

INTRODUÇÃO A LÓGICA DE PROGRAMAÇÃO USANDO O VISUALG – CONCEITOS BÁSICOS E FLUXOGRAMA

João Rubens Marchete Filho

Luis Fernando Alves da Silva

Pedro Ramires da Silva Amalfi Costa

CURSO TÉCNICO DE INFORMÁTICA E INFORMÁTICA PARA INTERNET

INTRODUÇÃO A LÓGICA DE PROGRAMAÇÃO USANDO O VISUALG – CONCEITOS BÁSICOS E FLUXOGRAMA

João Rubens Marchete Filho

Luis Fernando Alves da Silva

Pedro Ramires da Silva Amalfi Costa

XXXX

Filho, João Rubens Marchete.

Introdução a Lógica de Programação usando o VisuAlg – Conceitos Básicos e Fluxograma / João Rubens Marchete Filho, Luís Fernando Alves da Silva, Pedro Ramires da Silva Amalfi Costa. - - Mococa, 2014. 46f. :il.

Apostila (Informática e Informática para Internet) – Centro Estadual de Educação e Tecnologia Paula Souza.

1. Lógica de Programação. 2. Fluxograma. 3. Operadores Aritméticos e variáveis. 4. Pseudocódigo
I. Título.

CPSCETEC / SP

CDU - xxx:yyy

Sumário

1. Introdução a Lógica de Programação	6
1.1. Definições.....	6
1.1.1. Lógica.....	6
1.1.2. Algoritmo	7
1.1.3. Programa.....	9
1.2. Desenvolvimento algoritmos	10
1.2.1. Descrição Narrativa	10
1.2.2. Fluxograma.....	11
1.2.3. Pseudocódigo.....	12
Exercícios.....	13
2. operadores aritméticos e variáveis	15
EXERCÍCIOS	20
2.1. PRIORIDADE nas operações aritméticas	22
Exercícios.....	23
2.2 Operadores relacionais	24
2.2.1 Atribuição	24
2.2.2 Igualdade.....	24
2.2.3 Diferença	24
2.2.4 Maior	25
2.2.5 Menor	25
2.2.6 Maior igual e Menor igual	25
2.3. Variáveis.....	26
Numéricas.....	26
Caracteres	27
Lógica	27

Respostas	28
3. Pseudocódigo	31
3.1 Formato	31
3.2 Comando iniciais	31
3.2.1 Comando Algoritmo	31
3.2.2 Comando Var	31
3.2.3 Comando Inicio.....	32
3.2.4 Comando Fim	32
3.2.5 Comandos de Entrada e Saida.....	32
3.2.6 Comando ESCREVA.....	32
3.2.7 Comando LEIA	32
Exemplo.....	33
3.3 Dicas para nomear variáveis	34
Exercícios.....	35
Respostas	37
Referência bibliografica	50

1. INTRODUÇÃO A LÓGICA DE PROGRAMAÇÃO

Uma boa forma de começar a entender a lógica de programação é primeiro entender conceitos básicos e definições de o que vem a ser essa tal lógica de programação, onde a usamos e como ela me é útil.

1.1.DEFINIÇÕES

Vamos agora ver alguma definições para começarmos a entender este novo mundo de possibilidades.

1.1.1. Lógica

Lógica, termo grego *logiké*, relacionado com a razão, logos, palavras ou discurso, que significa a ciência do raciocínio.

A lógica é um instrumento do pensar, conseqüentemente, é uma busca pela verdade, pela razão, pelo pensar corretamente, e para isso usa-se regras e fórmulas que comprovam ou alcançam uma determinada teoria, meta ou lógica.

A lógica é a principal ferramenta para a evolução humana pois foi quando o homem começou a assimilar eventos que aconteciam ao seu cotidiano com ações que aconteciam antes que se obtivesse o conhecimento para se criar aquele evento, um exemplo disso é a descoberta da plantação, os homens pré-históricos relacionaram que o lugar onde se jogavam as sementes eram os mesmos que meses depois eles encontravam seus alimentos, então se eles plantassem essas sementes próximo de suas casas e cuidassem dela, lá nasceria outras plantas que lhe dariam alimento e não teriam mais de procura-las.

1.1.2. Algoritmo

Algoritmo é uma sequência de instruções que quando seguidas em uma determinada ordem resulta no mesmo resultado ou na mesma saída, pensando por este ponto podemos afirmar que tudo que fazemos no nosso cotidiano tem a ver com o algoritmo e consequentemente é uma parte da lógica.

Imaginem o seguinte problema, queremos fazer uma pizza de presunto e mussarela, para resolver este problema precisamos primeiro definir quais serão os passos para a resolução deste problema:

- a) Preparar a massa
- b) Passar molho de tomate
- c) Adicionar o presunto
- d) Adicionar a mussarela
- e) Adicionar a cebola
- f) Adicionar o orégano
- g) Adicionar a azeitona
- h) Deixar o fogão aceso por 15 minutos
- i) Colocar a pizza no forno
- j) Esperar ela ficar no ponto ideal para servir
- k) Remover do forno
- l) Cortar do jeito desejado
- m) Se servir

Pronto, definimos o passo-a-passo de desenvolvimento de uma pizza, agora podemos observar que se alterarmos a ordem descritas o resultado pode não ser algo que lembre uma pizza. Imagine se primeiro adicionarmos ao forno apenas a mussarela, presunto e molho e depois de 15 minutos adicionar a massa, tirar do forno e colocar o orégano, se servir e por aí vai até completar os demais passos, isso não vai ser uma pizza e sim uma gororoba. Então podemos afirmar que um passo-a-passo é um algoritmo, sendo ele ou não um código de computador.

Outra coisa que podemos reparar nesta sequência é que podemos dividir nossos passos em 3 grupos: entrada, processamento e saída.

Entrada: É definido como entradas em algoritmo toda informação que sem ela não se consegue obter o resultado desejado, no exemplo da pizza, as nossas entradas seriam os ingredientes pois sem eles não é possível concretizar a pizza.

- a) Preparar a massa
- b) Passar molho de tomate
- c) Adicionar o presunto
- d) Adicionar a mussarela
- e) Adicionar a cebola
- f) Adicionar o orégano
- g) Adicionar a azeitona

Processamento: É definido como processamento em algoritmo todas as vezes que utilizamos os dados de entrada para modificar ou criar novos valores, tornando possível a criação do resultado final desejado. No exemplo da pizza, só ter os ingredientes (nossas entradas) não faz com que se transforme em pizza, então todos os passos que foram utilizados para misturar ou juntar os ingredientes e colocá-los no forno são nossos processamentos.

- h) Deixar o fogão aceso por 15 minutos
- i) Colocar a pizza no forno
- j) Esperar ela ficar no ponto ideal para servir

Saída: É definido como saída em algoritmo sempre que obtivermos o resultado final. No exemplo da pizza é quando ela já está pronta para cortarmos e comermos.

- k) Remover do forno
- l) Cortar do jeito desejado
- m) Se servir

1.1.3. Programa

Uma definição para programa é uma coleção de instruções que descrevem uma tarefa a ser realizada pelo computador.

Alguns programas mais utilizados por nós é a calculadora, o documento de texto, a planilha eletrônica, o navegador, todos eles são programas distintos, com propósitos distintos mas com instruções específicas para cada caso. Não vamos usar a calculadora para navegar na internet, muito menos a planilha eletrônica para escrever um texto.

O interessante nos programas é que não existe um jeito de fazê-los, existem várias formas e cada uma delas tem suas peculiaridades, não tem como afirmar que um jeito é melhor do que outro, mas posso afirmar que cada um encontra o jeito que mais gosta de usar.

Para se criar os programas usamos as linguagens de programação, elas funcionam como um intermediário entre o usuário e a máquina. Imaginem como se o usuário fosse um arquiteto brasileiro construindo uma casa e que só fala português e estivesse tentando falar para um funcionário o que ele tem de fazer para a casa ficar pronta só que seu funcionário só fala alemão, então o tradutor que pega tudo o que o arquiteto quer em português e traduz para o funcionário em alemão é o que chamamos de linguagem de programação.

Existem diversas linguagens de programação no mundo, vou colocar em seguida algumas delas, umas novas outras não tão recentes assim:

- C ++
- C #
- Java
- Javascript
- Python
- Ruby
- Cobol
- Pascal
- Delphi

1.2. DESENVOLVIMENTO ALGORITMOS

OK, imagino que neste momento está louco de vontade de colocar a mão na massa e desenvolver diversos algoritmos mas antes disso preciso explicar rapidamente a você que existem 3 formas de se desenvolver um algoritmo que são: Descrição Narrativa, Fluxograma e Pseudocódigo.

1.2.1. Descrição Narrativa

São fatos narrado conforme acontecem, lembrando o exemplo da pizza, uma descrição narrativa dela seria:

Primeiro prepare a massa, passe molho de tomate por cima da massa já pré-assada de maneira uniforme, em seguida lance por cima do molho o presunto cortado em fatias ou moído até se sentir por satisfeito com a quantidade, faça o mesmo com a mussarela em fatias ou moída, não se esqueça de adicionar a cebola, o orégano e as azeitonas (verdes ou pretas) a seu gosto. Pré-aqueça o fogão por 15 minutos em fogo alto, abaixe a temperatura para médio ou baixo e coloque a pizza para assar, neste momento fique observando esporadicamente o forno até estar no ponto ideal para ser servida, com cuidado para não queimar a mão remova-a do forno contando-a como desejar e bom apetite, está pronto para ser servida.

Um dos problemas em se utilizar a forma de descrição narrativa é que se escreve muito para se falar pouca coisa e por se tratar de um texto, está sujeito a ser impreciso, esquecer alguma etapa do processo e consequentemente não dar certo. Outro problema é no fato de obrigar o usuário a ler o texto faz com que demore mais para entender as etapas do processo e o seu relacionamento entre as etapas, enquanto nos outros modelos esta informação é passada de uma maneira mais visual e consequentemente mais rápida.

1.2.2. Fluxograma

Usa símbolos universais que nos ajudarão a entender o que o algoritmo que dizer. O bom deles é que por ser um padrão universal, fluxogramas escritos por programadores brasileiros poder ser lidos e entendidos por programadores alemães, franceses, italianos, americanos e todos vão conseguir entender o que está acontecendo com o programa e qual o resultado final do mesmo.

Normalmente, antes de escrevermos um programa em uma das diversas linguagens de programação existentes, usamos o fluxograma para esboçar o programa para em seguida transformarmos ele pronto em um código que pode ser rodado em uma máquina.

Um ponto fraco do fluxograma é que ele se torna muito complicado a partir do momento em que seu programa vai crescendo mas ainda é aconselhado a utilização do mesmo como ferramenta na criação de seus programas.

Tente entender o fluxograma a baixo sem eu explicar nada. Não se preocupe caso não entenda tudo pois será explicado mais detalhadamente nos próximos capítulos desta apostila. Use o espaço em branco ao lado desta figura para escrever o que acha que cada parte faz.

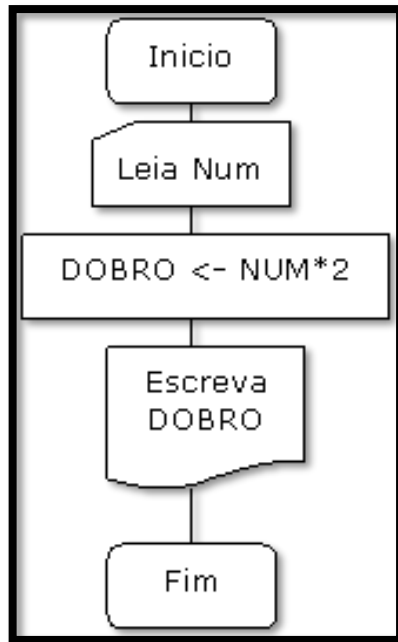


Figura 1 – exemplo de Fluxograma

1.2.3. Pseudocódigo

É uma linguagem intermediária entre a nossa linguagem e a linguagem de programação. Nesta apostila vai ser utilizado o pseudocódigo VisuAlg pois usa o português para escrever nossos programas, pode ser executada em um computador, ao contrário de outras linguagens ela é de graça, é uma forma simples de ser representada, utilizada, manipulada e entendida.

Ela é uma boa forma de se aprender a programar, eu sempre falo em minhas aulas que ela é a estrutura principal de uma casa, pois se aprendida corretamente, vai auxiliar a aprender várias outras linguagens sem muita dificuldade, ao contrário do que acontece quando não se aprende-a corretamente ou sequer sabe sobre sua existência.

Do mesmo modo no fluxograma, vou adicionar um exemplo de um código escrito usando o pseudocódigo VisuAlg, não se preocupe caso não entenda tudo ou nada do exemplo pois vai ser explicado melhor no decorrer desta apostila.

```
Programa area_quadrado
Início
Variaveis:
    lado: real  area: real
    Escreva "Introduza o lado do quadrado"
    Leia lado
    Area<- lado * lado
    Escreva "A área do quadrado é", area
Fim
```

Figura 2 - exemplo de Pseudocódigo

EXERCÍCIOS

1. Descreva quais os passos e separe-os em entrada, processamento e saída para:
 - Trocar o pneu de um carro;
 - Tomar banho;
 - Fazer um bolo;
 - Somar dois números;
2. No subcapítulo 1.1.3 foi colocado uma lista com as linguagens de programação, pesquise sobre elas e informe o ano de surgimento de cada uma delas.
3. Faça uma descrição narrativa de escovar seus dentes, atravessar uma rua e acessar um site.

4. Tente descrever passo-a-passo o que entendeu que está acontecendo no exemplo passado no fluxograma, separando-o em entrada, processamento e saídas.
5. Faça o mesmo do exercício 4 mas agora para o exemplo passado no fluxograma.

2. OPERADORES ARITMÉTICOS E VARIÁVEIS

Vamos começar agora a ver como escrever os nossos algoritmos em VisuAlg e ao mesmo tempo como trabalhamos eles no fluxograma ok?

Abaixo tem a operação e o símbolo usado em cálculos no VisuAlg e fluxogramas, repare que na multiplicação é usado o asterisco e na divisão uma barra.

OPERAÇÃO	SÍMBOLO
Adição	+
Subtração	-
Multiplicação	*
Divisão	/

Outros operadores aritméticos não são usados como símbolos mas sim como palavras como:

Raiz Quadrada: SQRT (n)

Quando formos realizar uma equação matemática onde possui uma raiz quadrada devemos usar a palavra SQRT e na frente dela o número desejado como no exemplo SQRT 9 estamos calculando a raiz quadrada de 9 cujo resultado é 3.

Potência de 2: SQR (n)

Quando formos realizar uma equação matemática onde precisamos elevar ao quadrado um determinado número, devemos usar a palavra SQR e na frente o número desejado como no exemplo SQR 4 estamos elevando 4 ao quadrado cujo resultado é 16.

Divisão Inteira: (N1) DIV (N2)

Quando formos realizar uma equação matemática onde precisamos dividir um número pelo outro mas só queremos a parte inteira daquela equação usaremos da seguinte forma: primeiro adicionamos o número que queremos dividir seguido da palavra DIV e em seguida pelo número de vezes que vamos dividir como no exemplo 9 DIV 2 onde resultará em 4.

Resto da Divisão: (N1) MOD (N2)

Quando formos realizar uma equação matemática onde precisamos dividir um número pelo outro mas só queremos o resto daquela equação usaremos da seguinte forma: primeiro adicionamos o número que queremos dividir seguido da palavra MOD e em seguida pelo número de vezes que vamos dividir como no exemplo 9 MOD 2 onde resultará em 1.

Imagine o seguinte problema: Calcular a soma final das compras realizadas em um dia. Sabendo que foram comprados quatro produtos temos: P1, P2, P3 e P4 e sabemos que conseguimos calcular a soma total gasta somando todos os produtos da seguinte forma:

$$\text{Soma Final} = P1 + P2 + P3 + P4$$

Para montar o algoritmo proposto, faremos 3 perguntas:

A) Quais são os dados de entrada?

R: Os dados de entrada são P1, P2, P3, P4

B) Qual será o processamento utilizado?

R: O processamento é a soma de todos os produtos comprados.

C) Quais serão os dados de saída?

R: Será o resultado encontrado na soma.

Feito isso, temos agora de criar os fluxograma do exercício proposto, mas para isso, temos de saber o que cada símbolo representa e como são eles:





Símbolo	Função
 TERMINAL	Indica o INÍCIO ou FIM de um processamento Exemplo: Início do algoritmo
 PROCESSAMENTO	Processamento em geral Exemplo: Calculo de dois números
 ENTRADA DE DADO MANUAL	Indica entrada de dados através do Teclado Exemplo: Digite a nota da prova 1
 EXIBIR	Mostra informações ou resultados Exemplo: Mostre o resultado do calculo

Figura 3 - principais simbolos de fluxograma

Como pode ser visto todo início de fluxograma e termino do mesmo é usado o primeiro símbolo só que dentro do primeiro tem escrito início e no ultimo escrito fim.

Vamos ver como seria um fluxograma do exemplo proposto e vamos entender melhor como cada símbolo funciona.

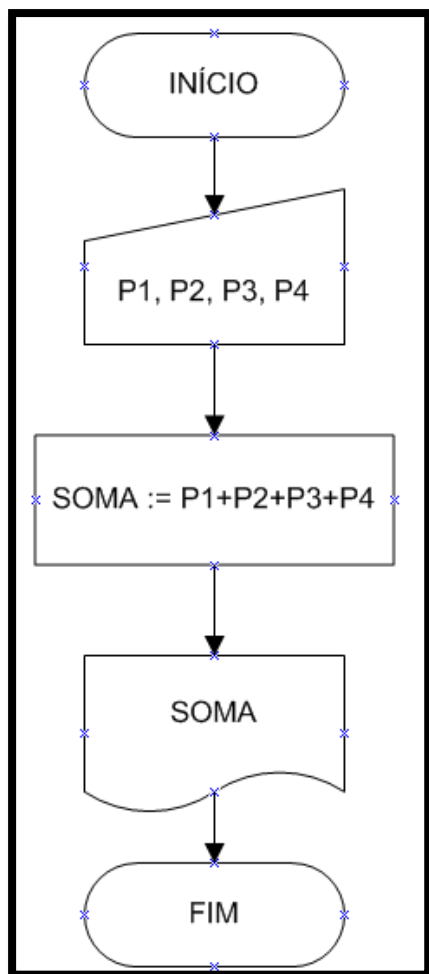


Figura 4 - explicando um fluxograma

No primeiro balão iniciamos o nosso fluxograma.

No segundo balão entramos com os dados necessarios para encontrar o resultado desejado.

No terceiro balão podemos observar que foi somado os valores dos produtos e este resultado foi armazenado na variável SOMA.

É mostrado o resultado armazenado na variavel SOMA para o usuário.

E por ultimo, encerra-se o fluxograma.

Alguns outros exemplos de fluxogramas vão ser colocados a baixo, aproveite o lado da figura para adicionar o texto tentando explicar o que acontece em cada símbolo conforme feito no exemplo a cima.

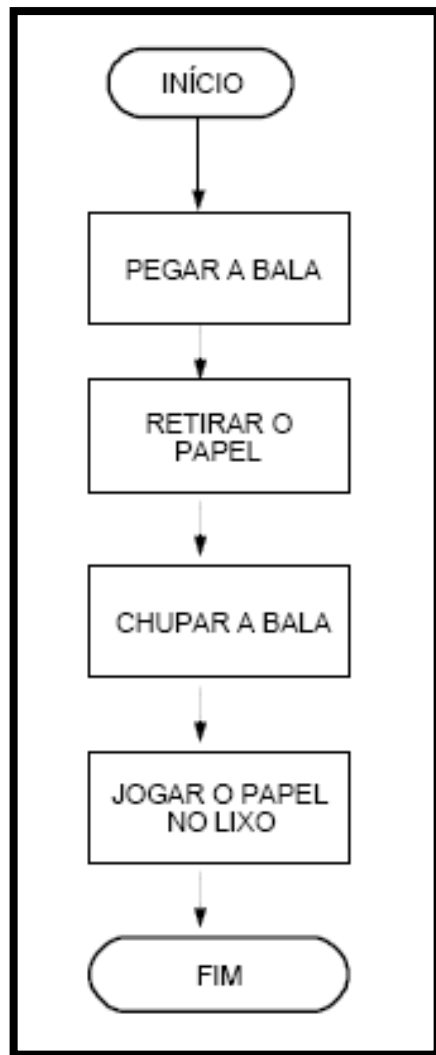


Figura 5 – exercício 1 de fluxograma

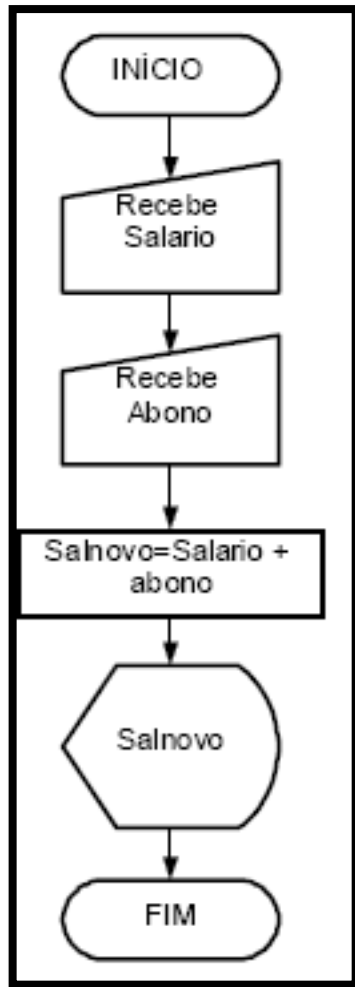


Figura 6 - exercício 2 de fluxograma

EXERCÍCIOS

1 - Desenvolva um fluxograma que:

- Leia 4 números
- Calcule a média
- Mostre o resultado

2 - Desenvolva um fluxograma que:

- Leia os lados de um quadrado
- Calcule a área, sabendo que a fórmula $\text{Área} := \text{Lado} * \text{Lado}$
- Mostre o resultado

3 - Desenvolva um fluxograma que:

- Leia 5 idades
- Calcule a média das idades
- Mostre o resultado.

2.1. PRIORIDADE NAS OPERAÇÕES ARITMÉTICAS

Uma coisa importante quando se vai realizar operações aritméticas é que dependendo da operação, temos de calcular uma parte da fórmula para depois seguir o cálculo para não resultar em um valor errado e para isso usamos algumas regras pra nos ajudar:

1º () Parênteses – é calculado tudo o que se está em parênteses para depois calcular o resto.

2º Exponenciação e Raiz – depois de ter acabado os parênteses, são calculados as raízes e as exponenciações, o que vier primeiro.

3º Multiplicação e Divisão – em seguida são calculados as multiplicações e divisões, o que vier primeiro.

4º Adição e Subtração – e por último é calculado as adições e subtrações, o que vier primeiro.

Exemplo:

$$3 * (1-2) + 4 * 2$$

$$3 * -1 + 4 * 2$$

$$-3 + 4 * 2$$

$$-3 + 8$$

$$5$$

$$5 + 9 + 7 + 8 / 4$$

$$5 + 9 + 7 + 2$$

$$14 + 7 + 2$$

$$21 + 2$$

$$23$$

$$1 - 4 * 3 / 6 - \text{SQR}(3)$$

$$1 - 4 * 3 / 6 - 9$$

$$1 - 12 / 6 - 9$$

$$1 - 2 - 9$$

$$- 1 - 9$$

$$- 10$$

EXERCÍCIOS

4 - Resolva as equações matemáticas respeitando as prioridades aritméticas:

a. $2 * 7 \text{ MOD } 3 - 4$

b. $5 \text{ DIV } 2 - (10 * 2) / 4 + \text{SQR}(2)$

c. $10 / 2 - 2 * (50 + 4) - \text{SQRT}(9)$

2.2 OPERADORES RELACIONAIS

Uma coisa que temos de ver antes de começar a criar nossos programas no VisuAlg é como usar os operadores relacionais.

2.2.1 Atribuição

O primeiro que vimos e já estamos usando é o de Atribuição. O Operador de atribuição é usado para informar ou melhor atribuir algum valor a alguma variável, como podemos ver nos exemplos anteriores onde uma variável recebia o valor de uma equação matemática como no exemplo a baixo:

Média:= (n1+n2+n3+n4) /4

Falamos que a variável média recebe o valor de n1+n2+n3+n4 tudo dividido por 4. Podemos usar como símbolo de atribuição este (:= dois pontos igual) ou este (<- sinal de menor e traço formando uma seta).

2.2.2 Igualdade

Outro operador relacional é o igualdade, representado pelo símbolo de igual (=), é usado quando queremos que o programe informe se é verdadeiro ou não uma determinada condição. Exemplo:

2+4 = 6 (essa é uma expressão verdadeira já que 2 + 4 é igual a 6)

3+ 5 = 15 (essa é uma expressão falsa já que 3 + 5 é 8 e não 15)

2.2.3 Diferença

A diferença é semelhante ao igualdade, só que o seu oposto. É representado pelo sinal de maior e menor (<>).

2+4 = 6 (essa é uma expressão falsa já que 2 + 4 não é diferente a 6)

3+ 5 = 15 (essa é uma expressão verdadeira já que 3 + 5 é 8 e não 15)

2.2.4 Maior

Usado quando criamos condições numéricas para nossos programas onde será retornado verdadeiro ou falso. Seu símbolo é o maior ($>$).

$3 > 1$ (verdadeiro, já que 3 é maior que o número 1)

$2 > 7$ (falso, uma vez que 2 não é maior que o número 7)

$5 > 5$ (falso já que 5 não é maior que 5)

2.2.5 Menor

Mesma coisa que o maior só que o oposto. Seu símbolo é o menor ($<$).

$3 < 1$ (falso, já que 3 é maior que o número 1)

$2 < 7$ (verdadeiro, uma vez que 2 não é maior que o número 7)

$5 < 5$ (falso já que 5 não é menor que 5)

2.2.6 Maior igual e Menor igual

Maior igual e menor igual tem as mesmas finalidades do maior e do menor visto anteriormente mas caso os números sejam iguais eles serão verdadeiros.

$3 \geq 1$ (verdadeiro já que mesmo 3 não sendo igual a 1, ainda é maior)

$2 \geq 7$ (falso, uma vez que 2 não é nem maior nem igual ao número 7)

$5 \geq 5$ (verdadeiro já que 5 não é menor que 5 mas é igual a 5)

$3 \leq 1$ (falso, já que 3 não é igual nem menor a 1)

$2 \leq 7$ (verdadeiro, uma vez que mesmo 2 não sendo igual a 7, 2 também não é maior que o número 7)

5 <= 5 (verdadeiro já que 5 não é menor que 5 mas é igual a 5)

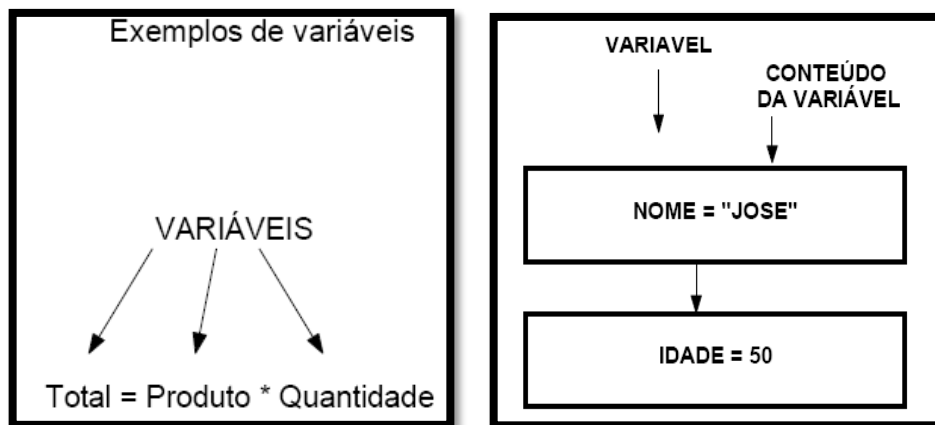
2.3. VARIÁVEIS

Outra coisa que já usamos em exercícios anteriores mas que agora vai ser explicado melhor é o uso das variáveis, nossos programas não funcionarão sem as variáveis para armazenarmos as informações que recebem do usuário ou por meio de processamentos realizados na execução de nossos programas.

Cada variável corresponde a uma posição de memória, cujo conteúdo pode ser alterado ao longo do tempo durante a execução de um programa.

Embora uma variável possa assumir diferentes valores, ela só pode armazenar um valor a cada instante.

Como pode ser visto nas imagens a baixo, já usávamos as variáveis em nossos programas mas agora vamos entender um pouco mais sobre elas e sobre seus tipos.



Existem 3 tipos de variáveis que mais usamos no VisuAlg:

Numéricas: Específicas para armazenamento de números, que posteriormente poderão ser utilizados para cálculos. Podem ser ainda classificadas como Inteiras ou Reais.

As variáveis do tipo inteiro são para armazenamento de números inteiros. Exemplo: Idade:= 15;

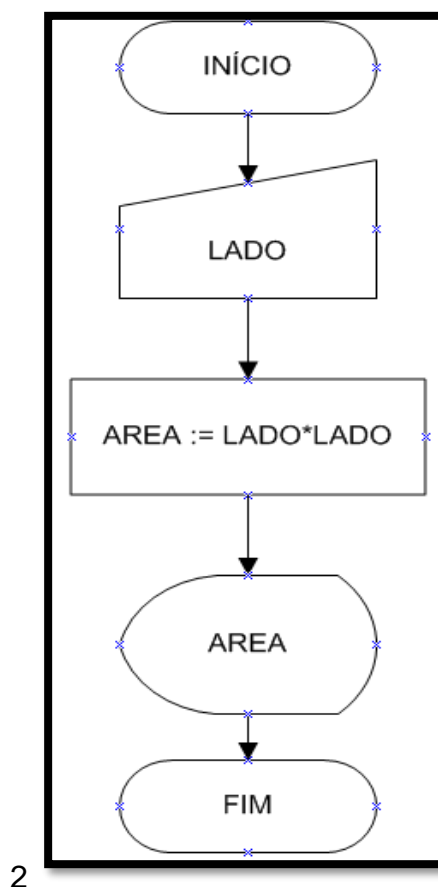
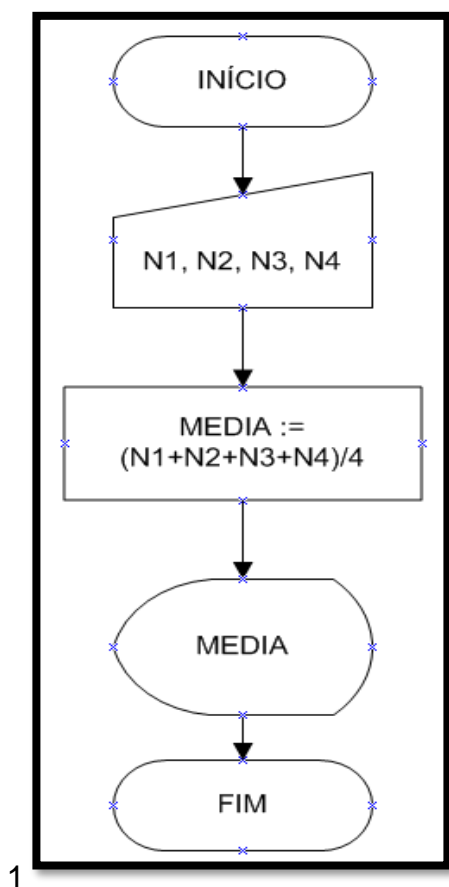
As variáveis Reais são para o armazenamento de números que possuam casas decimais. Exemplo: Produto:=20.59 (conotação americana, usa o ponto para separar o valor inteiro do fracionado);

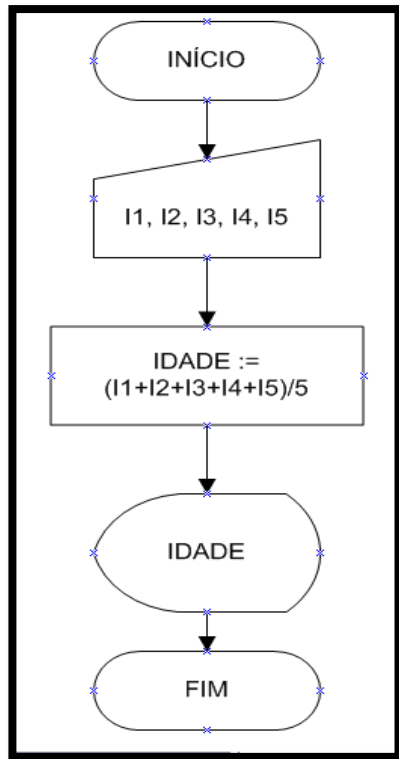
Caracteres: Específicas para armazenamento de conjunto de caracteres que não contenham números (literais). Exemplo: nome:= “Pedro Ramires”;

Como pode ser observado quando vamos atribuir um valor a uma variável numérica, não precisamos de usar aspas em volta de seu valor agora o mesmo já não acontece quando é uma variável do tipo caracter pois temos a obrigação de informar aonde começa e onde termina o valor que será atribuído aquela variável usando as aspas para fazer esta indicação.

Lógica: Armazena apenas dados lógicos como Verdadeiro ou Falso.

RESPOSTAS





3

4.A) $2 * 7 \text{ MOD } 3 - 4$

$14 \text{ MOD } 3 - 4$

$2 - 4$

-2

4.B) $5 \text{ DIV } 2 - (10 * 2) / 4 + \text{SQR}(2)$

$5 \text{ DIV } 2 - 20 / 4 + \text{SQR}(2)$

$5 \text{ DIV } 2 - 20 / 4 + 4$

$2 - 20 / 4 + 4$

$2 - 5 + 4$

$-3 + 4$

1

4.C) $10 / 2 - 2 * (50 + 4) - \text{SQRT}(9)$

$$10 / 2 - 2 * 54 - \mathbf{SQRT(9)}$$

$$\mathbf{10 / 2 - 2 * 54 - 3}$$

$$5 - \mathbf{2 * 54} - 3$$

$$\mathbf{5 - 108} - 3$$

$$\mathbf{-103 - 3}$$

$$\mathbf{-106}$$

3. PSEUDOCÓDIGO

Forma genérica de escrever um algoritmo, utilizando uma linguagem simples (na língua de quem o escreve, de forma a ser entendida por qualquer pessoa) sem necessidade de conhecer a sintaxe de nenhuma linguagem de programação.

Não pode ser executado num sistema real (computador), pois deixaria de ser “pseudo”.

3.1 FORMATO

```
algoritmo "semnome"  
var  
//Declaração das variáveis  
Início  
// Seção de Comandos  
Fim
```

3.2 COMANDO INICIAIS

- Comando Algoritmo
- Comando Var
- Comando Início
- Comando Fim

3.2.1 Comando Algoritmo

Só serve para indicar o nome do algoritmo

Exemplo:

Algoritmo TESTE

3.2.2 Comando Var

Indica onde as variáveis serão declaradas.

Passa a ser opcional em alguns algoritmos que apenas imprimem instruções.

Exemplo:

Var

n1, n2 : inteiro

n3, n4 : real

nome, cep : literal

3.2.3 Comando Início

Indica onde começarão as instruções. É obrigatório.

Exemplo:

Início

Escreva ("Isto é um algoritmo")

3.2.4 Comando Fim

Serve apenas para indicar que o algoritmo está terminado, é obrigatório.

3.2.5 Comandos de Entrada e Saída

Comando de Entrada: LEIA

Comando de Saída: ESCREVA, ESCREVAL

3.2.6 Comando ESCREVA

Imprime mensagens e conteúdos de variáveis

Mensagens devem ser colocadas entre aspas

Conteúdos de variáveis devem ser mostrados após a inserção de uma vírgula para separar a variável da mensagem a ser exibida

Exemplo: ESCREVA "O seu salário é",novosal

Exemplo 2: ESCREVA "Oi"

Exemplo 3: ESCREVA "Você tem", idade, "anos"

3.2.7 Comando LEIA

Lê a entrada fornecida pelo usuário e armazena este conteúdo numa variável

Exemplo: Leia (nota1)

No exemplo acima, a nota fornecida pelo usuário será armazenada na variável chamada “nota1”

Exemplo

Com base no fluxograma a baixo compare-o com o pseudocódigo para entender suas relações, lembrando sempre de tentar observar as entradas de dados, seus processamentos e suas saídas.

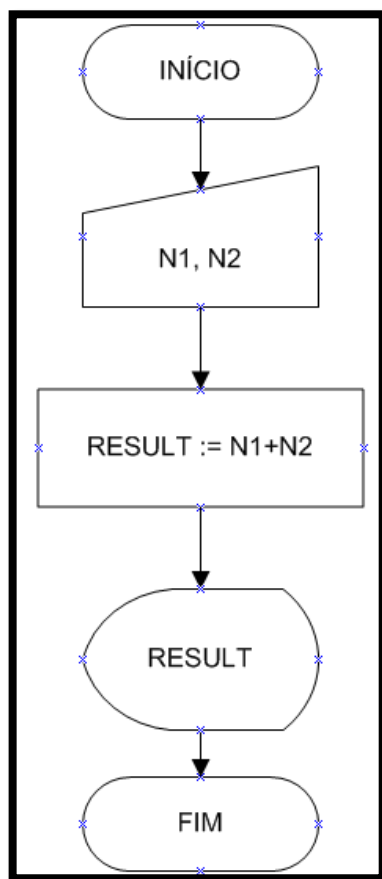


Figura 7 - fluxograma de somar dois numeros

Algoritmo SOMA

Var

n1,n2,result: inteiro

Início

Escreva “Digite o 1º numero”

Leia (n1)

Escreva “Digite o 2º numero”

Leia (n2)

result ← n1+n2

Escreva “A soma dos dois numeros é igual a”, result

Fim

3.3 DICAS PARA NOMEAR VARIÁVEIS

- Não iniciar o nome com algarismos (números). Por exemplo: não usar 1algoritmo. O correto seria algoritmo1 ou algoritmo_1.
- Não utilizar caracteres especiais, como acentos, símbolos (@/?# etc, entre outros.
- Não utilizar nomes iguais para representar variáveis diferentes.
- Não usar o mesmo nome do algoritmo para nomear uma variável e vice-versa.
- Ser sucinto e utilizar nomes coerentes.

EXERCÍCIOS

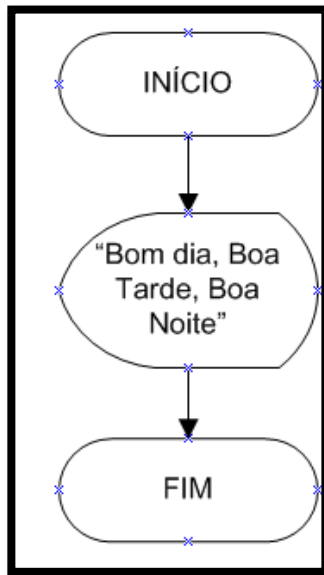
Vamos começar fazendo exercícios simples para criarmos a assimilação de tudo o que aprendemos até o momento usando fluxograma com conceito novo de pseudocódigo, todos os exercícios a baixo devem ser criados os dois (fluxogramas e pseudocódigos).

- a) Faça um programa que envie a mensagem “Bom dia, Boa Tarde, Boa Noite”.
- b) Faça um programa que receba o nome do usuário e envie a mensagem “Olá NOME tudo bem?”
- c) F.U.P (Faça um Programa) que some o valor de 5 contas e mostre o resultado.
- d) FUP que receba 3 idade e informe qual a média das idades.
- e) FUP que receba o valor de um produto e que calcule qual o valor daquele produto se o mesmo receber 10% de desconto.
- f) F.U.P. que calcule a Área de um trapézio sabendo que a formula é $A = ((B+b)*h)/2$
- g) F.U.P. que calcule a Área de um círculo sabendo que a formula é $A = \pi r^2$ onde π vale 3,14
- h) F.U.P. que calcule o salário de um professor sabendo que ele deve informar o valor da hora aula dele, a quantidade de aulas que ele dá em uma semana, sabendo também que um mês tem 4.5 semanas, e que o salário do professor tem de descontar o INSS de 11% para achar o salário líquido daquele professor.

- i) F.U.P. que leia a temperatura em graus Celsius e apresente-a em Fahrenheit, sabendo que a formula para a conversão é $F = (9 \times ^\circ\text{C} + 160) / 5$
- j) FUP que receba dois números e mostre os seguintes resultados para o usuário, sabendo que só pode usar 4 variáveis para executar o exercício:
- A soma do primeiro pelo segundo
 - A subtração do primeiro pelo segundo
 - A divisão do primeiro pelo segundo
 - A multiplicação do primeiro pelo segundo
 - O quadrado do primeiro numero
 - A raiz quadrada do segundo numero
 - O resto da divisão do primeiro pelo segundo
 - A parte inteira da divisão do primeiro pelo segundo.
- K) FUP onde receba o salário do funcionário, os anos de empresa, e a quantidade de filhos sabendo que para cada 2 anos de empresa o funcionário vai receber 10 % de aumento e para cada filho um aumento de R\$100,00. Calcule o novo salário do funcionário e informe-o sobre o novo valor.

RESPOSTAS

a) Fluxograma:



Pseudocódigo:

algoritmo "semnome"

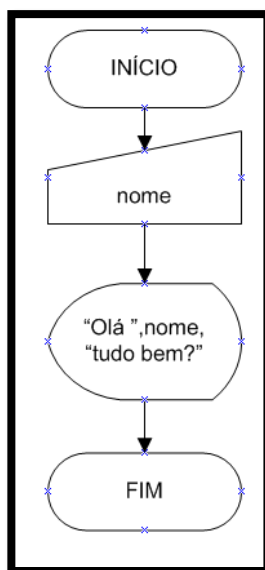
var

inicio

ESCREVAL ("Bom Dia, Boa Tarde, Boa Noite")

fimalgoritmo

b) Fluxograma:



Pseudocódigo:

algoritmo "semnome"

var

nome:caracter

inicio

escreval ("Digite seu Nome:")

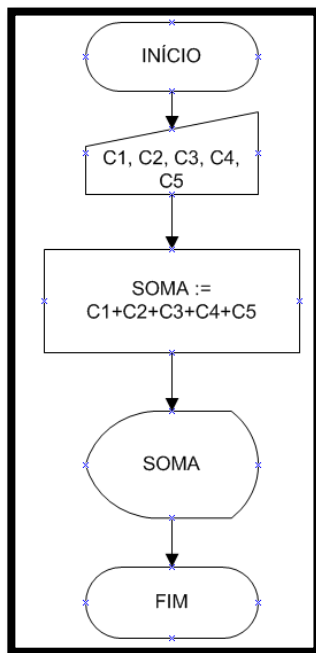
leia(nome)

escreval ("Olá ", nome, " tudo bem?")

fimalgoritmo

c)

Fluxograma:



Pseudocódigo:

algoritmo "semnome"

var

c1, c2, c3, c4, c5, soma:inteiro

inicio

escreval ("Digite o valor da 1ª
conta: ")

leia(c1)

escreval ("Digite o valor da 2ª
conta: ")

leia(c2)

escreval ("Digite o valor da 3ª
conta: ")

leia(c3)

escreval ("Digite o valor da 4ª
conta: ")

leia(c4)

escreval ("Digite o valor da 5ª
conta: ")

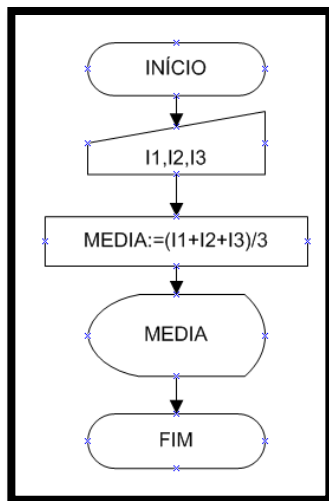
leia(c5)

soma := c1+c2+c3+c4+c5

escreval ("O Total Gasto foi de
R\$",soma)

fimalgoritmo

D) Fluxograma:



Pseudocódigo: algoritmo "semnome"

var

i1, i2, i3:inteiro

media:real

inicio

escreval ("Digite o valor da 1ª idade:")

leia(i1)

escreval ("Digite o valor da 2ª idade:")

leia(i2)

escreval ("Digite o valor da 3ª idade:")

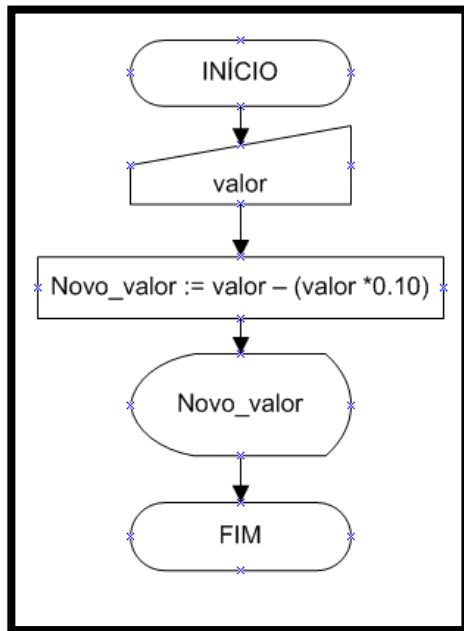
leia(i3)

media := (i1+i2+i3)/3

escreval ("A média das idades é de ",media)

fimalgoritmo

E) Fluxograma:



Pseudocódigo:

algoritmo "semnome"

var

valor, novo_valor:real

inicio

escreval ("Digite o valor do produto:")

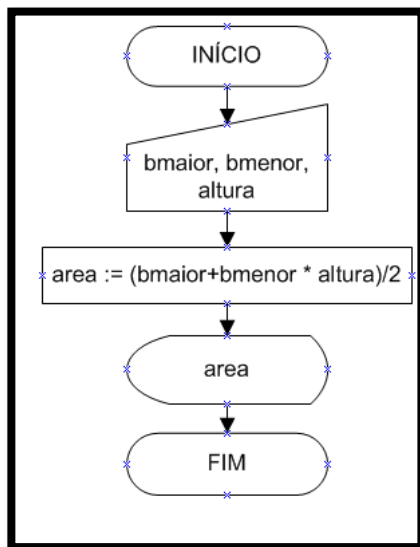
leia(valor)

novo_valor := valor - (valor * 0.10)

escreval ("O valor com o desconto é de R\$",novo_valor)

fimalgoritmo

F) Fluxograma:



Pseudocódigo:

algoritmo "semnome"

var

bmaior, bmenor, altura, area:real

inicio

escreval ("Digite o valor da base maior:")

leia(bmaior)

escreval ("Digite o valor da base menor:")

leia(bmenor)

escreval ("Digite o valor da altura:")

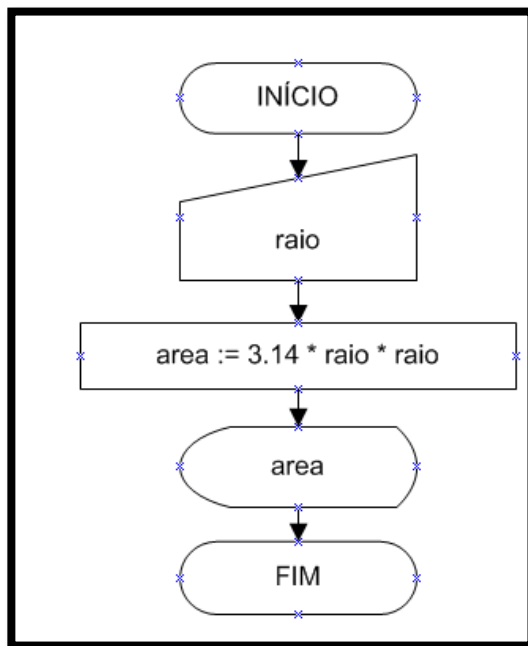
leia(altura)

area := (bmaior+bmenor * altura)/2

escreval ("O valor da área é de: ",area)

fimalgoritmo

G) Fluxograma:



Pseudocódigo:

algoritmo "semnome"

var

raio, area:real

inicio

escreval ("Digite o valor do raio:")

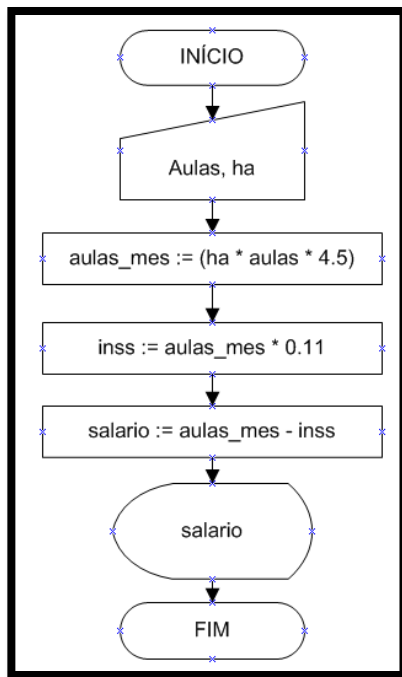
leia(raio)

area := 3.14 * raio * raio

escreval ("O valor da área é de: ",area)

fimalgoritmo

H) Fluxograma:



Pseudocódigo:

algoritmo "semnome"

var

ha, aulas, aulas_mes, inss, salario :real

inicio

escreval ("Digite o valor da hora aula:")

leia(ha)

escreval ("Digite a quantidade de aulas por semana:")

leia(aulas)

aulas_mes := (ha * aulas * 4.5)

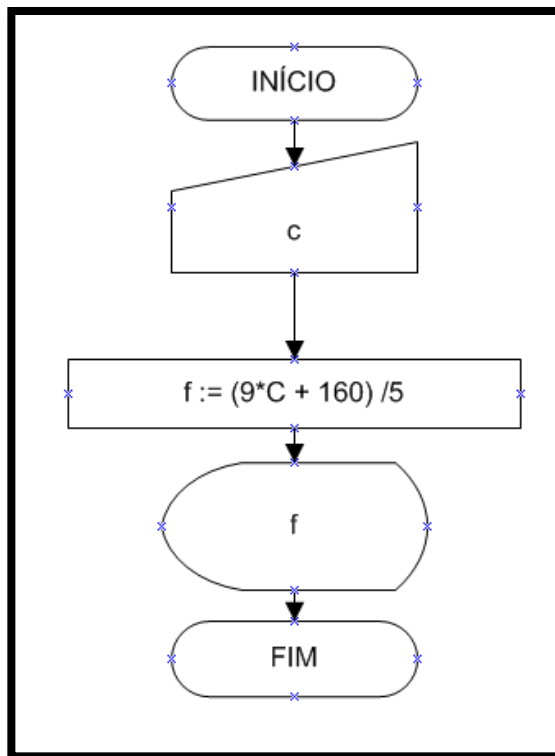
inss := aulas_mes * 0.11

salario := aulas_mes - inss

escreval ("O valor do salário do professor é de R\$",salario)

fimalgoritmo

I. Fluxograma:



Pseudocódigo:

algoritmo "semnome"

var

c, f :real

inicio

escreval ("Digite a temperatura em °C:")

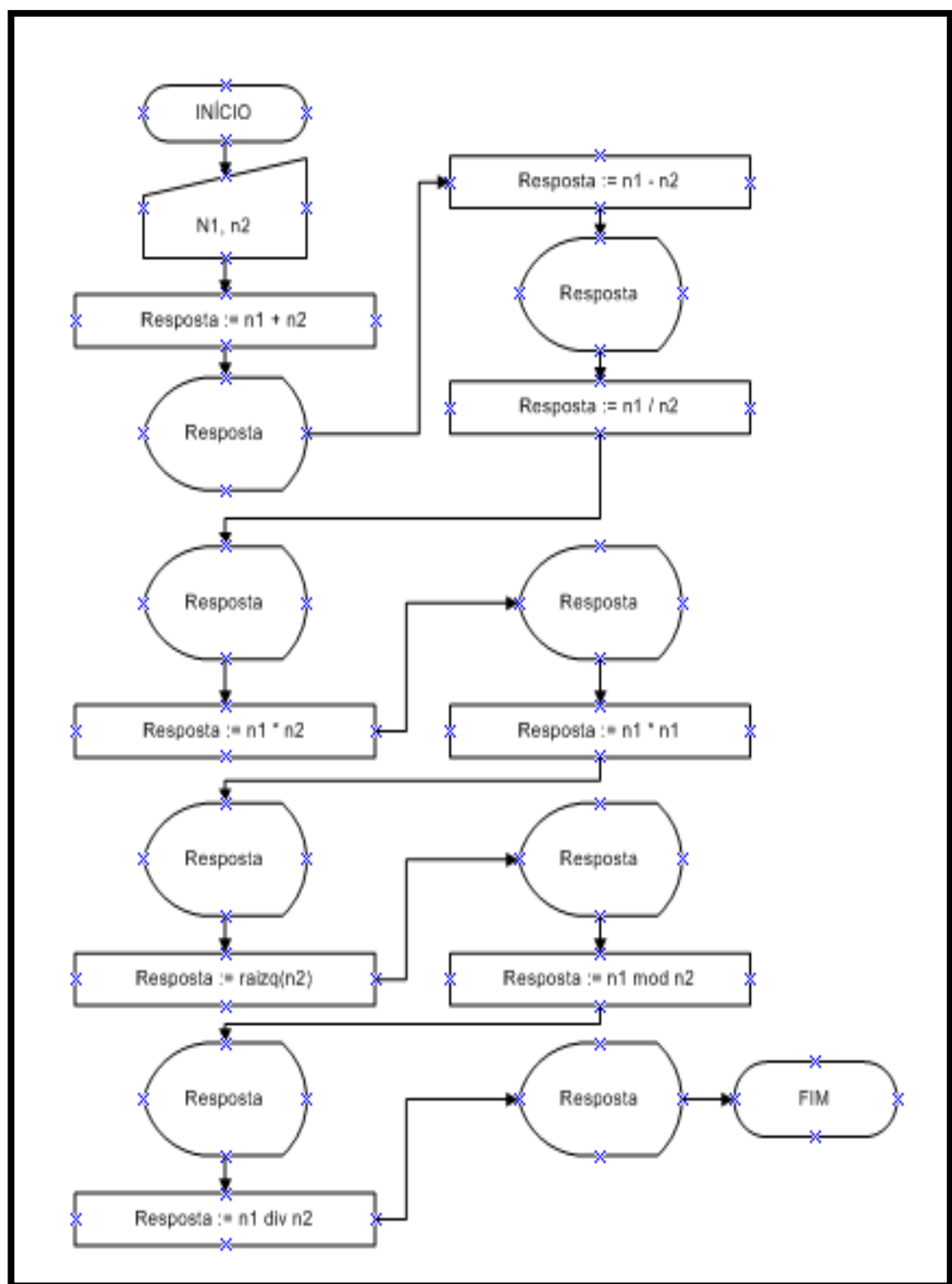
leia(c)

$f := (9 \cdot C + 160) / 5$

escreval ("Em Fahrenheit a temperatura fica °",f)

fimalgoritmo

J) Fluxograma:



Pseudocódigo:

algoritmo "semnome"

var

resposta :real

n1, n2,resposta2:inteiro

inicio

escreval ("Digite o primeiro numero:")

leia(n1)

escreval ("Digite o segundo numero:")

leia(n2)

resposta := n1 + n2

escreval ("A soma dos numeros é ",resposta)

resposta := n1 - n2

escreval ("A subtração dos numeros é ",resposta)

resposta := n1 / n2

escreval ("A divisão dos numeros é ",resposta)

resposta := n1 * n2

escreval ("A multiplicação dos numeros é ",resposta)

resposta := n1*n1

escreval ("O Quadrado do primeiro numero é ",resposta)

resposta := raizq(n2)

escreval ("A raiz quadrada do segundo numeros é ",resposta)

resposta2 := n1 mod n2

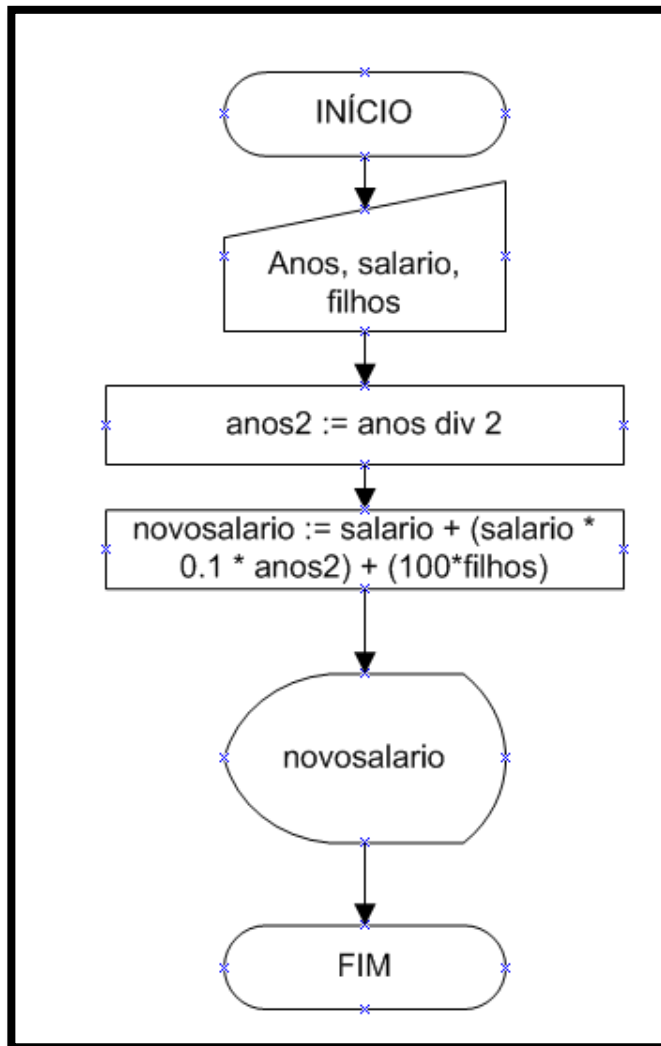
escreval ("O resto da divisão dos numeros é ",resposta2)

resposta2 := n1 div n2

escreval ("A parte inteira dos numeros é ",resposta2)

fimalgoritmo

k) Fluxograma:



Pseudocódigo:

algoritmo "semnome"

var

resposta,salario, novosalario :real

anos, filhos,anos2:inteiro

inicio

escreval ("Digite o seu salário:")

leia(salario)

escreval ("Digite quantos anos de empresa possui:")

leia(anos)

escreval ("Digite a quantidade de filhos que possui:")

leia(filhos)

anos2 := anos div 2

novosalario := salario + (salario * 0.1 * anos2) + (100*filhos)

escreval ("O seu novo salário é de R\$",novosalario)

fimalgoritmo

REFERÊNCIA BIBLIOGRAFICA

ALMEIDA, Rafael Soares. Aprendendo Algoritmo com VisuAlg. 1.ed. Editora Ciência Moderna, 2013.

BARRY, Paul; GRIFFITHS, David. Use a Cabeça! Programação. 1.ed. Alta Books, 2010.

FARRELL, Joyce. Lógica e Design de Programação. 1.ed. Cengage Learning, 2009.

XAVIER, Gley Fabiano Cardoso. Lógica de Programação. 12.ed. Senac, 2012.