**LÓGICA DE PROGRAMAÇÃO**

**LÓGICA**

Lógica é uma palavra derivada do grego “logos”, que significa ciência do raciocínio e demonstração; seu significado está associado à coerência de racionalidade, sendo aplicada a várias ciências e não somente à matemática, como sugerido. Contudo, podemos entender a lógica de forma mais simples, como algo presente no nosso dia a dia: a arte de organizar nosso raciocínio para executar uma determinada tarefa. Por exemplo, usamos a lógica para resolver problemas matemáticos (sequências), charadas, testes de raciocínio; para realizar tarefas simples, como abastecer um veículo, ir para a escola, preparar uma receita; para resolver um problema, onde se torna necessário identificar o que deve ser feito bem como a ordem destas ações. Vale observar, portanto, que o conceito de lógica está diretamente relacionado com o raciocínio.

A lógica nos ajuda a:

- analisar situações de forma a construir argumentos mais convincentes e realistas

- analisar informações da forma mais clara e objetiva, evitando interpretações e conclusões equivocadas

- entender problemas de grande complexidade, ajudando a dividi-los em partes menores, facilitando a busca de soluções eficientes

- trabalhar o desenvolvimento do raciocínio, permitindo a análise mais precisa de situações e, consequentemente, tomar decisões mais adequadas

- ordenar nossos pensamentos, permitindo determinar uma sequência de ações para se atingir um objetivo desejado

Portanto a Lógica se faz necessária no nosso dia-a-dia, seja na vida pessoal, acadêmica ou profissional porque nos permite pensar de maneira clara e simples aplicar este raciocínio na solução de problemas, execução de tarefas, tomadas de decisões e etc.

**LÓGICA DAS PROPOSIÇÕES**

Parte da lógica que trata das proposições, que são sentenças que podem ser verdadeiras ou falsas. Ela é fundamental para a construção de algoritmos, pois permite tomar decisões baseadas em condições que envolvem variáveis lógicas, como "verdadeiro" (V) ou "falso" (F). Esses conceitos são amplamente usados em programação, especialmente na hora de criar algoritmos que fazem escolhas, verificações ou repetições, por meio de operadores lógicos como E, OU e NÃO.

### Operador E

Resulta em **verdadeiro** somente quando ambas as proposições envolvidas são verdadeiras. Se pelo menos uma das proposições for falsa, o resultado será **falso**.

Tabela verdade

|  |  |  |
| --- | --- | --- |
| 1º valor (A) | 2º valor (B) | Resultado |
| F | **F** | **F** |
| F | **V** | **F** |
| V | **F** | **F** |
| V | **V** | **V** |

**Operador OU**

Resulta em **verdadeiro** se pelo menos uma das proposições for verdadeira. O resultado só será **falso** quando ambas as proposições forem falsas.

Tabela verdade

|  |  |  |
| --- | --- | --- |
| 1º valor (A) | 2º valor (B) | Resultado |
| F | **F** | **F** |
| F | **V** | **V** |
| V | **F** | **V** |
| V | **V** | **V** |

**Operador NÃO**

Inverte o valor lógico da proposição. Se a proposição for verdadeira, o "NÃO" a transforma em falsa, e se for falsa, ela se torna verdadeira.

**Tabela verdade**

|  |  |
| --- | --- |
| 1º valor (A) | Resultado |
| F | **V** |
| V | **F** |

**Prioridade para resolução de expressões lógicas**

**1) Eliminar os parênteses**

**2) Operador não**

**3) Operador e**

**4) Operador ou**

O uso destes operadores é muito importante pois através deles podemos descrever inúmeras situações a serem avaliadas pelo computador e especificar o que deve sre feito diante do resultado destas análises (VERDADEIRO ou FALSO)

**Exemplos de resolução de equações lógicas**

A ←10

B ←25

C← 3.5

TESTE←”V” //verdadeiro

RESULTADO ← “F” //falso

1) (A+B=45) ou (TESTE=”V”)

(10+25=45) ou (“V”=”V”)

(35=45) ou (“V”=”V”)

F ou V = VERDADEIRO

2) (não(A=15)) e (RESULTADO=”V”)

(não(10=15)) e (“F”=”V”)

(não F) e F

V e F = FALSO

3) (A>B) e (NOME=”MARIA”) e (C=3.14)

(10>25) e (“JOAO”=”MARIA”) e (3.5=3.14)

F e F e F

F e F = FALSO

4) não ((A-B>5) e (C/2=1) ou (TESTE=”F”))

não((10-25)>5) e (3.5/2=1) ou (“V”=”F”)

não(-15>5) e (3.5/2=1) ou (“V”=”F”)

não (15>5) e (1.75=1) ou (“V”=”F”)

não F e F ou F

V e F ou F

F ou F = FALSO

**LÓGICA NATURAL**

Visa trabalhar a identificação dos passos necessários (em ordem lógica) para se alcançar um determinado objetivo e representa-los de forma mais informal, usando palavras/expressões do nosso cotidiano (linguagem natural).

***Ex1: Roteiro para preparar um misto quente***

1-Pegar um pão de sal

2-Cortar o pão ao meio

3-Pegar uma fatia de queijo

4- Colocar no pão

5-Pegar uma fatia de presunto

6-Colocar no pão

7-Fechar o pão

8-Colocar o misto na sanduicheira

***Ex2:Tirar dinheiro no banco***

1-Ir até uma agência

2-Encontrar um terminal que não esteja sendo usado

3-Inserir meu cartão

4-Escolher a opção de SAQUE

5-Digitar a quantia a ser retirada

6-Digitar a senha

7-Retirar o cartão

8-Pegar o dinheiro

***Ex3:Trocar o pneu furado de um carro***

1-Abrir o porta-malas

2-Pegar o estepe

3-Pegar o macaco

4-Afrouxar as porcas da roda

5-Colocar o macaco

6-Subir o carro usando o macaco

7-Retirar as porcas da roda

8-Retirar a roda/pneu furado

9-Colocar o estepe

10-Colocar as porcas na roda

11-Descer o carro

12-Apertar as porcas

Podem haver algumas variações nestes processos, pois um problema pode ter várias soluções. O importante aqui é observar que foram descritos os passos necessários na ordem correta para resolve-los.

**Exercícios**

1. Crie um roteiro lógico que descreva todos os passos para se calcular a média aritmética de 3 números
2. Crie um roteiro lógico para calcular o consumo de um veículo, a partir da distância percorrida e da quantidade de combustível usada
3. Crie um roteiro lógico que descreva todos os passos para calcular a densidade de um elemento químico a partir dos valores da sua massa e volume
4. Crie um roteiro lógico para que descreva todos os passos para calcular a área e o perímetro de um quadrado, a partir da medida de seu lado
5. Crie um roteiro lógico que descreva todos os passos para converter uma temperatura na escala Celsius para as escalas Kelvin e Fahrenheit

**LÓGICA DE PROGRAMAÇÃO**

Parte da lógica que tem como objetivo associar a lógica natural com a criação de códigos (algoritmos) a serem implementados e executados no computador, sendo um passo importante no estudo da programação de computadores.

**ALGORITMOS**

A criação de algoritmos é uma das formas existentes para descrever o raciocínio lógico que será usado na construção de um programa.

Basicamente, um algoritmo é um roteiro de ações, descritas numa ordem lógica, que devem ser executadas pelo computador para que o objetivo seja alcançado. Porém, estas ações não são representadas de qualquer forma, mas através de algumas palavras específicas, conhecidas também como palavras reservadas de uma pseudolinguagem chamada PORTUGOL.

A seguir temos o exemplo de roteiro lógico que descreve todos os passos para calcular área e perímetro de um retângulo:

1 - Saber a medida da base

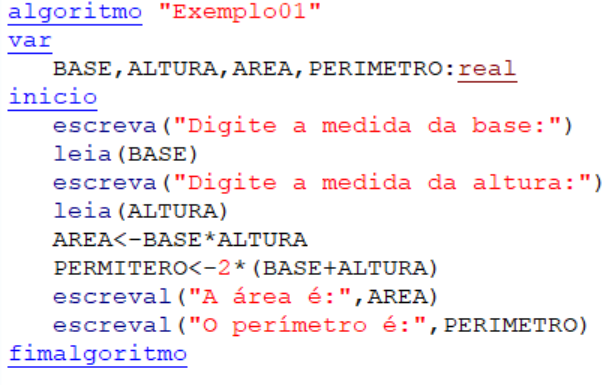
2 - Saber a medida da altura

3 - Calcular a área do retângulo (multiplicar base pela altura)

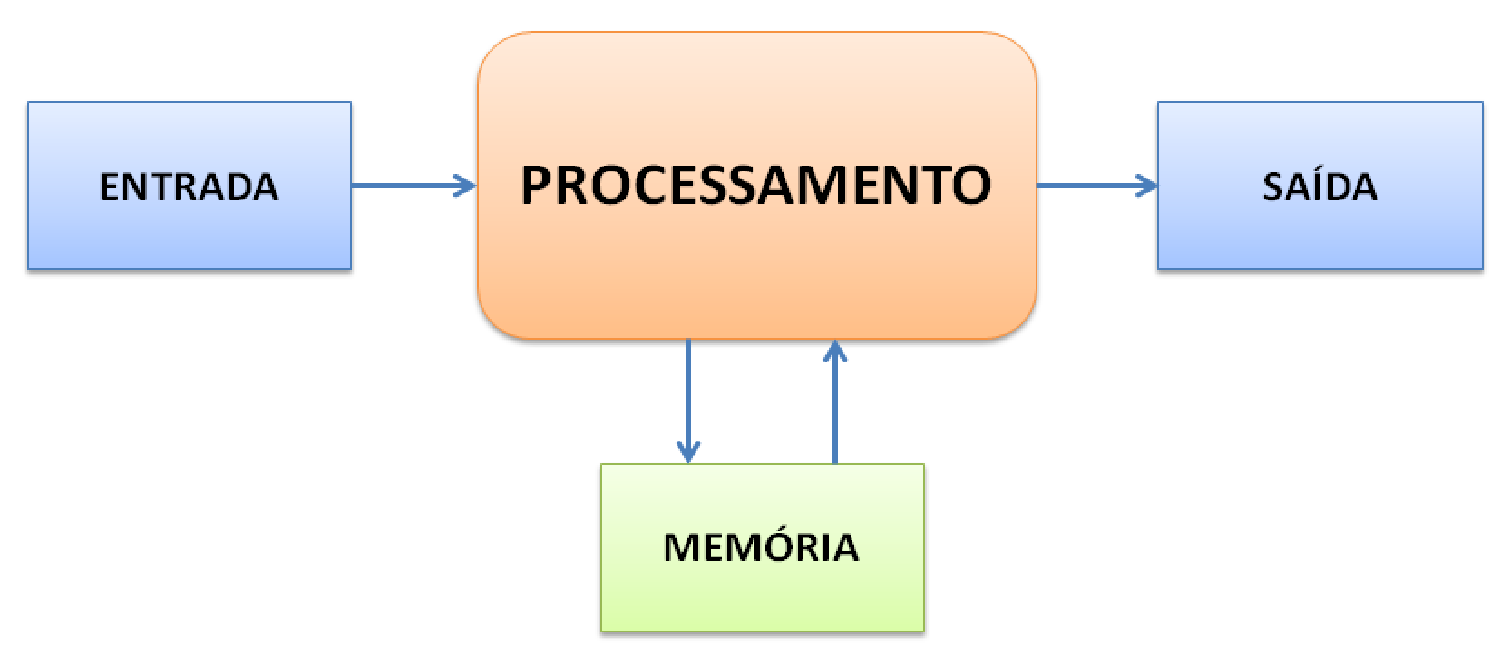
4 - Calcular o perímetro do retângulo (multiplicar a soma da base e altura por 2)

5 - Exibir os resultados

Agora temos um algoritmo onde a lógica do roteiro acima foi implementada:



Basicamente, algoritmos são compostos por operações de ENTRADA, PROCESSAMENTO e SAÍDA.



**ENTRADA**

Entrada de dados é o processo onde o computador solicita que o usuário digite os valores que ele precisa para gerar os resultados desejados; o fornecimento é feito através de teclado, leitor de código de barras, etc

**PROCESSAMENTO**

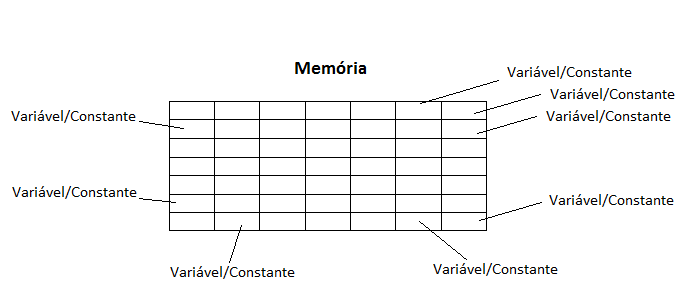
Ações que o computador deverá executar para obter os resultados desejados; estas ações devem ser descritas pelo programador, numa ordem lógica, através dos comandos da linguagem escolhida

**SAÍDA**

Processo onde o computador exibe para o usuário (na tela ou impressora) os resultados gerados durante o processamento.

**MEMÓRIA**

Dispositivo interno do computador que é responsável pelo armazenamento dos dados que o computador recebe ou produz durante o processamento. Pode ser entendida como uma grande “caixa”, dividida em várias “partes”, onde cada uma é responsável por armazenar um valor. Estas partes são chamadas de variáveis. Para facilitar a compreensão, a memória pode ser representada da seguinte forma:



Durante o processo de desenvolvimento do algoritmo ou programa, cabe ao programador a tarefa de identificar as variáveis que serão usadas.

**ESTRUTURA DE UM ALGORITMO**

algoritmo “nome”

var

DECLARAÇÃO DE VARIÁVEIS

inicio

COMANDOS

fimalgoritmo

nome: identificador interno do algoritmo

DECLARAÇÃO DE VARIÁVEIS: onde são especificadas os espaços de memória que o algoritmo deverá utilizar para guardar os valores recebidos ou produzidos por ele

COMANDOS: onde são descritas as instruções que especificam cada ação que o algoritmo deve executar

**VARIÁVEIS E TIPOS DE DADOS**

O armazenamento de dados em uma aplicação é fundamental, uma vez que é através deste processo que os dados recebidos ou gerados durante o processamento ficam à disposição, execução da aplicação. Sem este processo, seria impossível ao computador processar dados e, com isso, obter informações.

No tópico anterior, vimos que a memória é como uma grande “caixa”, cheia de “divisões”. Cada uma dessas “divisões” fica responsável por armazenar um dado (valor). Assim, qualquer dado que uma aplicação receba ou calcule deverá ser armazenado em uma dessas partes, para não se perder durante o processamento.

Mas, se existem tantas divisões, como o computador sabe onde armazenar ou buscar um valor? Simples. Ao criar um algoritmo ou programa, você deve especificar quantos espaços ele utilizará. É um processo conhecido como **declaração de variáveis.**  Assim, você deverá declarar nos seus algoritmos todas as variáveis que eles utilizarão, de acordo com algumas regras do algoritmo ou linguagem em questão. Como fazemos isso? Devemos identificar quantas variáveis o algoritmo/programa deve usar, escolhendo, para cada uma, um nome, que deve seguir a estas regras:

* Começar com letra
* Usar apenas letras, dígitos e o caracter “\_”
* Não podem ser iguais às palavras reservadas
* Devem ser significativos

Contudo, não basta apenas citar as variáveis que serão utilizadas. É necessário também associar cada uma a um **tipo de dado**. Mas o que seria isso? Tipos de dados são palavras reservadas que indicam como o valor de uma variável pode ser formado. Assim, ao fazer a declaração, você dá nome à sua variável e define qual será o tipo de dado dela. A tabela a seguir descreve os tipos de dados usados em algoritmos:

|  |  |
| --- | --- |
| **Descrição** | **Em algoritmo** |
| Valores numéricos inteiros, ou seja, não possuem casas decimais | inteiro |
| Valores numéricos reais, ou seja, possuem casas decimais | real |
| Valores formados por um ou mais caracteres (letras, dígitos, símbolos) | caracter |
| Valores lógicos (VERDADEIRO/FALSO, TRUE/FALSE) | lógico |

A declaração de variáveis no Visualg é feita na seção VAR:

Var

NOME\_VARIAVEL:tipo\_de\_dado

A atribuição de valores é feita através do símbolo <-. Valores do tipo caracter devem ser colocados entre aspas.

**Exemplos**

1) Faça a declaração de variáveis para armazenar os seguintes dados de um veículo: placa,chassi, marca, modelo, ano de fabricação e valor de venda.

2) Faça a declaração de variáveis para armazenar os seguintes dados de um produto: código, descrição, valor, data de fabricação e data de validade

3) Faça a declaração de variáveis para armazenar os seguintes dados de um funcionário: CPF, nome, data de nascimento, endereço de email, número do telefone fixo e número de telefone celular.

4) Faça a declaração de variáveis para armazenar os seguintes dados de um aluno: nome, data de nascimento, telefone, peso e altura

5) Escreva as instruções para atribuir às variáveis do Exemplo 1 os valores HBA0001,9BD555,FORD,FIESTA,2011 e 23500.00

6) Escreva as instruções para atribuir às variáveis do Exemplo 2 os valores 1, SUCO DE ABACAXI, 7.50, 01/02/2023 e 30/06/2023

7) Escreva as instruções para atribuir às variáveis do Exemplo 3 os valores 000000000-01, JOAO DA SILVA, 10/07/1989, [joaodasilva@mymail.com](mailto:joaodasilva@mymail.com), (31)3790-0000 e (31)90010-0001

8) Escreva as instruções para atribuir às variáveis do Exemplo 4 seus dados pessoais

**OPERADORES**

São símbolos ou termos utilizados nos algoritmos para representar alguns tipos de operações que devem ser executadas de acordo com a situação. Embora este termo sugira algo como cálculos (operações matemáticas), o conceito de operadores é mais extenso.

Em algoritmos, temos três tipos de operadores:

**1) ARITMÉTICOS OU MATEMÁTICOS**

Usados para criação de expressões matemáticas. Através deles, o programador pode descrever qualquer cálculo que deve ser realizado pelo computador. Os principais operadores aritméticos são:

|  |  |
| --- | --- |
| **Operador** | **Significado** |
| + | Soma |
| - | Subtração |
| \* | Multiplicação |
| / | Divisão real (resultado com parte decimal) |
| div | Quociente da divisão (resultado inteiro) |
| mod | Resto da divisão (resultado inteiro) |
| ^ ou \*\* | Potência (no visualG se usa apenas o operador ^) |
| raizq(valor) | Raiz quadrada (valor é o número do qual se deseja calcular a raiz) |

Para atribuir um valor à uma variável, deve-se usar o operador <-

**Exemplos**

|  |  |
| --- | --- |
| Expressão | Representação em algoritmos |
| SOMA = A+B+C | SOMA<- A+B+C |
| DELTA = B2-4AC | DELTA<-(B^2)-4\*A\*C |
| VALOR = RAIZ(Y)-3 | VALOR <- raizq(Y) – 3 |
| JUROS=5% DO PRECO | JUROS<-5\*PRECO/100 |
| AUMENTO=12% DO SALARIO | AUMENTO<-12\*SALARIO/100 |
| MEDIA = X1+X2+X3  3 | MEDIA<-(X1+X2+X3)/3 |

**2) RELACIONAIS**

Usados para descrever comparações (condições) entre dados. São eles:

|  |  |
| --- | --- |
| **Operador** | **Significado** |
| > | Maior que |
| < | Menor que |
| = | Igual a |
| >= | Maior ou igual a |
| <= | Menor ou igual a |
| <> | Diferente de |

**3) LÓGICOS**

Operadores usados para criar expressões lógicas (combinações de condições), obtendo como resultado final um valor booleano (verdadeiro ou falso)

|  |  |
| --- | --- |
| **Operador** | **Significado** |
| E | Conjunção |
| OU | Disjunção |
| NAO | Negação |

**Atividades Propostas**

Escrever as expressões abaixo na forma computacional

a) FATOR=X/Y+3xB

b) VOLUME= 4/3 π r³

c) DIAGONAL= A² = B² + C²

d) INDICE = raiz(A+B-C)

e) X1=-B-raiz(DELTA)/(2xA)

f) DIGITO=resto da divisão de SOMA por 5

g) RESULTADO=resto da divisão de X por Y mais quociente da divisão de

X por Y

**COMANDOS BÁSICOS**

Como já foi discutido, um algoritmo é um conjunto de ações que, ao serem executadas, levarão ao objetivo proposto. Porém, estas ações devem ser descritas através dos comandos do algoritmo. Podemos entender comandos como palavras que representam instruções que possuem um significado pré-estabelecido.

**1) SAÍDA DE DADOS (ESCRITA)**

Usado para fazer com que o algoritmo mostre na tela uma mensagem, valor ou conteúdo de uma variável. Representado pelas palavras **imprima, escreva ou escreval (as duas últimas são usadas no Visualg).**

SINTAXE:

escreva ()

escreval ()

Entre parênteses deve ser especificado o que o algoritmo deve exibir.

Obs.:

- Mensagens devem estar entre aspas

- É possível exibir mais de um elemento, bastando separá-los por vírgulas

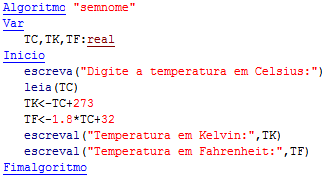
**2) ENTRADA DE DADOS (LEITURA)**

Usado para fazer com que algoritmo solicite ao usuário o fornecimento de algum dado, via digitação, leitura de código de barras, etc. O valor é transferido para a variável especificada na leitura, para que possa ser utilizado no restante do algoritmo. O comando usado para representar a entrada de dados em um algoritmo é o **leia**.

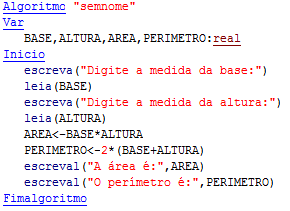
leia(NOME DA VARIÁVEL)

**Exemplos**

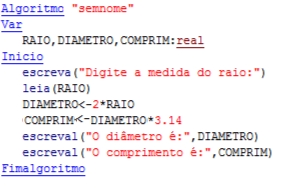
1) Criar um algoritmo que leia uma temperatura na escala Celsius, convertendo-a para as escalas Kelvin e Fahrenheit, mostrando estes resultados na tela.



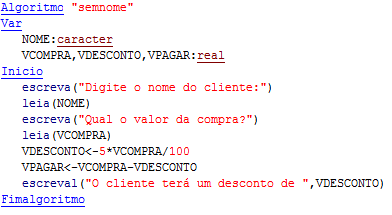
2) Criar um algoritmo que leia os valores da base e da altura de um retângulo. Pede-se que o algoritmo calcule e mostre os valores da área e do perímetro deste retângulo.



3) Criar um algoritmo que leia a medida do raio de uma circunferência. Pede-se que o algoritmo calcule e mostre os valores do diâmetro e do comprimento desta circunferência;



4) Criar um algoritmo para ler o nome e valor da compra efetuada por um cliente das Lojas X. Pede-se que o algoritmo calcule e mostre o valor do desconto que o cliente terá direito (que corresponde a 5% do valor da compra) e o valor que o cliente deverá pagar pela compra.



**Atividades Propostas**

1) Criar um algoritmo que leia a medida do lado de um quadrado. Pede-se que o algoritmo calcule e escreva a área deste quadrado (lado x lado) e o seu perímetro.

2) Criar um algoritmo que leia um número real. Pede-se que o algoritmo calcule e escreva o valor do quadrado deste número e da sua raiz quadrada

3) Criar um algoritmo que leia o nome e o ano de nascimento de uma pessoa. Pede-se que o algoritmo calcule e escreva qual será a idade desta pessoa ao final deste ano, junto com seu nome.

4) Criar um algoritmo que leia dois números inteiros; pede-se que o algoritmo calcule e escreva:

1. O resultado da divisão real (com casas decimais)
2. O quociente da divisão entre os números
3. O resto da divisão entre os números

5) Crie um algoritmo que leia o valor de venda de um veículo, calcule e escreva o valor do IPVA (imposto) deste veículo, que corresponde a 4% do seu valor de venda e o valor do seu seguro, que corresponde a 3% do valor de venda

6) Criar um algoritmo que leia o nome e salário bruto de um empregado. Pede-se que o programa calcule e escreva:

1. O valor do desconto previdenciário (11% sobre o salário bruto)
2. O valor do desconto sindical (4% sobre o salário)
3. O salário líquido (salaŕio bruto menos os descontos)

7) Criar um algoritmo que leia uma quantidade de horas e uma quantidade de minutos. Pede-se que ele converta esse horário em minutos e em segundos. Considere que uma hora tem 60 minutos e um minuto tem 60 segundos

8) Criar um algoritmo que leia os seguintes dados de uma cidade: nome, número de habitantes do sexo masculino, número de habitantes do sexo feminino e área (em quilômetros quadrados). Pede-se que o algoritmo calcule e escreva:

1. A população total da cidade
2. A densidade demográfica da cidade (população dividida pela área)

9) Criar um algoritmo que leia uma distância em metros, calcule e escreva o valor correspondente em quilômetros (dividir por 1000), em milhas (dividir por 1609) e em jardas (multiplicar por 1,09631)

10) Criar um algoritmo que leia nome, peso (em kg) e altura (em metros) de uma pessoa. Pede-se que o algoritmo calcule o valor do IMC desta pessoa e mostre o resultado junto do seu nome. Ex: Fulano, seu IMC é 22,3.

11) Criar um algoritmo que leia as medidas da base maior e base menor de um trapézio. Pede-se que o algoritmo calcule e escreva a medida da área deste trapézio ((base maior + base menor)\*altura)/2

12) Criar um algoritmo que leia os seguintes dados de uma pessoa inscrita para uma prova de seleção na Empresa X: nome, quantidade de acertos na prova de Matemática e quantidade de acertos na prova de Informática. Pede-se que o algoritmo calcule escreva os seguintes resultados:

1. Pontuação na prova de Matemática (cada acerto vale 1,5 ponto)
2. Pontuação na prova de Informática (cada acerto vale 2,5 pontos)
3. Pontuação total ao final da seleção

13) Criar um algoritmo que leia os seguintes dados de uma equipe sobre sua participação em um campeonato de futebol: nome, número de vitórias, número de empates, número de derrotas, quantidade de gols marcados e quantidade de gols sofridos. Pede-se que o algoritmo calcule e escreva os seguintes resultados:

1. Total de jogos disputados (somar vitórias, empates e derrotas)
2. Saldo de gols da equipe (diferença entre gols marcados e gols sofridos)
3. Média de gols marcados por jogo
4. Média de gols sofridos por jogo

14) Criar um algoritmo que leia o nome e a altura (em metros) de uma pessoa. Pede-se que o algoritmo calcule o peso ideal desta pessoa, considerando as fórmulas a seguir:

* Para homens = (72.6\*altura) - 58
* Para mulheres = (62.1\*altura) - 44.7

O algoritmo deve mostrar o valor de cada um dos cálculos acima.

15) Crie um algoritmo que leia o valor da compra efetuada por um cliente na Loja X. Sabendo que ele pode parcelar a compra em até 3 vezes sem juros, pede-se que o algoritmo calcule e escreva o valor de cada parcela para pagamento em 2x e o valor de cada parcela para pagamento em 3x.

**ESTRUTURA CONDICIONAL**

Até agora, todos os algoritmos estudados possuem algo em comum: o resultado produzido por eles é consequência da execução ordenada e sequencial das instruções definidas para cada um. Ou seja, um comando é executado após o término da execução do comando anterior. Porém, existem situações onde esse fluxo de execução deverá ser “quebrado”. Por exemplo, um algoritmo que receba a nota de um aluno com o objetivo de determinar sua situação (aprovado ou reprovado) terá, num determinado momento, dois caminhos a seguir, dependendo da nota em questão. Assim, torna-se necessário utilizar uma estrutura que permita ao desenvolvedor especificar as decisões que o computador deve tomar e o caminho a ser seguido para cada situação possível – a estrutura condicional.

De forma mais simples, podemos entender que uma estrutura condicional permite especificar uma ou mais condições para o algoritmo analisar e, para cada um dos possíveis resultados desta análise (verdadeiro ou falso), descrever um bloco de comandos a ser executado..

Em algoritmos, temos as estruturas condicionais **simples, compostas** e **encadeadas.**

# ESTRUTURA CONDICIONAL SIMPLES

se CONDIÇÃO(ÕES) entao

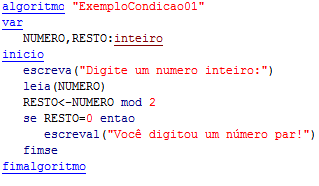
BLOCO DE COMANDOS

fimse

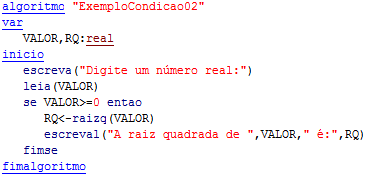
Neste tipo de estrutura, o algoritmo analisa a(s) condição(ões) descrita(s). Se o resultado for VERDADEIRO, ele executa o bloco de comandos especificado na sentença **entao**. Caso o resultado seja FALSO, a estrutura é ignorada e o algoritmo segue executando os comandos posteriores.

**Exemplos**

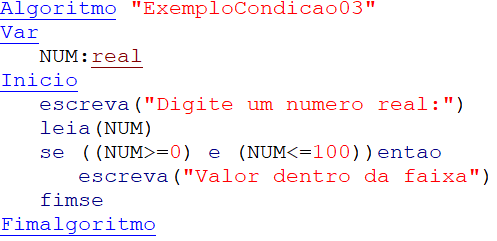
1) Criar um algoritmo que leia um número inteiro e verifique se ele é par. Caso seja, exibir a mensagem “Você digitou um número par!”



2) Criar um algoritmo que leia um número, calculando e escrevendo sua raiz quadrada apenas se ele for igual ou maior que 0 (zero).



3) Criar um algoritmo que leia um valor numérico real. Pede-se que o algoritmo verifique e escreva a mensagem “Valor dentro da faixa” caso o número digitado esteja no intervalo entre 0 e 100.



**Atividades Propostas**

1) Criar um algoritmo que leia dois números inteiros. Pede-se que o algoritmo calcule e escreva a diferença e a divisão real dos valores caso o primeiro número seja maior que o segundo

2) Criar um algoritmo que leia o nome de um aluno e a nota que ele obteve em cada uma das duas provas realizadas no curso. Pede-se que o algoritmo:

* Calcule e mostre a nota final do aluno (média das duas notas)
* Verifique e mostre a mensagem “Aluno aprovado” caso a nota final seja igual ou superior a 70

3) Criar um algoritmo que leia o nome e valor de vendas de um empregado da Loja X. Pede-se que o algoritmo calcule e escreva o valor da comissão (3% do valor das vendas) somente se seu valor de vendas for acima de 50000

4) Crie um algoritmo para ler o nome e salário de um empregado que trabalha na Indústria X. Pede-se que o algoritmo calcule e escreva o valor do aumento que este empregado vai ganhar (5% sobre o salário), considerando que ele só terá direito a esse aumento se o seu salário for entre 1100 e 2500 reais.

5) Crie um algoritmo para ler o nome, peso e altura de uma pessoa. Pede-se que o algoritmo calcule e escreva o IMC da pessoa (peso dividido pelo quadrado da altura); caso o valor do IMC seja acima de 25, o algoritmo deve exibir a mensagem “Você está acima do peso ideal”

6) Crie um algoritmo que leia o nome e a idade de uma pessoa. Caso esta idade seja maior ou igual a 18, o algoritmo deverá exibir uma mensagem contendo o nome da pessoa e a frase “Você é maior de idade!”

7) Criar um algoritmo que leia um valor numérico inteiro. Se este valor for 2, 5 ou 9, o programa deverá exibir a mensagem “Valores aceitos”

8) Criar um algoritmo que leia um número real. Caso ele seja maior que 100, o algoritmo deve calcular e mostrar o quadrado deste número, sua raiz quadrada e a sua metade

9) Criar um algoritmo que leia dois números inteiros. Se o primeiro for menor que o segundo, calcular o quociente e o resto da divisão entre eles; se o primeiro for igual ou maior que o segundo, calcular a raiz quadrada de cada um

10) Criar um algoritmo que leia um número inteiro. Caso este número seja igual a 1,3 ou 5, o algoritmo deve atribuir a uma variável INDICE (declarada com inteiro) o valor 333.

11) Criar um algoritmo que o cpf leia a idade de uma pessoa que deseja comprar ingresso para um evento. Sabe-se que o ingresso custa 120 reais, mas pessoas com idade entre 18 e 40 anos tem direito a 25% de desconto. Pede-se que o algoritmo calcule e mostre o valor a ser pago.

12) Criar um algoritmo que leia o nome de um aluno, sua nota na avaliação final e a quantidade de faltas que ele teve. Caso a nota seja igual ou superior a 70 e ele tenha no máximo 10 faltas, o algoritmo deve mostrar que ele foi aprovado

13) Criar um algoritmo que leia o valor da compra de um cliente e a forma de pagamento. Caso ele deseje pagar com PIX ou DINHEIRO, terá desconto de 5%. Pede-se que o algoritmo calcule e mostre o valor da compra e o valor a pagar

# ESTRUTURA CONDICIONAL COMPOSTA

# se CONDIÇÃO(ÕES) entao

BLOCO DE COMANDOS 1

senao

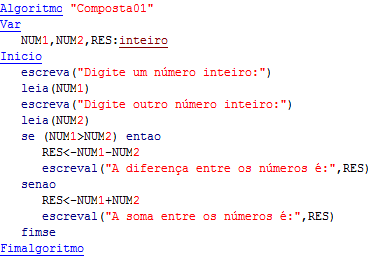
BLOCO DE COMANDOS 2

fimse

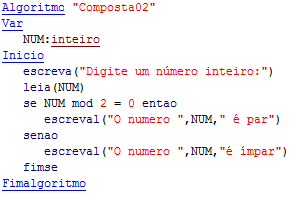
Neste tipo de estrutura, o algoritmo analisa a(s) condição(ões) descrita(s). Se o resultado for VERDADEIRO, ele executa o bloco de comandos especificado na sentença **entao**; caso o resultado seja FALSO, ele executa o bloco de comandos descrito na sentença **senao.**

**Exemplos**

1) Criar um algoritmo que leia dois números inteiros; se o primeiro número for maior que o segundo, deve-se calcular a diferença entre eles; caso contrário, deve-se calcular a soma entre eles.

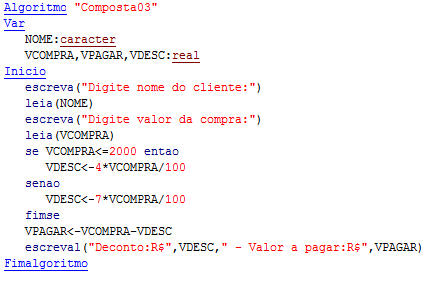


2) Criar um algoritmo que leia um número inteiro e verifique se ele é divisível por 2. Se for, exibir a mensagem “Você digitou um número par!”; caso contrário, ele deve exibir a mensagem “Você digitou um número ímpar!”



3) Criar um algoritmo que leia o nome e o valor da compra efetuada por um cliente da Loja X. Pede-se que o programa calcule o valor do desconto que o cliente terá direito e o valor a pagar pela compra, conforme descrição a seguir:

* Valor da compra até 2000 reais, desconto de 4%
* Valor da compra acima de 2000 reais, desconto de 7%



**Atividades Propostas**

1) Crie um algoritmo que leia dois números reais. Pede-se que o algoritmo calcule e escreva:

* A soma e o produto dos números se o primeiro for menor ou igual ao segundo
* A raiz quadrada do primeiro e a raiz quadrada do segundo se o primeiro for maior que o segundo

2) Crie um algoritmo que leia o nome e o valor da compra efetuada por um cliente da Loja X. Pede-se que o algoritmo mostre o brinde que o cliente vai ganhar, de acordo com o valor da compra efetuada, conforme descrito a seguir:

* se o valor da compra for até 1000 reais, o brinde será uma agenda
* se o valor da compra for superior a 1000 reais, o brinde será um pen-drive

3) Crie um algoritmo que leia o nome de um aluno matriculado no curso de Lógica da Escola X e as notas que ele obteve em cada uma das duas provas realizadas. Pede-se que o algoritmo calcule e escreva:

a) A nota final do aluno, considerando que ela corresponde à média aritmética das notas obtidas

b) A situação do aluno (APROVADO se nota final >= 75 ou REPROVADO se nota final abaixo de 75)

4) Criar um algoritmo que leia a população de uma cidade e sua área (em metros quadrados). pede-se que o algoritmo calcule e escreva:

a) A densidade demográfica da cidade (população/área)

b) A classificação da cidade, que pode ser “ALTA” se a densidade demográfica for acima de 10 ou “BAIXA” se a densidade demográfica for abaixo ou igual a 10.

5) Crie um algoritmo que leia três números correspondentes aos coeficientes de uma equação do segundo grau (A, B e C). Pede-se que o algoritmo:

a) Calcule o valor do DELTA (B ao quadrado - 4xAxC)

b) Verifique e calcule, se possível, as raízes da equação sabendo que:

* se o delta for menor que 0, não é possível calcular as raízes
* se o delta for maior ou igual a 0, devem ser calculadas e mostradas as duas raízes

6) Criar um algoritmo que leia dois números reais. Pede-se que o algoritmo calcule e escreva os seguintes resultados:

* O quadrado e a raiz quadrada do primeiro número, caso a soma dos valores lidos seja acima de 100
* A diferença e a média aritmética dos valores caso a soma deles seja igual ou inferior a 100.

7) Criar um algoritmo que leia a renda mensal de um cliente do Banco X. Pede-se que o algoritmo verifique se ele terá direito ao chamado Crédito Especial e, caso tenha, qual será o valor. De acordo com o banco, apenas clientes com renda superior a 7000 tem acesso ao crédito, cujo valor é de 15% da renda mensal. Pede-se que o algoritmo mostre na tela o valor do crédito ou uma mensagem informando que o cliente não tem direito.

8) Criar um algoritmo que leia o nome, nota final e número de faltas de um aluno matriculado no Curso X de Programação. Sabendo que o aluno só estará aprovado se tiver nota final igual ou superior a 70 e também no máximo 8 faltas, pede-se que o algoritmo verifique e mostre a situação deste aluno (Aprovado ou Reprovado).

9) Criar um algoritmo que leia o nome, cargo e salário de um empregado. Pede-se que o algoritmo calcule e escreva o valor do aumento e o novo salário do empregado, considerando que terão direito a aumento de 5% apenas os funcionários de cargo VENDEDOR que tenham salário até 2500 reais; para os demais funcionários o aumento será de 3%.

10) Criar um algoritmo que leia um ano, verifique e mostre se ele é ou não bissexto, (obs: ano bissexto é aquele que é divisível por 4).

11) Criar um algoritmo que leia um número inteiro; se este número for acima de 25 e abaixo de 75 , deve ser exibida a mensagem “Valor na faixa”, caso contrário, deve ser exibida a mensagem “Valor não pode ser aceito”.

12) Criar um algoritmo que leia o login de uma pessoa e sua senha. Caso o login seja ADMIN e a senha seja 12345, o algoritmo deve exibir a mensagem “ACESSO PERMITIDO”, caso contrário deve exibir a mensagem “LOGIN/SENHA INCORRETOS, ACESSO BLOQUEADO”.

13) Criar um algoritmo que leia um número inteiro. Se este número for igual a 1 ou 7 o algoritmo deve exibir a mensagem “Final de semana”, caso contrário a mensagem exibida será “Dia útil”.

14) Criar um algoritmo que leia o cpf, o valor da compra efetuada e a forma de pagamento (1-Pix/2-Dinheiro/3-Cartao) escolhida por um cliente das Lojas X. Pede-se que o algoritmo calcule e mostre o valor do desconto e o valor a pagar pela compra, considerando que este será de 5% se o cliente pagar em PIX ou DINHEIRO (caso pague no CARTAO, o desconto será de apenas 2%)

**ESTRUTURA CONDICIONAL ENCADEADA**

se CONDIÇÃO(ÕES)

entao BLOCO DE COMANDOS 1

senao

se CONDIÇÃO(ÕES)

entao BLOCO DE COMANDOS 2

senao CONDIÇÃO(ÕES) ou BLOCO DE COMANDOS

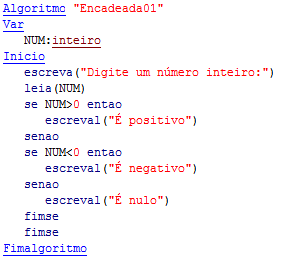
fimse

fimse

Neste caso, deve-se descrever na primeira sentença **entao** o bloco de comandos que o algoritmo deve executar caso o resultado da primeira condição ou expressão lógica seja VERDADEIRO e, na sentença **senao,** deve-se descrever outra condiçãoa ser analisada**.**

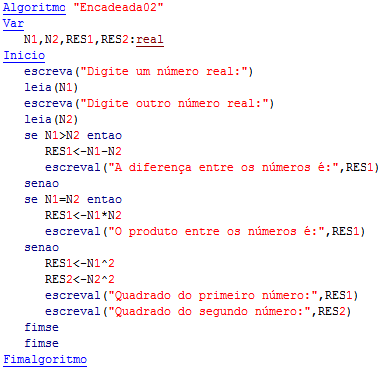
**Exemplos**

1) Criar um algoritmo que leia um número inteiro; pede-se verificar se o número é positivo, negativo ou nulo.



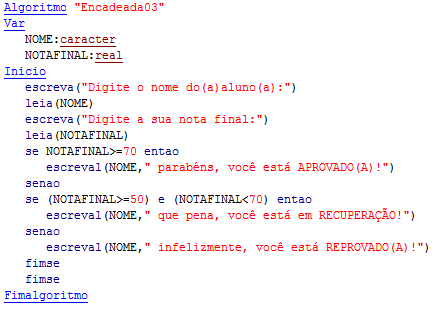
2) Criar um algoritmo que leia dois números reais. Pede-se que o algoritmo calcule e escreva:

* A diferença entre o primeiro número e o segundo, apenas se o primeiro for maior que o segundo
* O produto entre eles, caso sejam iguais
* O quadrado de cada número, apenas se o primeiro for menor que o segundo



3) Criar um algoritmo que leia o nome e a nota final de um aluno em Lógica. Pede-se que o algoritmo verifique e mostre a situação do aluno ao final do curso,considerando que:

* se o aluno tiver nota igual ou superior a 70, estará APROVADO
* se o aluno tiver nota entre 50 e 70, estará em RECUPERAÇÃO
* se o aluno tiver nota abaixo de 50, estará REPROVADO



4) Criar um algoritmo que leia o valor da compra efetuada por um cliente da Loja X. Pede-se que o algoritmo calcule e escreva:

1. O valor do desconto que o cliente terá, de acordo com a tabela a seguir:

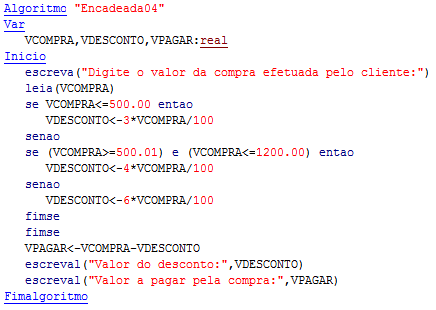
VALOR DA COMPRA DESCONTO

até 500,00 3% sobre valor da compra

500,01 a 1200,00 4% sobre valor da compra

acima de 1200,00 6% sobre valor da compra

1. Valor a pagar pela compra

‘

**Atividades Propostas**

1) Criar um algoritmo que leia o nome de um cliente e o valor da compra que ele efetuou na Loja X. Pede-se que o algoritmo verifique e mostre qual o brinde que o cliente receberá, conforme descrição a seguir:

VALOR DA COMPRA BRINDE

De 100,00 a 250,00 Chaveiro

De 250,01 a 500,00 Agenda

Acima de 500,00 Camiseta

2) Criar um algoritmo que leia o nome de uma pessoa, seu peso e sua altura. Pede-se que o algoritmo calcule e escreva:

1. O valor do IMC desta pessoa, cujo cálculo é o peso dividido pelo quadrado da altura
2. A condição física da pessoa, que pode ser:

* ABAIXO DO PESO IDEAL se o IMC for abaixo de 20
* NO PESO IDEAL se o IMC for entre 20 e 25
* ACIMA DO PESO IDEAL se o IMC for acima de 25

3) Criar um algoritmo que leia a temperatura na escala Celsius e uma opção de conversão, que pode ser 1-Para Kelvin, 2-Para Fahrenheit ou 3-Para Reamur. O algoritmo deverá calcular e mostrar a temperatura na escala escolhida (pesquisa as fórmulas para conversão).

4) Criar um algoritmo que leia o valor da compra feita por um cliente na Loja X e a opção de pagamento escolhida para parcelar a compra, que pode ser: A-Pagamento em 3 vezes, B-Pagamento em 5 vezes, C-Pagamento em 8 vezes. Pede-se que o algoritmo calcule e escreva o valor de cada parcela

5) Criar um algoritmo que leia um número inteiro. Pede-se que o algoritmo mostre uma mensagem de acordo com o valor digitado, conforme descrito a seguir:

NÚMERO DIGITADO MENSAGEM

1, 2 ou 5 OK

3, 4 FORA DO PADRÃO

demais valores MARCAÇÃO INVÁLIDA

6) Criar um algoritmo que leia o valor da compra efetuada por um cliente das LOJAS X. Pede-se que o algoritmo calcule e mostre o valor do desconto e o valor que o cliente deverá pagar pela compra. Segue a tabela de descontos:

VALOR DA COMPRA DESCONTO

Até 350,00 0

350,01 a 750,00 3% sobre o valor da compra

acima de 750,00 5% sobre o valor da compra

7) Criar um algoritmo que leia o nome de um instrutor de escola X, seu código de atuação e quantidade de aulas dadas no mês. Pede-se que o algoritmo calcule e escreva:

1. O salário base do instrutor (número de aulas \* valor da aula)
2. O desconto no salário do instrutor (10% do salário base)
3. O salário final (salário base - desconto)

A tabela com os códigos de atuação está descrita a seguir:

COD.ATUAÇÃO VALOR DA AULA

1 ou 3 35,87

2 28,74

4 ou 5 22,89

8) Criar um algoritmo que leia um peso em quilos e uma opção de conversão de unidade (1-Libra/2-Onça/3-Toneladas). Pede-se que o algoritmo converta e mostre o peso digitado para a unidade escolhida. Pesquise as fórmulas de conversão na internet.

9) Criar um algoritmo que leia o valor do abastecimento efetuado por um cliente dos Postos Alfa. Pede-se mostrar qual será o brinde do cliente, de acordo com a seguinte tabela:

VALOR DO ABASTECIMENTO BRINDE

até 80,00 Não tem

80,01 a 140,00 Ducha simples

140,01 a 200,00 Ducha com secagem

200,01 ou mais Lavagem com cera

10) Criar um algoritmo que leia o CPF, nome e valor da compra efetuada por um cliente da Loja XYZ de Departamentos. Sabe-se que o cliente terá direito a alguns cupons que serão usados em um sorteio, conforme tabela a seguir:

VALOR DA COMPRA NUMERO DE CUPONS

até 200 reais 0

200,01 a 550,00 reais 2

550,01 a 999,99 5

1000,00 ou mais 10

Pede-se que o algoritmo verifique e mostre quantos cupons o cliente vai receber

11) Criar um algoritmo que leia o valor do empréstimo contraído por um cliente da Financeira XYZ e a opção de pagamento que ele escolheu. Pede-se calcular e mostrar:

1. Valor dos juros
2. Valor total do emprestimo
3. Valor de cada parcela

As opções de pagamento são:

1 - Cinco parcelas (juros de 20% sobre o emprestimo)

2 - OIto parcelas (juros de 30% sobre o emprestimo)

3 - Doze parcelas (juros de 40% sobre o emprestimo)

4 - Quinze parcelas (juros de 65% sobre o emprestimo)

**ESTRUTURA DE SELEÇÃO**

Em alguns casos, a estrutura condicional pode ser tornar um tanto complexa. Imagine uma situação onde você precisa testar 9 possíveis valores para uma variável e, diante de cada um, determinar uma rotina para ser executada. Provavelmente, você teria uma estrutura SE bastante complicada, extensa e até mesmo difícil de compreender. Para situações como essa, é recomendável o uso de uma estrutura de seleção, representada da seguinte forma:

Escolha VARIAVEL

Caso PRIMEIRO VALOR

BLOCO DE COMANDOS 1

Caso SEGUNDO VALOR

BLOCO DE COMANDOS 2

Caso TERCEIRO VALOR

BLOCO DE COMANDOS 3

...

Outro Caso

BLOCO DE COMANDOS N

FimEscolha

Este tipo de estrutura é bem simples de montar: à frente da palavra Escolha devemos especificar a variável cujo valor será analisado. Em cada sentença caso identificamos um valor que a variável poderá assumir, descrevendo em seguida a rotina que deverá ser executada. Opcionalmente, podemos definir uma rotina para a sentença outro caso, que só será executada caso a variável receba um valor diferente dos especificados nas sentenças caso.

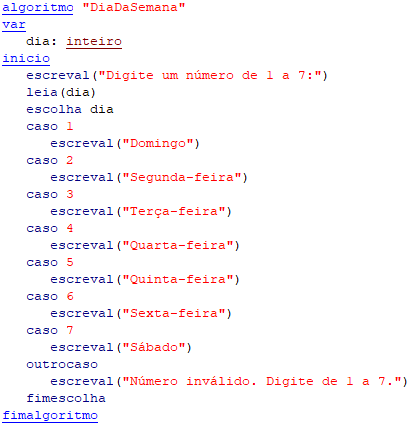
A grande vantagem dessa estrutura é a rapidez no processamento. Ela vai direto para a rotina que deve ser executada (numa estrutura Se com várias condições, cada condição é testada até que se encontre a verdadeira). Por isso é chamada de estrutura de seleção.

Infelizmente, cada linguagem tem a sua limitação para uso desse tipo de estrutura. Nos algoritmos, devemos observar os seguintes aspectos:

* Não podemos testar variáveis reais
* Só podemos testar igualdade
* Não podemos testar condições compostas

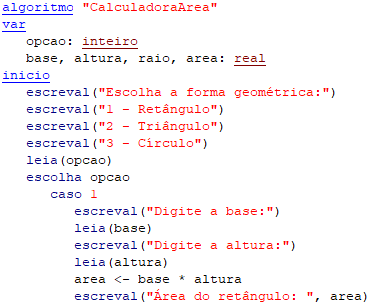
**Exemplos**

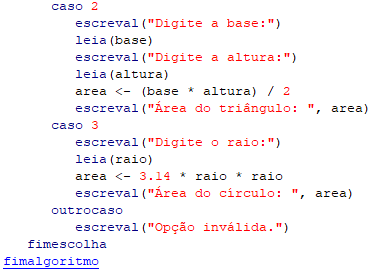
1) Algoritmo para ler um número inteiro e mostrar o dia da semana correspondente



# 

2) Algoritmo para calcular valores a partir dos dados de figuras geométricas



-+.

**Atividades Propostas**

1) Um banco cobra tarifas diferentes para tipos de conta:

1 = Conta Corrente: tarifa fixa de 12,50  
2 = Conta Poupança: tarifa de 2% sobre o saldo  
3 = Conta Salário: tarifa de 1% sobre o saldo

Crie um algoritmo que leia o tipo de conta e o saldo, e exiba a tarifa a ser cobrada.

2) Criar um algoritmo para uma empresa que organiza corridas de rua. Pede-se que sejam lidos o cpf, nome, idade do atleta e o código da corrida na qual ele deseja se inscrever, conforme tabela a seguir:

|  |  |  |
| --- | --- | --- |
| Codigo | Corrida | Valor inscrição |
| C5 | 5 km | 65,00 |
| C10 | 10 km | 78,00 |
| C15 | 15 km | 94,00 |
| C21 | 21 km | 115,00 |

Pede-se que o algoritmo calcule e mostre quanto o atleta pagará pela inscrição na prova escolhida, junto com seu CPF e nome. Obs: Atletas com 60 anos ou mais tem 50% de desconto.

3) Crie um algoritmo que simule um conversor de moedas. O programa deve exibir um menu com as seguintes opções:

1 - Converter de **Real para Dólar** 2 - Converter de **Real para Euro** 3 - Converter de **Real para Libra Esterlina** 4 - Converter de **Real para Peso Argentino**

O usuário deve informar a opção desejada e o valor em reais.

Considere as seguintes taxas de conversão (valores fictícios para o exercício):

* 1 Dólar = R$ 5,00
* 1 Euro = R$ 5,50
* 1 Libra = R$ 6,20
* 1 Peso Argentino = R$ 0,04

4) Criar um algoritmo que leia o valor de uma carga a ser entregue por uma transportadora, a forma de transporte (veja tabela a seguir) e peso desta carga.

FORMA VALOR DA TAXA

N (caminhão normal) 0,35 x peso x valor da carga

T (trem) 0,28 x peso x valor da carga

E (caminhão expresso) 0,55 x peso x valor da carga

Pede-se que o algoritmo calcule e escreva o valor da taxa de entrega.

5) Criar um algoritmo que leia um número de bytes e uma opção para converter este valor em outras unidades, conforme tabela a seguir:

OPÇÃO CONVERTER PARA CÁLCULO

1 KB (kilobytes) dividir por 1024

2 MB (megabytes) dividir por 1024 ao quadrado

3 GB(gigabytes) dividir por 1024 elevado ao cubo

4 TB(terabytes) dividir por 1024 elevado à quarta potência

6) Criar um algoritmo que leia o CPF de um cliente da operador MegaNet, código do plano de internet que ele deseja contratar a forma de pagamento, conforme tabela a seguir:

A - 100 MB - 50,00

B - 200 MB - 79,00

C - 400 MB - 99,00

D - 500 MB - 139,00

As formas de pagamento disponíveis são:

1 - Pix

2 - Débito em Conta

3 - Cartão de Crédito

Clientes que optarem pelo pagamento via PIX tem 5% de desconto.

Pede-se que o algoritmo calcule e mostre quanto o cliente pagará pela mensalidade do plano contratado.

7) Criar um algoritmo que leia a opção de cálculo de área de uma das figuras geométricas (retângulo, triângulo, círculo, trapézio ou hexágono). A partir da opção escolhida o algoritmo deve ler também as medidas da figura e efetuar o cálculo do resultado). As opções são:

R = Retângulo (base × altura)

T = Triângulo (base × altura ÷ 2)

C = Círculo (π × raio²)

Z = Trapézio ((base maior + base menor) × altura ÷ 2)

H = Hexágono Regular ((3 × √3 × lado²) ÷ 2)

**ESTRUTURAS DE REPETIÇÃO**

Até então, todos os algoritmos criados possuem uma característica em comum: não importam os dados recebidos, os cálculos efetuados ou resultados exibidos, todos eram compostos por conjuntos de comandos escritos para serem executados apenas uma vez. Porém, existem várias situações onde essa limitação faça com que os algoritmos sejam desenvolvidos de forma complexa, ou mesmo impede o desenvolvimento dos mesmos. Vamos imaginar, por exemplo, um algoritmo para apurar os resultados de uma pesquisa envolvendo 100 pessoas. Podemos observar algumas ações são executadas várias vezes, mas seguindo o mesmo roteiro (por exemplo, identificação do entrevistado, coleta de respostas, etc). Assim, se estivéssemos criando um algoritmo para tal finalidade, deveríamos usar um recurso que permitisse a múltipla execução destas ações. Para isso, utilizaremos as estruturas de repetição (loops), que são simplesmente blocos de comandos executados mais de uma vez em um algoritmo.

Ao criarmos uma estrutura de repetição, devemos pensar em como o processo será interrompido. Afinal de contas, toda repetição deve parar, caso contrário, estaremos diante de uma situação conhecida como loop infinito. Para interromper uma repetição, existem duas possibilidades: especificamos quantas vezes ela deve acontecer (repetição determinada) ou determinamos uma condição para sua interrupção (repetição indeterminada). Não vamos discutir aqui qual dos métodos é o melhor, até porque eles são aplicados em diferentes situações.

**1) ESTRUTURA PARA**

A primeira estrutura de repetição que iremos utilizar é a estrutura PARA, que é utilizada apenas para estruturas determinadas. Sua sintaxe é a seguinte:

**Para VARIÁVEL <- VALOR INICIAL Ate VALOR FINAL Passo INTERVALO Faça**

**BLOCO DE COMANDOS**

**Fim Para**

VARIÁVEL: será utilizada como contador

VALOR INICIAL: primeiro valor do contador

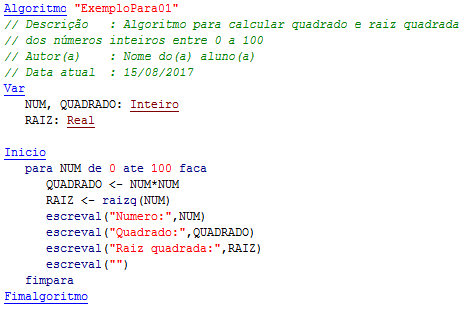
VALOR FINAL: último valor do contador

INTERVALO: valor do incremento/decremento do contador

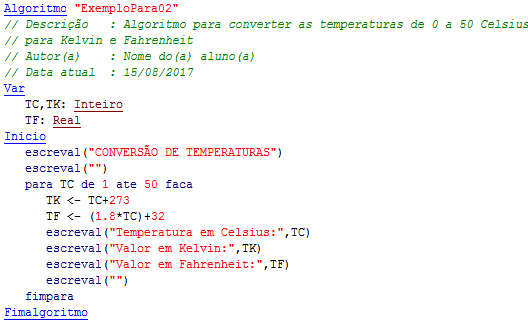
Ao encontrar a estrutura PARA, o algoritmo atribui à variável usada como contador o valor inicial, descrito pelo programador. Em seguida, executa o bloco de comandos descrito para a estrutura. Após finalizar a execução, ele atribui à variável contadora o próximo valor da sequência (observando o incremento/decremento) e executa novamente o bloco de comandos. Esse processo será repetido até que a variável contadora receba o valor definido como final. A partir daí, o bloco de comandos da repetição é executado pela última vez. Após finalizada, o algoritmo continua executando os comandos descritos após a mesma.

**Exemplos**

Exemplo 01



Exemplo 02



Exemplo 03

