal e contém uma única fechadura;

A pessoa que irá abri-la não vai explodi-la com uma bazuca ou algo similar.

Feito isso, agora podemos dar início, descrevendo de forma sucinta os passos a serem dados objetivando abrir a porta:

Abrindo a Porta	
Passo 1:	Ir até a porta;
Passo 2:	Girar a porta e verificar se a porta está fechada. Se a resposta for SIM, passo 3, se NÃO, passo 6.
Passo 3:	Buscar a chave e coloca-la na fechadura;
Passo 4:	Girar a chave no sentido horário até que encontre Resistência ao movimento;
Passo 5:	Girar a maçaneta;
Passo 6:	Abrir a porta;

O que vemos acima são as rotinas básicas para se abrir uma porta, mas não é a única e nem tão pouco a mais detalhada. Porém ilustra claramente os três elementos citados no início dessa sessão - regras, seqüência e objetivo. Algumas regras foram estabelecidas logo no início, a fim de dizer o que era ou não permitido, depois executamos passos seqüencialmente e tudo isso com o objetivo de abrir a porta.

Com base nesse exemplo podemos dizer que a Lógica de Programação é a técnica de encadear rotinas para atingir um objetivo.

> **Lógica de Programação** é a técnica de encadear rotinas para atingir um objetivo.

I.3 – LÓGICA MATEMÁTICA

Mais uma lógica?! Só pode tá brincando...

Calma. Não podemos falar de Lógica e omitir sua presença na Matemática, até por que a informática, como parte da Matemática Aplicada, faz uso de muitos conceitos matemáticos de lógica, na elaboração de linguagens de programação, construção de dispositivos, periféricos etc.

Nessa sessão, vamos examinar alguns tópicos bastante aplicados na programação de computadores e essenciais ao entendimento dos assuntos desta apostila.

I.3.1 – PROPOSIÇÕES

Leia atentamente as frases seguintes:

- Diogo é maconheiro.
- Hoje não choveu.
- O **SPFC** é tricampeão do Mundo.
- O Sol é uma estrela.
- 2 > 1+3.
- Todo ser vivo é carnívoro.

O que elas tem em comum? As três frases acima, são afirmativas, declaram alguma coisa, ou seja, são expressões que encerram um pensamento de sentido completo e podem ser classificadas como verdadeiras ou falsas e isso, na matemática, chama-se proposição.

Proposição é toda expressão que encerra um pensamento de sentido completo e pode ser classificada como **V** (verdadeira) ou **F** (falsa).

Além de poderem ser classificadas como verdadeiras ou falsas as proposições obedecem a dois princípios:

I. Princípio da não-contradição:

Uma **proposição** não pode ser verdadeira e falsa ao mesmo tempo.

II. Princípio do terceiro excluído:

Toda **proposição** ou é verdadeira ou é falsa; não existe um terceiro valor lógico.

Em geral indicamos uma proposição por letras minúsculas: p,q,r,s,t,...
Os símbolos **V** ou **F** são chamados de valores lógicos.

I.3.2 – OPERADORES LÓGICOS

A partir de proposições simples podemos formar novas proposições com o emprego de símbolos lógicos chamados de conectivos. Trabalharemos com três tipos básicos de conectivos listados abaixo:

Tipo	Função
NÃO	NEGAÇÃO
E	CONJUNÇÃO
OU	DISJUNÇÃO
XOU	DISJUNÇÃO

Embora o **NÃO** (negação) figure na tabela anterior, ele não é um conectivo, trata-se de um operador que inverte o valor lógico de uma proposição.

I.3.2.1 – NÃO (Negação)

Esse operador como já mencionamos acima, inverte o valor lógico de uma proposição. Portanto, se uma dada proposição for verdadeira, então sua negação será falsa. A tabela a seguir, tabela-verdade, mostra o funcionamento do **NÃO**.

p	**NÃO** p
V	F
F	V

O que vemos acima é uma proposição p assumindo os valores V e F, e sua negação assumindo valores F e V respectivamente.

Temos abaixo três exemplos de proposições e suas respectivas negações:

EXEMPLOS:

p: 4 é maior que 5 (**F**)

NÃO p: 4 não é maior que 5 (**V**)

q: Gabriel é um vagabundo (**V**)

NÃO q: Gabriel não é um vagabundo (**F**)

A proposição **não p** tem sempre o valor oposto ao de **p**, isto é, é verdadeira quando **p** é falsa e é falsa quando **p** é verdadeira.

I.3.2.2 – E (Conjunção)

O conectivo **E** é um operador que une duas proposições dando origem a uma terceira, ou seja, a partir de duas proposições dadas podemos construir uma nova proposição e para isso basta intercalar entre as mesmas esse conectivo.

Veja abaixo:

a: A lua é Satélite da Terra **E** o Sol é um planeta.

b: $5\ menor\ que\ 3$ **E** $2^3\ igual\ a\ 8$.

c: Um quadrado de lado x tem diagonal $2x$ **E** área medindo x^2.

Esse conectivo é a forma pela qual se declara ao mesmo tempo duas proposições simples. A nova proposição obtida terá os seguintes valores lógicos:

p	q	p **E** q
V	V	V
V	F	F
F	V	F
F	F	F

O que essa tabela-verdade nos mostra é que em apenas um dos casos a nova proposição, p E q, será verdadeira e isso ocorre quando p e q são verdadeiras. Esse fato nos permite estabelecer um critério geral para as conjunções.

Uma **Conjunção** é **V** (verdadeira) se, e somente se, todas as proposições simples que a geram são simultaneamente verdadeiras e **F** (falsa) nos demais casos.

EXEMPLOS:

p : 6 é maior que 5 (**V**)

q : Um triângulo tem três lados (**V**)

r *:* Todo quadrilátero é um retângulo (**F**)

p **E** q : (**V**)

p **E** q **E** r : (**F**)

p **E** r : (**F**)

p **E** q **E** (**NÃO** r) : (**V**)

p **E** (**NÃO** r) : (**V**)

NÃO (p **E** q **E** r) : (**V**)

I.3.2.3 – OU (Disjunção)

O conectivo **OU** é um operador que une duas proposições dando origem a uma terceira, ou seja, a partir de duas proposições dadas podemos construir uma nova proposição e para isso basta intercalar entre as mesmas esse conectivo.

Veja abaixo:

a: Todo político é honesto **OU** 5 é raiz da equação $x^3 = 125$.

b: $1\ menor\ que\ 3$ **OU** $7^2\ igual\ a\ 51$.

c: 12 é divisível por 5 **OU** O Brasil é penta campeão de Futebol.

Esse conectivo é a forma pela qual se declara pelo menos uma das duas proposições simples. A nova proposição obtida terá os seguintes valores lógicos:

p	q	p **OU** q
V	V	V
V	F	V
F	V	V
F	F	F

O que essa tabela-verdade nos mostra é que em apenas um dos casos a nova proposição, p OU q, será falsa e isso ocorre quando p e q são falsas. Esse fato nos permite estabelecer um critério geral para as conjunções.

Uma **Disjunção** é **F** (falsa) se, e somente se, todas as proposições simples que a geram são simultaneamente falsas e **V** (verdadeira) nos demais casos.

EXEMPLOS:

p : 6 é maior que 5 (**V**)

q : Um triângulo tem três lados (**V**)

r *:* Todo quadrilátero é um retângulo (**F**)

p **OU** q : (**V**)

p **OU** q **OU** r : (**V**)

p **OU** r : (**V**)

(p **OU** q) **E** r : (**F**)

(**NÃO** p) **OU** r : (**F**)

NÃO (p **OU** q **OU** r) : (**F**)

I.3.2.4 – XOU (Disjunção Exclusiva)

O conectivo **XOU** é um operador que une duas proposições dando origem a uma terceira, ou seja, a partir de duas proposições dadas podemos construir uma nova proposição e para isso basta intercalar entre as mesmas esse conectivo.

Veja abaixo:

a: $8!\ igual\ a\ 40320$ **XOU** o Sol é uma estrela de primeira grandeza.

b: $2 \times 9\ igual\ a$ 17 **XOU** $2^3\ igual\ a$ 8.

c: Um triângulo tem uma diagonal **XOU** Todo triângulo é escaleno.

Esse conectivo é a forma pela qual se declara ao mesmo tempo duas proposições simples. A nova proposição obtida terá os seguintes valores lógicos:

p	q	p **XOU** q
V	V	F
V	F	V
F	V	V
F	F	F

O que essa tabela-verdade nos mostra é que em apenas dois dos casos a nova proposição, XOU, será verdadeira e isso ocorre quando apenas p é verdadeira ou apenas q é verdadeira. Esse fato nos permite estabelecer um critério geral para as disjunções exclusivas.

Uma **Disjunção Exclusiva** é **V** (verdadeira) se, e somente se, uma, e somente uma, das proposições simples que a gera é verdadeira e as demais falsas e **F** (falsa) nos demais casos.

EXEMPLOS:

p : 6 é maior que 5 **(V)**

q : Um triângulo tem três lados **(V)**

r *:* Todo quadrilátero é um retângulo **(F)**

p **XOU** q : **(F)**

(p **XOU** q) **XOU** r : **(F)**

p **XOU** r : **(V)**

(p **OU** q) **XOU** r : **(V)**

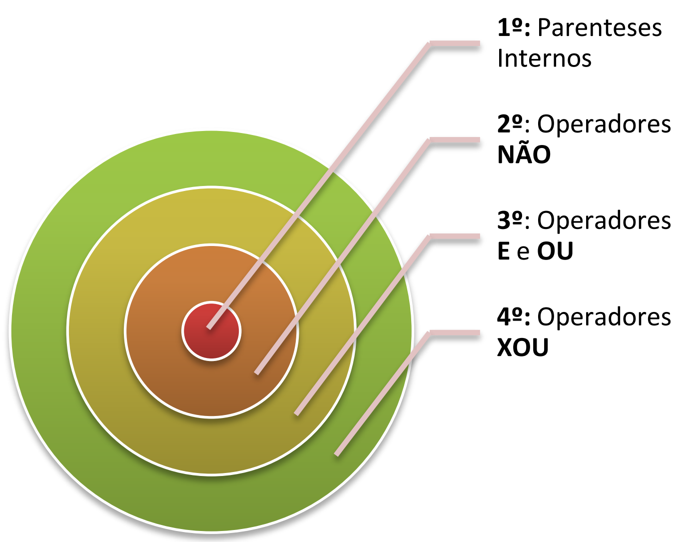
(**NÃO** p) **XOU** r : **(F)**

NÃO (p **XOU** q) : **(V)**

I.3.2.5 – HIERARQUIA LÓGICA

A palavra hierarquia é aplicada neste caso e nos seguintes no sentido de ordem, a fim de definir claramente, durante a resolução de uma expressão, quem deve ser primeiro solucionado. O conhecimento da ordem hierárquica é fundamental para construção de uma boa lógica, desprezá-la poderá se traduzir em expressões mal elaboradas e que não realizam o esperado.

O diagrama seguinte ilustra a precedência existente entre as estruturas que compõe uma expressão, devendo ser lido de dentro para fora e tal qual ele uma expressão também deve ser resolvida de dentro para fora, observando-se primeiro os parênteses mais internos, depois os operadores **NÃO**, logo em seguida os operadores, **E** e **OU**, e por último os operadores **XOU**.



EXEMPLOS:

Tomemos a expressão seguinte e sua resolução:

NÃO (F) **XOU** **NÃO** (V **OU** F) **E** (F)	**HIERARQUIA**
NÃO (F) **XOU** **NÃO** (V **OU** F) **E** (F)	Parênteses Internos
NÃO (F) **XOU** **NÃO** (V) **E** (F)	Operadores **NÃO**
V **XOU** F **E** (F)	Operadores **E OU**
V **XOU** F	Operadores **XOU**
V	

EXERCÍCIOS RESOLVIDOS

R1 – Assinale com P as sentenças que forem proposições e com X as demais:

a) Joana usa roupas caras. (**P**)
b) 3 é raiz da equação $x + 8 = 11$. (**P**)
c) O sol é um Planeta. (**P**)
d) Qual a capital do Brasil? (**X**)
e) Alguém mora nessa Casa. (**X**)

R2 – O que faz o operador lógico **NÃO**? Monte sua tabela verdade.

RESOLUÇÃO:

O operador lógico **NÃO** inverte o valor lógico de uma proposição. Sua tabela verdade para uma proposição p pode ser vista abaixo:

p	**NÃO** p
V	F
F	V

R3 – Quais os tipos de disjunções estudadas e o que as diferencia ?

RESOLUÇÃO:

As disjunções estudadas foram **OU** e **XOU**, sendo a última chamada de disjunção exclusiva. Ambas são conectivos, ou seja, unem duas proposições quaisquer formando uma nova. Enquanto a primeira retorna o valor lógico **V** (VERDADE), quando ao menos uma das proposições unidas é verdadeira, a segunda só retorna o valor **V**, se, e somente se, uma única proposição é verdadeira.

R4 – Classifique com **V** ou **F** cada uma das proposições seguintes:

a) Todo mamífero é carnívoro. (**F**)
b) A Terra é um planeta. (**V**)
c) $8 > 5$ **XOU** $\sqrt{9} = 3$. (**F**)
d) **NÃO**$(4^3 = 64)$ **XOU** $(\sin 30° = 0{,}5)$ (**V**)
e) $\ln e^5 = 5$ **E** ($\int_0^1 x^2 = 1/3$ OU F) (**V**)
f) V **XOU** V **E** F (**V**)
g) **NÃO** p **E** p (**F**)

EXERCÍCIOS PROPOSTOS

P1 – O que você entende por Lógica de Programação?

P2 – O que é proposição e cite seus dois princípios?

P3 – O que faz o operador **XOU**? Monte sua tabela verdade.

p	q	p **XOU** q

P4 – Dê exemplos de duas proposições e duas não proposições.

P5 – Você está numa parada de ônibus, aguardando para embarcar. Descreva em até dez passos como fazer para tomar o ônibus desejado.

Tomando Ônibus
Passo 1:
Passo 2:
Passo 3:
Passo 4:
Passo 5:
Passo 6:
Passo 7:
Passo 8:
Passo 9:
Passo 10:

P6 – Assinale com a letra correta observando a ordem hierárquica.

A – 1º	( ) Operadores **E** e **OU**.
B – 2º	( ) Parênteses Internos.
C – 3º	( ) Operadores **XOU**.
D – 4º	( ) Operadores **NÃO**.

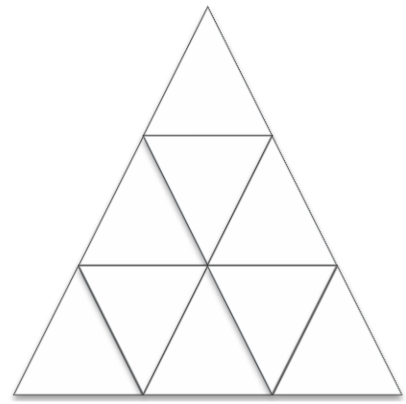
P7 – Classifique com **V** ou **F** cada uma das proposições:

a) $17^{15} < 32^{12}$. ( )

b) $\sqrt[2]{121} = \pm 11$. ( )

c) $\lim_{x \to 1} \frac{x^2 - 1}{x - 1} = 2$ ( )

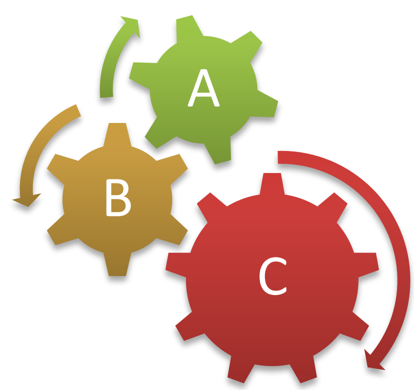
P8 – Quantos triângulos têm a figura abaixo?



P9 – Os números inteiros são dispostos em Matrizes conforme mostrado abaixo. Em qual matriz, linha e coluna aparecerá o número 842?

Matriz 1	Matriz 2	Matriz 3
$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$	$\begin{pmatrix} 10 & 11 & 12 \\ 13 & 14 & 15 \\ 16 & 17 & 18 \end{pmatrix}$	$\begin{pmatrix} 19 & 20 & 21 \\ 22 & 23 & 24 \\ 25 & 26 & 27 \end{pmatrix}$

P10 – Três engrenagens diferentes possuem números de dentes respectivamente: A=14;B=18 e C=27. Quantas voltas completas deve dar engrenagem A para que a engrenagem C dê também um número completo de voltas ? Justifique.



I.4 - EXPRESSÕES MATEMÁTICAS

Ao longo dessa apostila iremos trabalhar predominantemente com a lógica, mas será freqüente o uso de expressões matemáticas e para desenvolvê-las será preciso que tenhamos um critério de solução e uma forma de representação, isto é, deveremos saber em qual ordem resolve-las a fim de simplificarmos ao máximo a expressão original obtendo uma outra equivalente, porém mais simples e fácil de ser compreendida, e também devemos saber como escrever uma determinada expressão.

Dizer que duas expressões são equivalentes, significa que ambas produzem os mesmos valores lógicos e numéricos, mas não quer dizer que sejam iguais, se o fossem, então nada teria sido feito.

Nas subseções seguintes vamos apresentar os operadores e funções matemáticas mais comuns e suas representações dentro da linguagem de programação pascal - outras linguagens podem ter representações diferentes das apresentadas aqui, mas isso não altera a importância dessa seção, pois independente dos símbolos usados numa linguagem específica, permanece o critério de ordem que veremos adiante.

Duas expressões são ditas **EQUIVALENTES** quando produzem os mesmos valores lógicos e numéricos**.**

I.4.1 – OPERADORES ARITMÉTICOS

São todos os símbolos básicos presentes nas expressões matemáticas e representativos de uma operação específica como adição, subtração, multiplicação, divisão etc. Uma observação importante é que nem todos os símbolos são escritos tal como sua notação na Matemática.

Acompanhe pela tabela abaixo a operação e o respectivo operador (símbolo), mas não se esqueça, alguns operadores não obedecem a notação matemática usual.

Operação	Símbolo	Operando	Saída
SOMA	+	Z ou R	Z ou R
SUBTRAÇÃO	-	Z ou R	Z ou R
MULTILIPAÇÃO	*	Z ou R	Z ou R
DIVISÃO	/	Z ou R	Z

As letras Z e R representam respectivamente o conjunto dos números Inteiros e dos números Reais.

A tabela anterior nos mostra que cada operação possui um operador próprio, está disponível para determinados números (operando) e retorna um tipo de número de acordo com o operando. Nos exemplos a seguir mostramos uma expressão matemática com a notação usual e ao lado a mesma expressão escrita usando-se os operadores aritméticos.

EXEMPLOS:

Expressão Matemática	Pascal	Saída
$\frac{3}{5+1}$	$3/(5+1)$	$0{,}5$
$\frac{3{,}1416}{2}$	$3{,}1416/2$	$1{,}5708$
$\frac{9}{3-1}$	$9/(3-1)$	$4{,}5$
3×3	$3 * 3$	9

Uma última observação diz respeito a escrita de uma expressão e pode ser visto na tabela acima. Todas as expressões numa linguagem de programação devem ser escritas linearmente com o auxílio de parênteses para indicar a prioridade de uma operação em relação a outra. Para ficar claro, veja como ficaria a primeira e a terceira do exemplo anterior sem o uso do parêntese:

$\frac{3}{5+1}$	$3/5+1$	$2{,}6$
$\frac{9}{3-1}$	$9/3-1$	$2{,}0$

Na primeira linha, o valor $2{,}6$ não corresponde a saída esperada para expressão original, sem o parêntese o processador primeiro realiza a divisão de três por cinco para depois somar o resultado com um. O que foi dito para a primeira linha aplica-se a terceira, nove é dividido por três e em seguida subtrai-se um, resultando em $2,0$, que obviamente não é a saída esperada.

Agora, não é necessária uma excessiva preocupação com esse assunto, mais adiante, quando tivermos mais elementos, construiremos expressões mais complexas e dedicaremos um tópico inteiro a hierarquia entre operadores e funções.

I.4.2 – FUNÇÕES MATEMÁTICAS

Assim como os operadores aritméticos, as funções matemáticas são parte importante e indispensáveis a qualquer linguagem de programação.

O objetivo agora é apresentar algumas das principais funções matemáticas, presentes na linguagem pascal.

Para efeito didático dividimos as funções matemáticas em Trigonométricas e Numéricas.

I.4.2.1 – FUNÇÕES TRIGONOMÉTRICAS

As funções trigonométricas são o **seno**, **co-seno** e **arctg**, a partir dessas três podemos obter qualquer outra com o auxílio dos operadores aritméticos.

Seguindo o que já fora feito com os operadores aritméticos, abaixo apresentamos uma tabela resumo:

Função	Pascal	Operando	Saída
SENO	$SIN(\theta)$	Z ou R	R
CO-SENO	$COS(\theta)$	Z ou R	R
ARCTG	$ARCTAN(x)$	Z ou R	R

As funções trigonométricas trabalham sempre com ângulos em radianos e sempre retornam um número real.

A função seno devolve o seno de um ângulo, neste caso ângulo θ. O mesmo se aplica a função co-seno que obviamente devolve o co-seno.

A função arctg, assim com as duas anteriores, nos devolve um número real, correspondente ao ângulo em radianos cuja tangente tem valor x.

EXEMPLOS:

$\sin(0{,}524) \cong 0{,}50$ $\quad$ $\arctan(1) \cong 0{,}785$

$\cos(0{,}785) \cong 0{,}71$

Você já sabe como faria para obter a tangente de um ângulo qualquer, uma vez que não existe a função correspondente? Pense nisso.

As funções **Trigonométricas** incidem sobre ângulos em radianos e retornam números reais.

I.4.2.2 – FUNÇÕES NUMÉRICAS

Aqui encontraremos funções muito importantes e utilizadas com muita freqüência na resolução de inúmeros problemas.

Função	Pascal	Operando	Saída
$\lvert x\rvert$	$abs(x)$	Z ou R	Z
$\ln(x)$	$\ln(x)$	Z ou R	R
e^x	$exp(x)$	Z ou R	R
$\sqrt{x}$	$sqrt(x)$	Z ou R	R
x^2	$sqr(x)$	Z ou R	Z ou R

Na tabela acima, x representa um número qualquer, inteiro ou real, sobre qual a função opera retornando um outro número conforme a função e o valor de x.

Não temos em pascal uma função que retorne a potência de um número elevado a um expoente qualquer, mas há uma forma de se obter tal resultado aplicando-se adequadamente as funções $\exp(x)$ e $\ln(x)$, conforme mostrado na última linha do exemplo abaixo:

EXEMPLOS:

Expressão Matemática	Pascal	Saída
$\lvert -5\rvert$	$abs(-5)$	5
$\ln(2{,}7183)$	$\ln(2{,}7183)$	$\cong 1{,}00$
e^1	$\exp(1)$	$\cong 2{,}7183$
$\sqrt{3}$	$sqrt(3)$	$\cong 1{,}732$
9^2	$sqr(9)$	81
2^3	$\exp(3 * \ln(2))$	$8{,}00$

Você saberia como obter a raiz enésima $\sqrt[n]{x}$ de um número? Pense nisso.

I.3.2.5 – HIERARQUIA MATEMÁTICA

EXERCÍCIOS RESOLVIDOS

EXERCÍCIOS PROPOSTOS

I.4.3 – OPERADORES RELACIONAIS

Sua utilização dentro da programação é freqüente e servem para realizar comparações entre duas expressões ou sentenças.

Sempre que utilizarmos tais operadores estaremos interessados no valor lógico retornado pela comparação em questão.

Veja na tabela seguinte a operação e o símbolo em pascal correspondente:

Operação	Pascal
Igual a	=
Menor que	<
Maior que	>
Menor ou igual a	<=
Maior ou igual a	>=
Diferente de	<>

Em seções anteriores, quando falávamos de proposições e operadores lógicos, usamos nos exemplos alguns dos operadores relacionais listados.

O nosso intuito agora é apresentá-los formalmente e falar da sua utilização dentro da lógica de programação.

Acompanhe os exemplos e as dúvidas que permanecerem serão sanadas quando estudarmos algoritmos.

EXEMPLOS:

Imagine que você trabalhe numa operadora de telefonia móvel e sua função é desenvolver um sistema que emita uma mensagem de advertência sempre que o saldo do cliente atingir o valor mínimo de cinco reais.

Podemos encarar o problema da seguinte forma, se o saldo for menor ou igual cinco reais, então enviar mensagem padrão, como a seguinte: “Caro cliente. Seu saldo é inferior a cinco reais, por gentileza recarregue seu aparelho”. Veja abaixo:

Mensagem de Advertência	
Passo 1:	Verificar saldo;
Passo 2:	Comparar o saldo com o valor de R$ 5,00 s: saldo **<=** 5?
Passo 3:	Se saldo inferior a cinco reais, emita mensagem padrão;
Passo 4:	Se não, fazer nada;

No **passo 2**, já de posse do saldo do cliente, comparamos este com cinco reais (s: saldo **<=** 5?) e caso o valor lógico retornado por essa proposição seja **V**, emite-se a mensagem padrão, **passo 3**, se não, nada é feito, **passo 4**.

Outra coisa a se destacar é que sempre comparamos grandezas de mesma natureza, ou seja, não faz sentido comparar o saldo, grandeza numérica, com a o nome do cliente, texto.

Os operadores relacionais entram na programação em pontos de tomada de decisão, onde é necessário optar por um caminho, mas o caminho que será tomado depende de uma situação peculiar, no caso da mensagem de advertência, emiti-la ou não dependia do valor do saldo está abaixo ou acima de cinco reais.

IMPORTANTE

1. Nas comparações, estamos interessados no valor lógico (**V** ou **F**) que a mesma retorna, pois com base nessa informação podemos tomar uma decisão;

2. Sempre comparamos grandezas de mesma natureza.

I.4.4 – MISCELÂNIA

Após estudarmos operadores aritméticos, funções trigonométricas, funções numéricas e operadores relacionais, restam ainda um punhado de operadores e funções que devido características peculiares, mereciam uma ênfase e por essa razão figuram num subitem separado.

I.4.4.1 – OPERADORES

São dois, conforme mostramos abaixo:

Operador	Operação	
DIV	Retorna o quociente inteiro de uma divisão	
MOD	Retorna o resto de uma divisão	
Operador	**Operando**	**Saída**
DIV	Z	Z
MOD	Z	Z

Ambos operadores recebem de entrada números inteiros e devolvem números inteiros.

O operador **DIV** devolve o quociente inteiro da divisão entre inteiros, independentemente de haver ou não resto na divisão.

O operador **MOD** devolve o resto de uma divisão entre inteiros, portanto retorna também um número inteiro.

TEORIA

Considere a e b dois números inteiros. De acordo com o algoritmo da divisão de Euclides, podemos relacionar os números da seguinte forma:

$$a = b * \boldsymbol{q} + \boldsymbol{r}$$

Onde q e r são dois outros inteiros, respectivamente o quociente e o resto da divisão de a por b.

EXEMPLOS:

a	b	Pascal	Saída
5	3	$5\ DIV\ 3$	1
7	4	$7\ MOD\ 4$	3
8	11	$8\ DIV\ 11$	0
9	3	$9\ MOD\ 3$	0

I.4.4.2 – FUNÇÕES

São dois, conforme mostramos abaixo: