

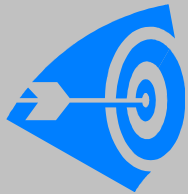
Introdução à Programação

Lógica de Programação



1.1 Lógica

A **lógica de programação** é necessária para pessoas que desejam trabalhar com desenvolvimento de sistemas e programas, ela permite definir a Seqüência lógica para o desenvolvimento.

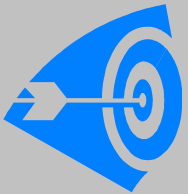


É a técnica de encadear pensamentos para atingir determinado objetivo.

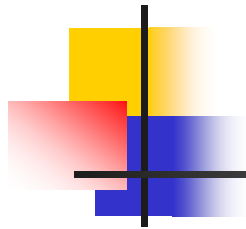


1.2 Seqüência Lógica

Estes pensamentos, podem ser descritos como uma seqüência de instruções, Que devem ser seguida para se cumprir uma determinada tarefa.



Seqüência Lógica são passos executados até atingir um objetivo ou solução de um problema.



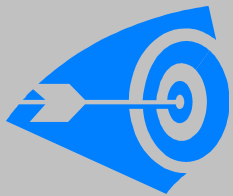
1.3 Instruções

Na linguagem comum, entende-se por instruções “**um conjunto de regras ou normas definidas para a realização ou emprego de algo**”.

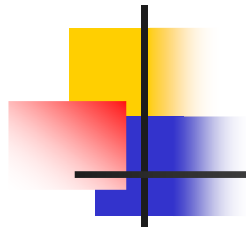
Em informática, porém, instrução é a informação que indica a um computador uma Ação elementar a executar.

Convém ressaltar que uma ordem isolada não permite realizar o processo completo, para isso é necessário um conjunto de instruções colocadas em ordem seqüencial lógica.

Por exemplo, se quisermos fazer uma omelete de batatas, precisaremos colocar em prática uma série de instruções: descascar as batatas, bater os ovos, fritar as batatas, etc...



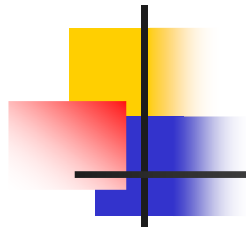
Instruções são um conjunto de regras ou normas definidas para a realização ou emprego de algo. Em informática, é o que indica ao computador uma ação elementar a executar.



1.4 Algoritmo

Um algoritmo é formalmente uma seqüência finita de passos que levam a execução de uma tarefa. Podemos pensar em algoritmo como uma receita, uma Seqüência de instruções que dão cabo de uma meta específica. Estas tarefas podem ser redundantes nem subjetivas na sua definição, devem ser claras e precisas.

Como exemplos de algoritmos podemos citar os algoritmos das operações básicas (adição, multiplicação, divisão e subtração) de números reais decimais. Outros exemplos seriam os manuais de aparelhos eletrônicos como uma televisão, que explicam passo-a-passo como, por exemplo, o auto-desligamento.

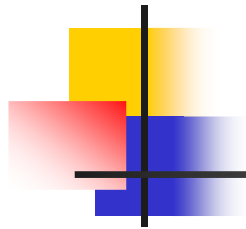


1.4 Algoritmo - Continuação

Até mesmo nas coisas mais simples, podemos descrever uma sequência lógica.
Por exemplo:

"Comer biscoitos"

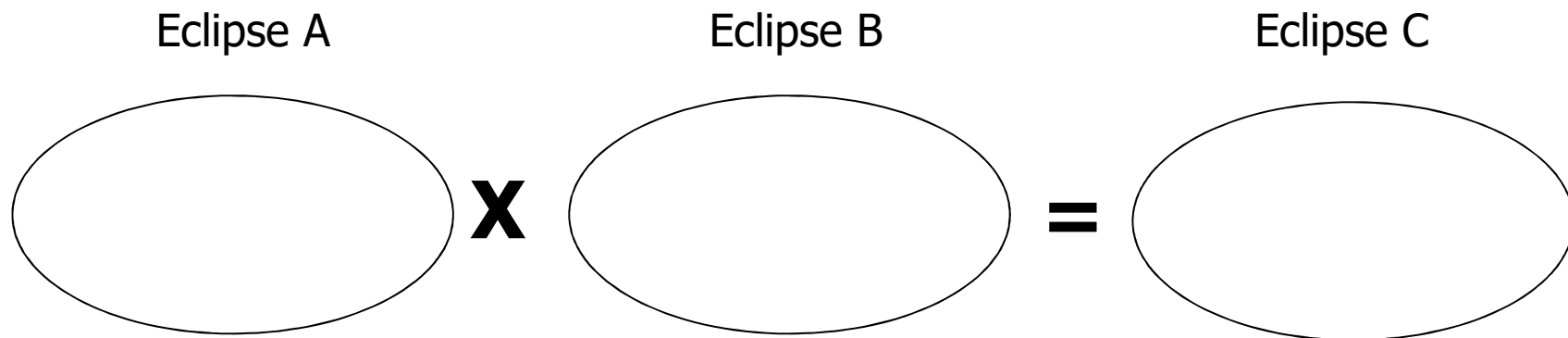
1. Pegar o pacote de biscoitos no armário
2. Abrir a embalagem dos biscoitos
3. Comer os biscoitos
4. Jogar no lixo a embalagem

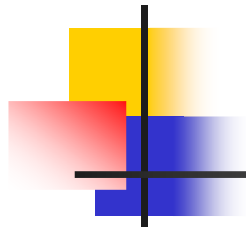


1.4 Algoritmo - Continuação

“Multiplicar dois números quaisquer”.

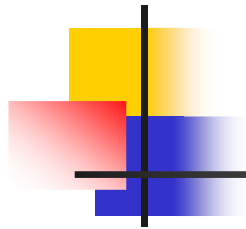
1. Escreva o primeiro número na eclipse A
2. Escreva o segundo número na eclipse B
3. Multiplique o número da eclipse A com o número da eclipse B e coloque o resultado na eclipse C



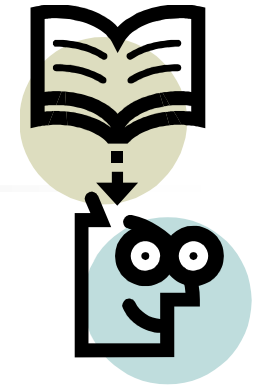


1.5 Programas

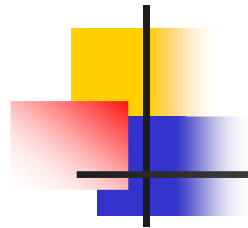
Os programas de computadores nada mais são do que algoritmos escritos numa linguagem de computador (Pascal, C++, Fortran, Delphi, Visual Basic, etc...) e que são interpretadas e executadas por uma máquina, no caso um computador. Notem que dada esta interpretação rigorosa, um programa é por natureza muito específico e rígido em relação aos algoritmos da vida real.



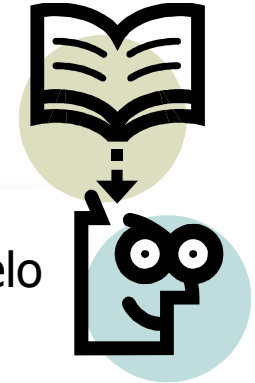
1.6 Exercícios



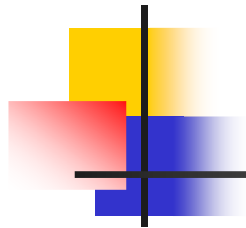
1) Crie uma seqüência lógica para lavar seu carro.



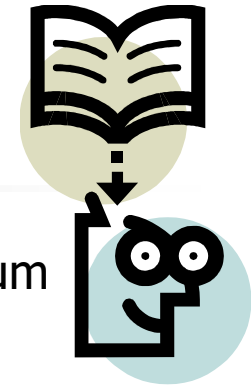
1.6 Exercícios - Continuação



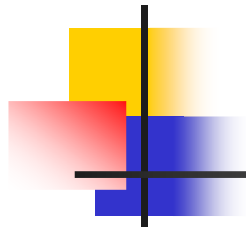
2) Faça um algoritmo p/ somar dois números e multiplicar o resultado pelo Primeiro número.



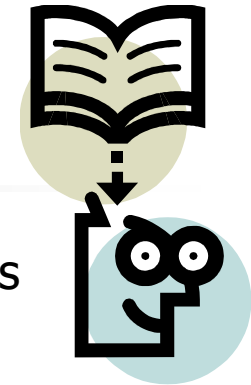
1.6 Exercícios - Continuação



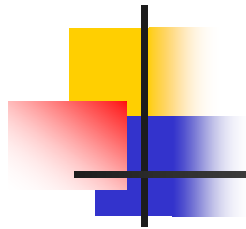
3) Descreva com detalhes a seqüência lógica para trocar um pneu de um carro



1.6 Exercícios - Continuação



4) Faça um algoritmo para trocar uma lâmpada. Descreva com detalhes

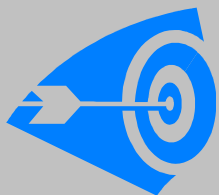


2 Desenvolvimento Algoritmo

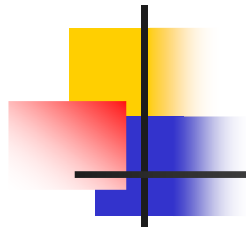
Pseudocódigo

Os algoritmos são descritos em uma linguagem chamada **pseudocódigo**. Este nome é uma alusão à posterior implementação em uma linguagem de programação, ou seja, quando formos programar em uma linguagem, por exemplo Visual Basic, estaremos gerando código em Visual Basic. Por isso os algoritmos são independentes das linguagens de programação. Ao contrário de uma linguagem de programação não existe um formalismo rígido de como deve ser escrito o algoritmo.

O algoritmo deve ser fácil de se interpretar e fácil de codificar. Ou seja, ele deve ser o intermediário entre a linguagem falada e a linguagem de programação.



O algoritmo deve ser fácil de se interpretar e fácil de codificar. Ou seja, ele deve ser o intermediário entre a linguagem falada e a linguagem de programação.



2.1 Regras para a construção do Algoritmo

Para escrever um algoritmo precisamos descrever a seqüência de instruções, de Maneira simples e objetiva. Para isso utilizaremos algumas técnicas:

- ✓ Usar somente um verbo por frase
- ✓ Imaginar que você está desenvolvendo um algoritmo para pessoas que não trabalham com informática
- ✓ Usar frases curtas e simples
- ✓ Ser objetivo
- ✓ Procurar usar palavras que não tenham sentido dúbio



2.1 Regras para a construção do Algoritmo – cont.

Para montar um algoritmo, é preciso primeiramente dividir o problema apresentado em três fases fundamentais.

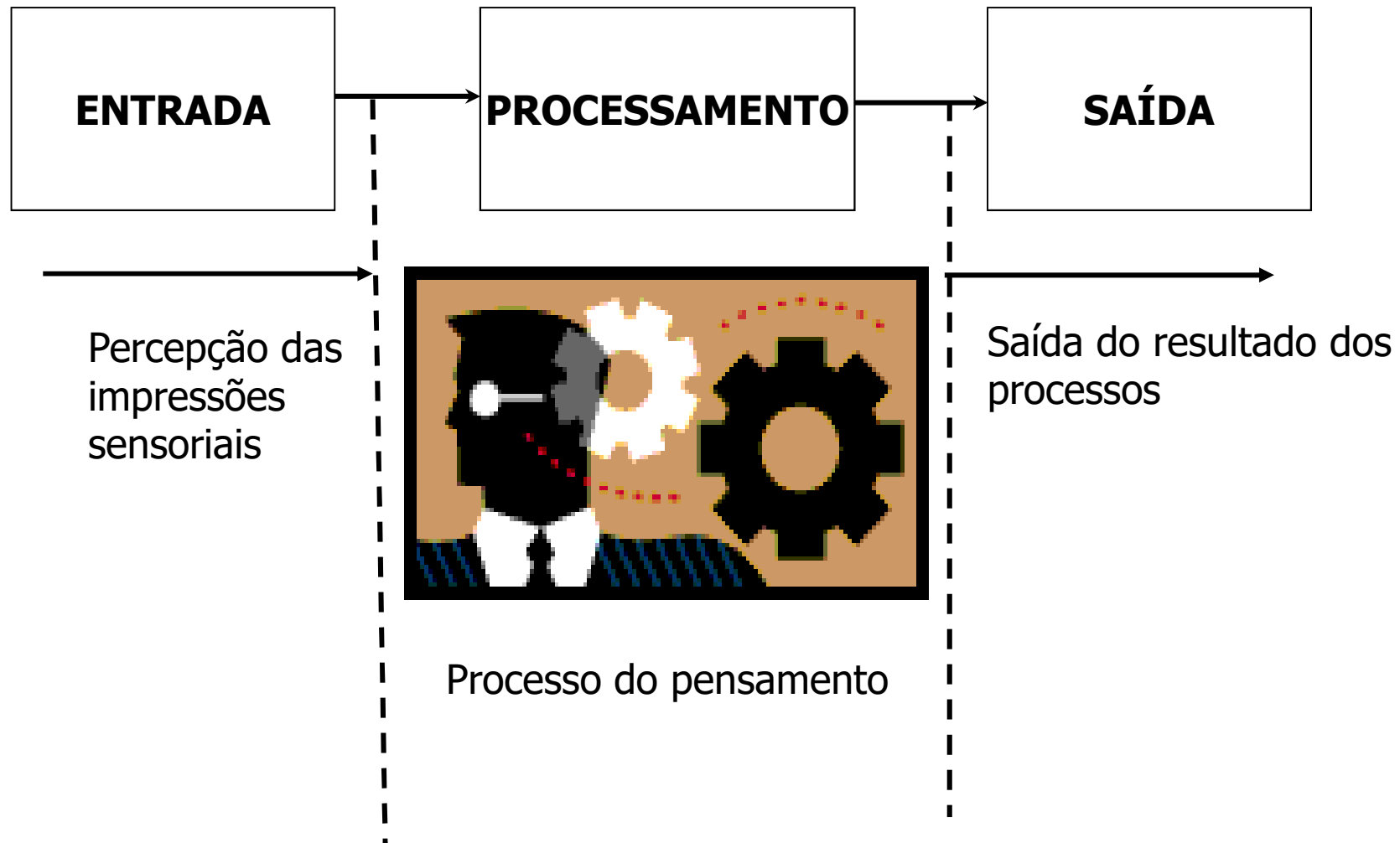


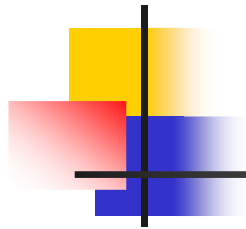
Entrada: são os dados de entrada do algoritmo

Processamento: são os procedimentos usados para chegar ao resultado final

Saída: são os dados já processados

Analogia de um algoritmo com o homem





2.2 Exemplo de Algoritmo

Exemplo de Algoritmo

Imagine o seguinte problema: Calcular a média final dos alunos da 3ª Série. Os alunos realizarão quatro provas: P1, P2, P3 e P4.

Onde:

$$\text{Média Final} = P1 + P2 + P3 + P4 / 4$$

Para montar o algoritmo proposto, faremos três perguntas:

a) Quais são os dados de entrada?

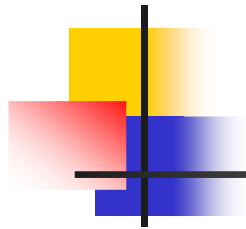
R: Os dados de entrada são P1, P2, P3 e P4

b) Qual será o processamento a ser utilizado?

R: O procedimento será somar todos os dados de entrada e dividi-los por 4 (quatro)

$$P1 + P2 + P3 + P4 / 4$$

c) Quais serão os dados de saída?



2.2 Exemplo de Algoritmo - Continuação

R: O dado de saída será a média final

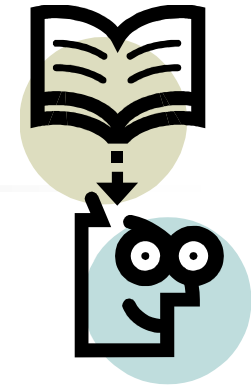
Algoritmo

Receba a nota da prova1
Receba a nota de prova2
Receba a nota de prova3
Receba a nota da prova4
Some todas as notas e divida o resultado por 4

Mostre o resultado da divisão



2.3 Exercícios



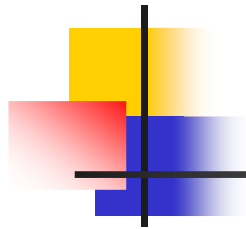
1) Identifique os dados de entrada, processamento e saída no algoritmo abaixo

- ✓ Receba código do pneu
- ✓ Receba valor do pneu
- ✓ Receba Quantidade de pneus
- ✓ Calcule o valor total do pneus(Quantidade * Valor do pneu)
- ✓ Mostre o código do pneu e seu valor total

2) Faça um algoritmo para “Calcular o estoque médio de carros de uma concessionária”, sendo que:

$$\text{ESTOQUE M\u00c9DIO} = (\text{QUANTIDADE M\u00cdNIMA} + \text{QUANTIDADE M\u00c1XIMA}) / 2$$

3) Teste o algoritmo anterior com dados definidos por voc\u00ea.

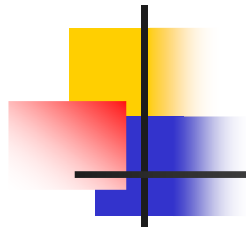


3 Diagrama de Bloco

O que é um diagrama de bloco?

O diagrama de blocos é uma forma padronizada e eficaz para representar os passos lógicos de um determinado processamento.

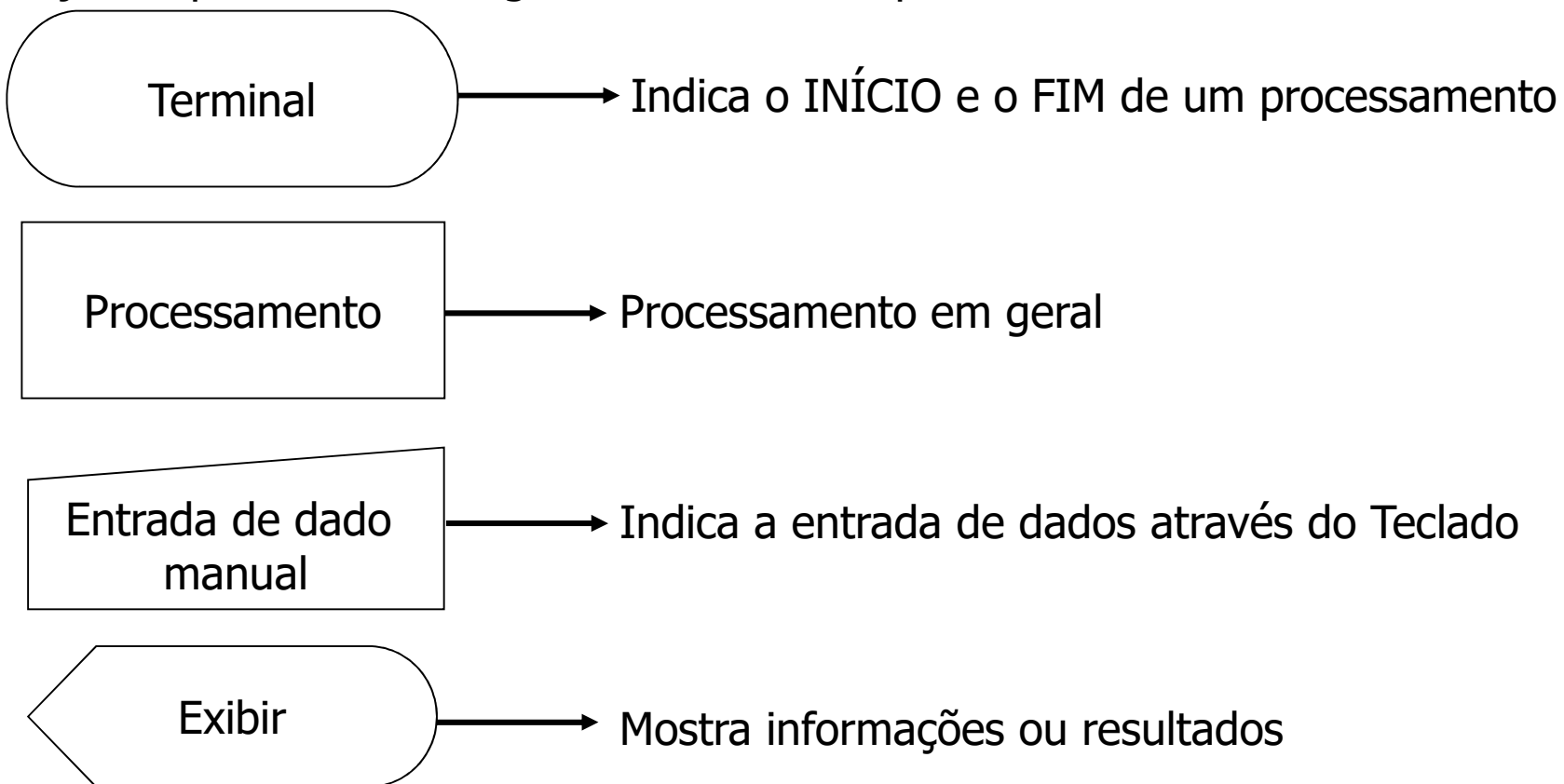
Com o diagrama podemos definir uma seqüência de símbolos, com significado bem definido, portanto, sua principal função é a de facilitar a visualização dos passos de um processamento.



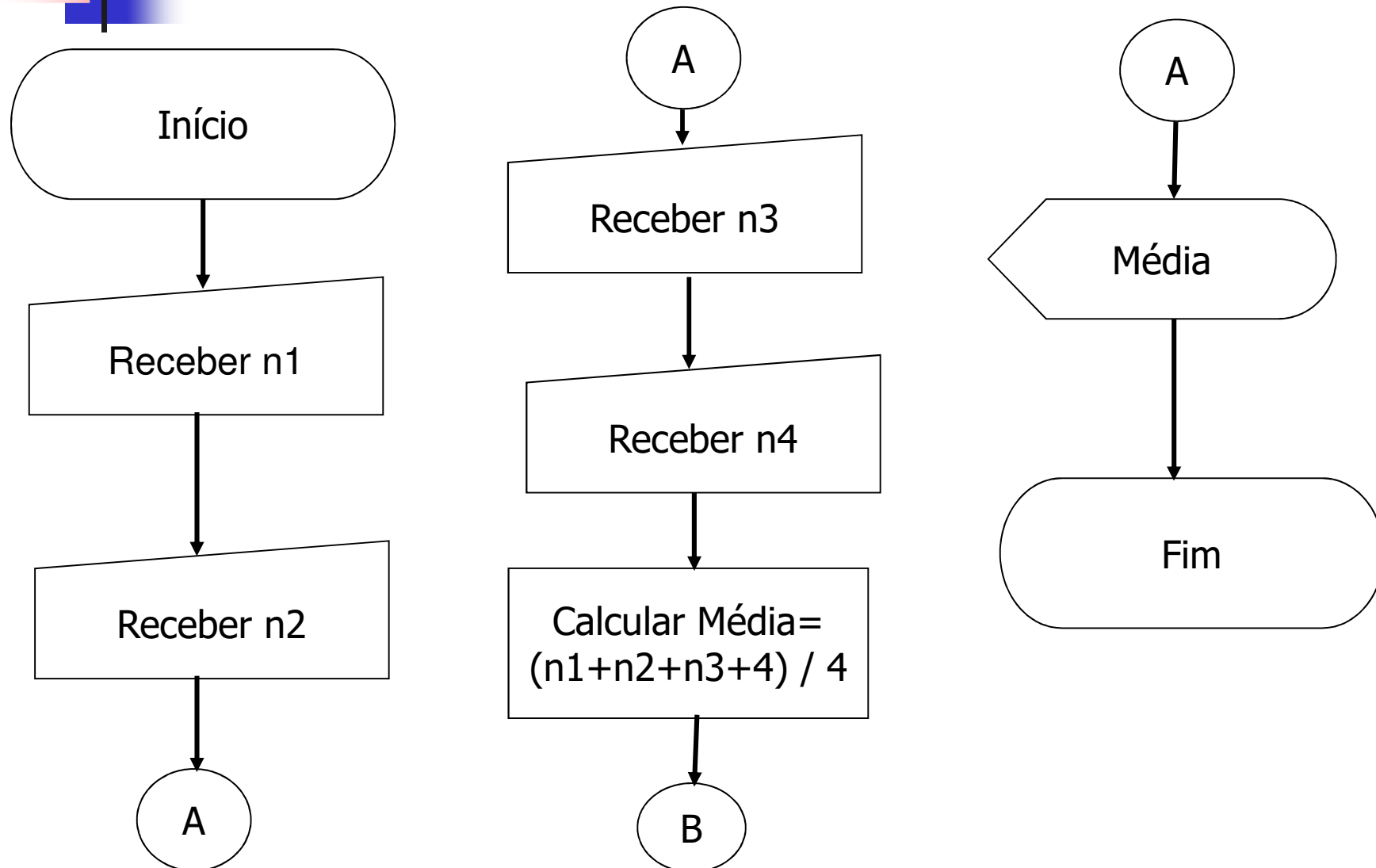
3.1 Simbologia

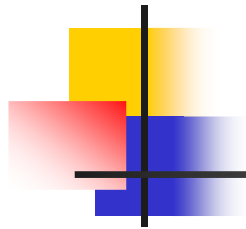
Existem diversos símbolos em um diagrama de bloco. No decorrer do curso apresentaremos os mais utilizados.

Veja no quadro abaixo alguns dos símbolos que iremos utilizar:



3.2 Exemplo de Diagrama de Bloco





3.3 Exercícios

1) Construa um diagrama de blocos que :

- ✓ Leia a cotação do dólar
- ✓ Leia um valor em dólares
- ✓ Converta esse valor para Real
- ✓ Mostre o resultado

2) Desenvolva um diagrama que:

- ✓ Leia 4 (quatro) números
- ✓ Calcule o quadrado para cada um
- ✓ Somem todos e
- ✓ Mostre o resultado

3) Construa um algoritmo para pagamento de comissão de vendedores de peças, levando-se em consideração que sua comissão será de 5% do total da venda e que você tem os seguintes dados:

- ✓ Identificação do vendedor
- ✓ Código da peça
- ✓ Preço unitário da peça
- ✓ Quantidade vendida

