

Turma Compartilhada

Programação de Computadores

Prof. Carlos Eduardo Weber



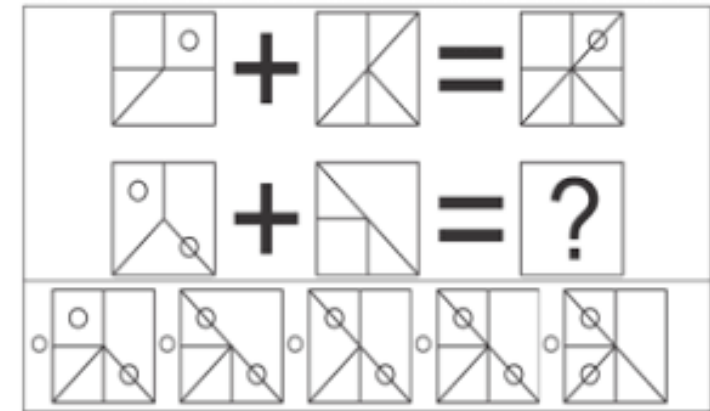


Aula 02 - Conceitos

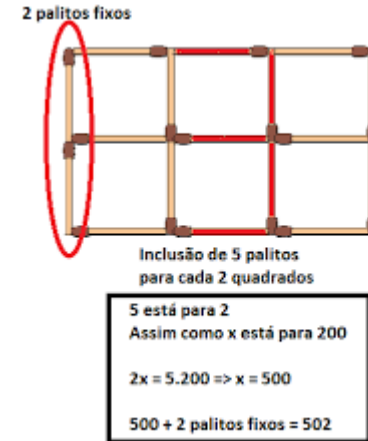
25/02/2025

✓ Raciocínio Lógico:

- Pensar antes de agir.
- Entender o problema proposto e suas possíveis soluções.
- Pensar com coerência sobre os vários caminhos possíveis e tomar a melhor decisão para alcançar o objetivo.



✓ Raciocínio Lógico:



- Trata das formas do **pensamento** em geral (dedução, indução, hipótese, inferência, entre outros.) e das operações intelectuais que visam à determinação do que é **verdadeiro ou não**.

✓ Lógica de programação:

- Transferência do pensar humano para a máquina.

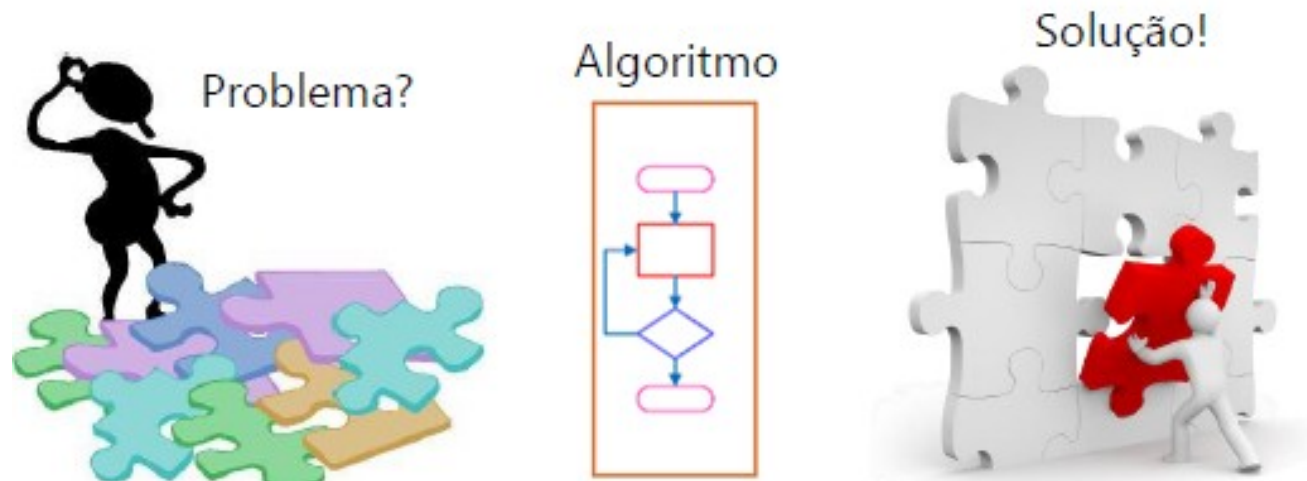


- ✓ Nós utilizamos a **lógica para resolver** um problema;
- ✓ A resolução do problema é o **objetivo**;
- ✓ Para chegar ao objetivo é necessário **seguir uma sequência**;
- ✓ Nesta sequência estão contidas **instruções**.
- ✓ Sendo assim, tudo isso se resume em:



✓ Definição de Algoritmo:

- Sequência **passo a passo FINITA** descritiva com o intuito de alcançar um objetivo bem definido.
- Tem por objetivo representar de forma fiel o raciocínio envolvido na Lógica de Programação.

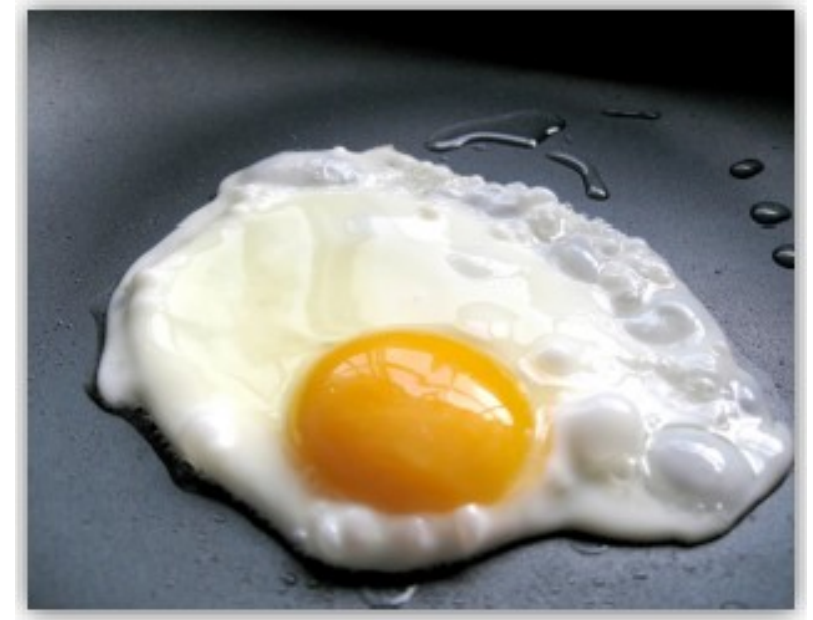


✓ No desenvolvimento de ALGORITMOS deve-se seguir algumas premissas:

- Ter um **COMEÇO** e um **FIM**, as ações devem ter uma sequência **finita** de passos;
- As instruções devem ser organizadas em uma **sequência lógica**;
- As instruções devem ser descritas de forma precisa, simples e sem **ambiguidades**.

Possível resolução do algoritmo “Fritar ovos”:

- ✓ Separar ingredientes
 - manteiga
 - sal
 - ovos
- ✓ Separar utensílios
 - colher
 - frigideira
- ✓ Preparar ambiente
 - acender o fogo
 - colocar a frigideira no fogão
- ✓ Modo de preparo
 - colocar a manteiga na frigideira, depois junte os ovos, coloque sal e mexa.
 - quando os ovos fritarem, desligue o fogo.

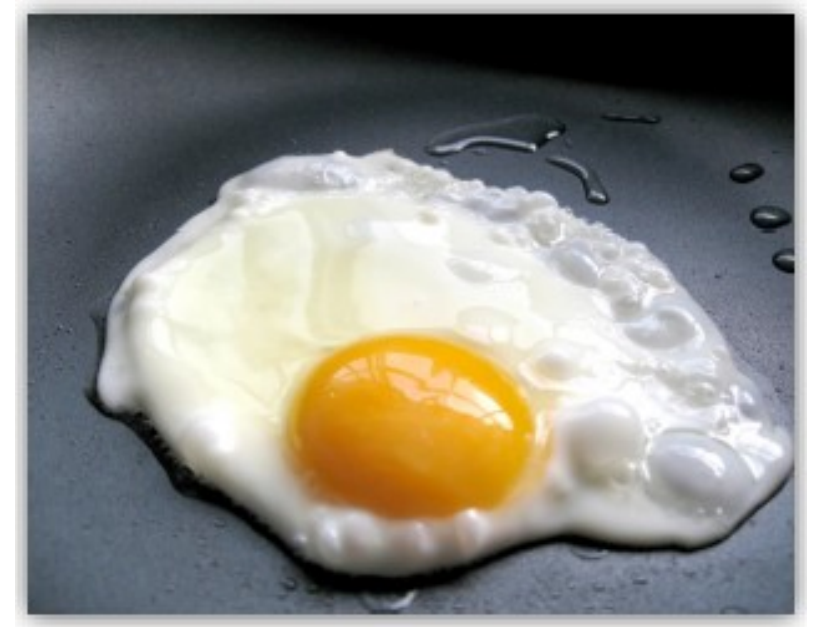


Alguns problemas com o algoritmo do “Fritar ovos”

- ✓ A linguagem natural é imprecisa:
 - Quanto de manteiga?
 - Quanto de Sal?
 - Quantos ovos?
 - Como mexer?
 - O que define que está pronto?
- ✓ O uso de uma linguagem mais restritiva e estruturada seria mais adequado para certas situações;
- ✓ Sem ambiguidades (possíveis entendimentos diferentes);
- ✓ Utilizar uma linguagem simples e genérica.

Veja o algoritmo do “Fritar ovos” de forma mais definida:

- separar 2 ovos;
- separar 100g de manteiga;
- separar 5g de sal;
- separar frigideira;
- separar colher de pau;
- acender fogo do fogão;
- colocar frigideira no fogo;
- colocar manteiga na frigideira;
- deixar a manteiga derreter;
- colocar o sal;
- quebrar levemente a casca do primeiro ovo;
- colocar a clara e a gema na frigideira;
- repetir os dois passos anteriores para o segundo ovo;
- mexer os ovos com a colher;
-



✓ Instrução de um Algoritmo:

- Cada linha do algoritmo pode ser chamada de instrução;
- Logo, pode-se dizer que um algoritmo é um conjunto de instruções;
- Uma instrução indica a um computador uma ação elementar a ser executada.

✓ Qual a conclusão que podemos tirar sobre os algoritmos?

- Algoritmo é uma espécie de **passo a passo**.
- Algoritmos realizam uma saída (ex. Um ovo frito) a partir de uma entrada (um ovo) através de uma **sequência**.
- Os passos têm que ser executados **um após o outro**.
- Um algoritmo está correto quando a sua sequência de instrução resulta em uma **saída esperada**.
- Podem existir um ou mais algoritmos para atingir um resultado, desde que o **resultado seja o mesmo**.

O algoritmo não é a
solução do problema,
— — — — —
mas o caminho que leva à
solução.



✓ As principais tarefas realizadas por um Algoritmo são:

- Ler e escrever dados;
- Tomar decisões com base nos resultados das expressões avaliadas;
- Repetir um conjunto de ações de acordo com uma condição.

✓ Formas de representação de Algoritmos:

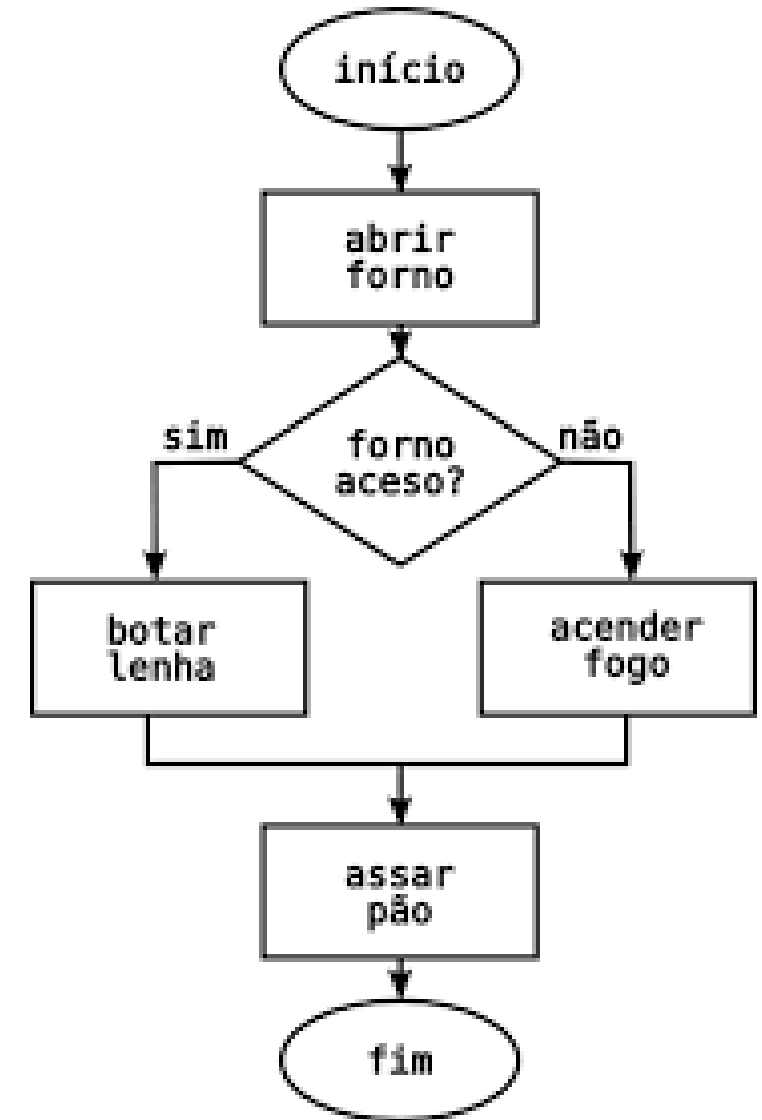
- Quando falamos em algoritmo sob o ponto de vista computacional **não podemos** utilizar uma **descrição narrativa para representá-lo**.
- Sendo assim, um **algoritmo** pode ser **representado** por:
 - Diagrama de Bloco (Fluxograma)
 - Pseudocódigos (Portugol)

✓ Diagrama de Bloco (Fluxograma)

- Uma **representação gráfica** que visa descrever métodos e sequências de um processo;
- Por meio de figuras geométricas (**símbolos**) e ligações (**setas**), descreve-se as **relações entre cada instrução** e o **fluxo** de informação ilustrando de forma descomplicada a transição de informações entre os elementos que o compõem;
- Os **símbolos** devem ser dispostos em **ordem lógica** e com **sintaxe correta** para atingir o objetivo de resolver o problema;
- Vantagem: O entendimento de elementos gráficos é mais **fácil** que o entendimento de texto.

✓ Diagrama de Bloco (Fluxograma)

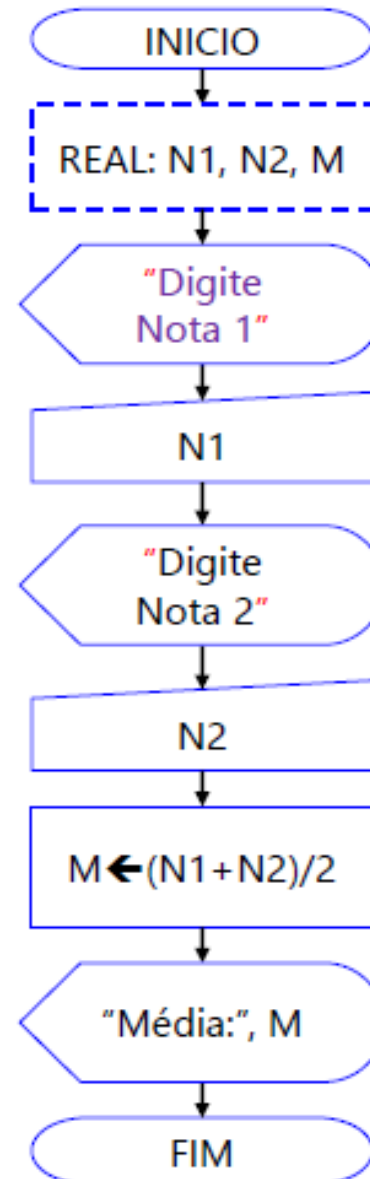
- Vantagem:
 - O entendimento de elementos gráficos é mais **fácil** que o entendimento de texto.
 - Para a programação e uso da lógica eficiente.



✓ Exemplo de Fluxograma

Flowgorithm

(<http://www.flowgorithm.org/download/>)

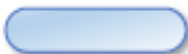










Este algoritmo apresenta os passos para se calcular a média de duas notas (N1 e N2) obtidas por um (a) aluno (a).

INSTRUÇÕES

- Relações
- Figuras Geométricas
- Ordem lógica
- Sintaxe correta

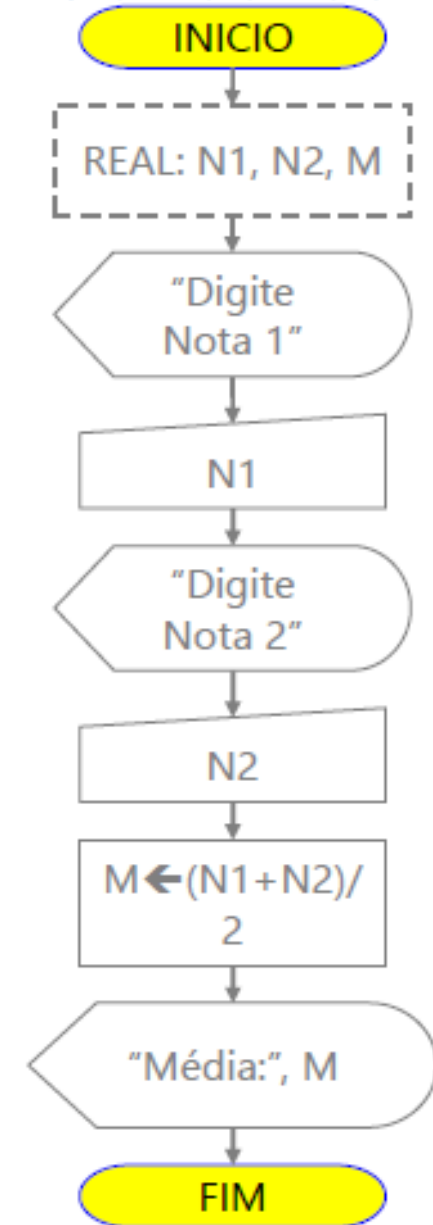
✓ Símbolos de Fluxograma

SÍMBOLOS	DESCRIÇÃO
	TERMINAL: símbolo utilizado como ponto para indicar o início e/ou fim do fluxo de um programa.
	TECLADO: símbolo que representa a entrada de dados por meio de um teclado de computador.
	DECLARAÇÃO: símbolo utilizado para representar a declaração de variáveis e/ou constantes a serem utilizadas no processo do fluxo.
	PROCESSAMENTO: símbolo ou bloco que representa algum tipo de processamento como cálculo, atribuições de valores ou qualquer manipulação de dados.
	SAÍDA DE DADOS EM VÍDEO: símbolo que representa a exibição de dados e/ou informações em dispositivos visuais, vídeo ou monitor.
	SAÍDA DE DADOS EM IMPRESSORA: símbolo que representa a impressão de dados e/ou informações.
	DECISÃO: símbolo que representa uma decisão que deve ser tomada indicando a possibilidade de desvios para diversos pontos do fluxo mediante a análise de uma condição.
	CONECTOR: símbolo que representa o particionamento de um diagrama.
	SETA DE FLUXO DE DADOS: símbolo que permite indicar o sentido do fluxo de dados. Serve exclusivamente para conectar outros símbolos ou blocos.

✓ Terminal

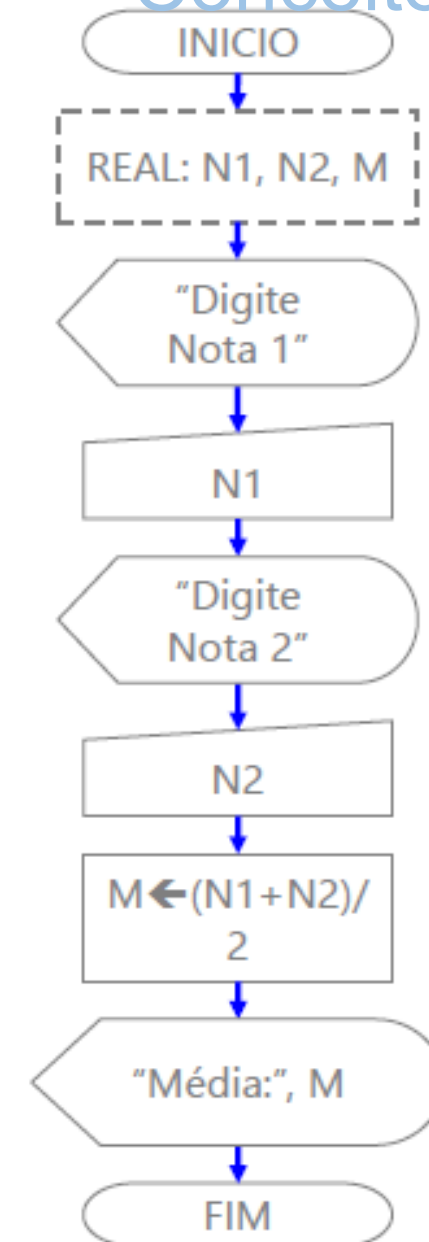
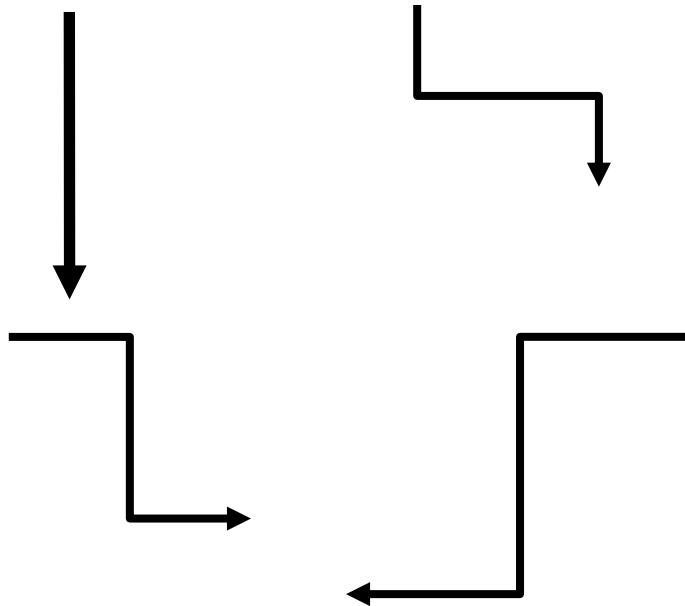


- Identifica o início ou o fim do fluxograma



✓ Fluxo

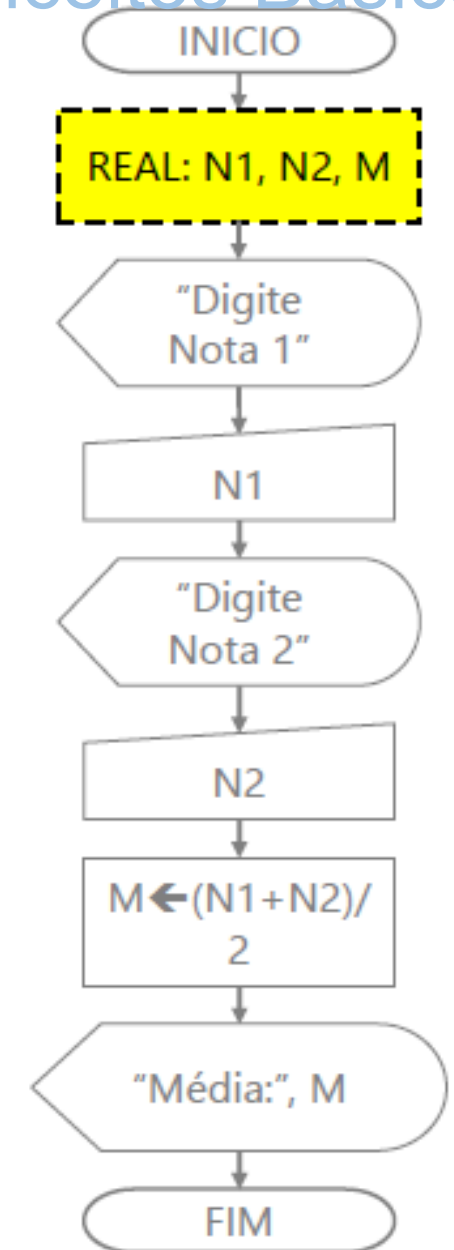
- Exibe o fluxo do fluxograma



✓ Declaração



- Símbolo usado para declarar as variáveis de um programa.

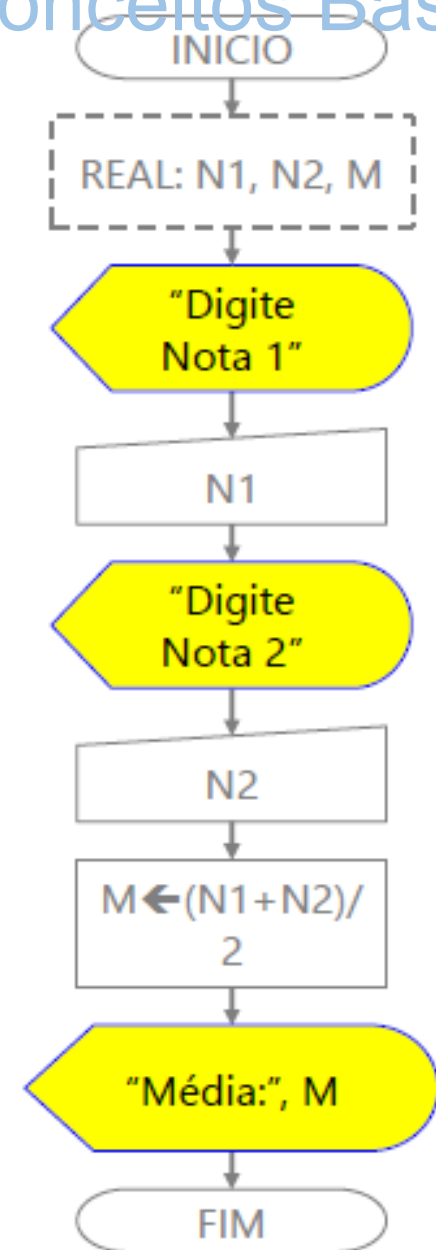


✓ Saída (exibição) de dados em vídeo

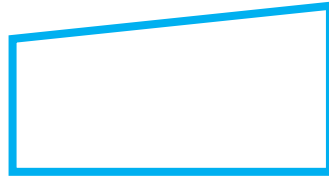
- Símbolo usado para mostrar algo na saída de vídeo.



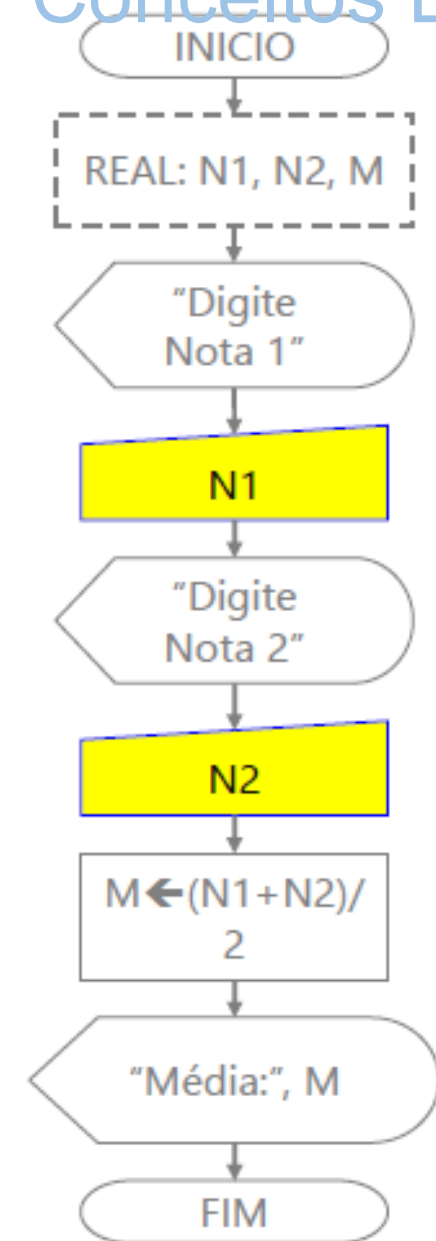
- Observação
- Caso o dado deva ser exibido (impresso) numa impressora, o símbolo a seguir deve ser usado:



- ✓ Entrada de dados (terminal / teclado)

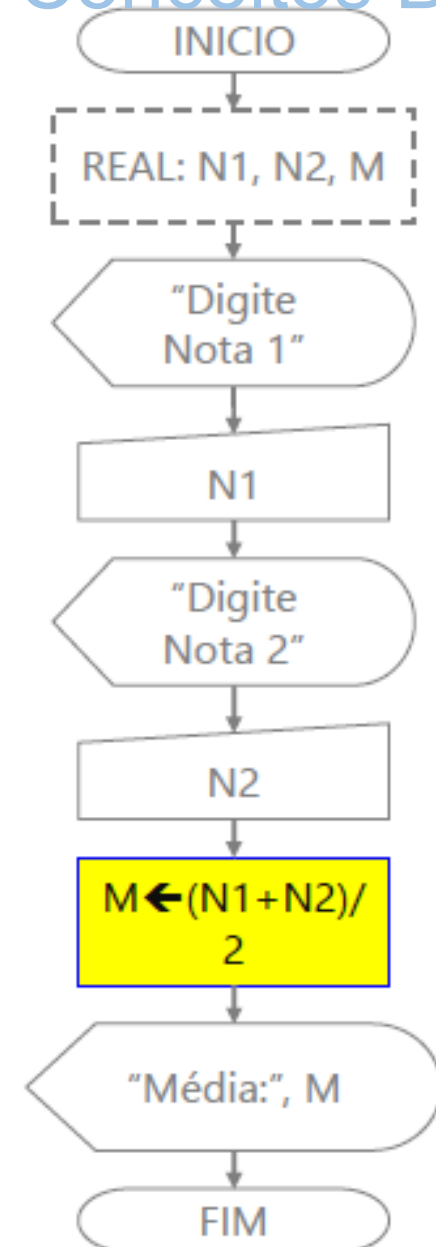
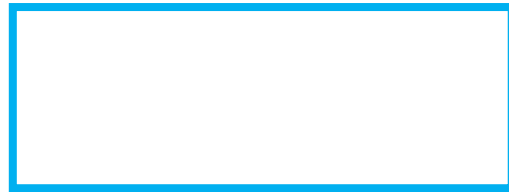


- Símbolo alternativo de entrada e saída:



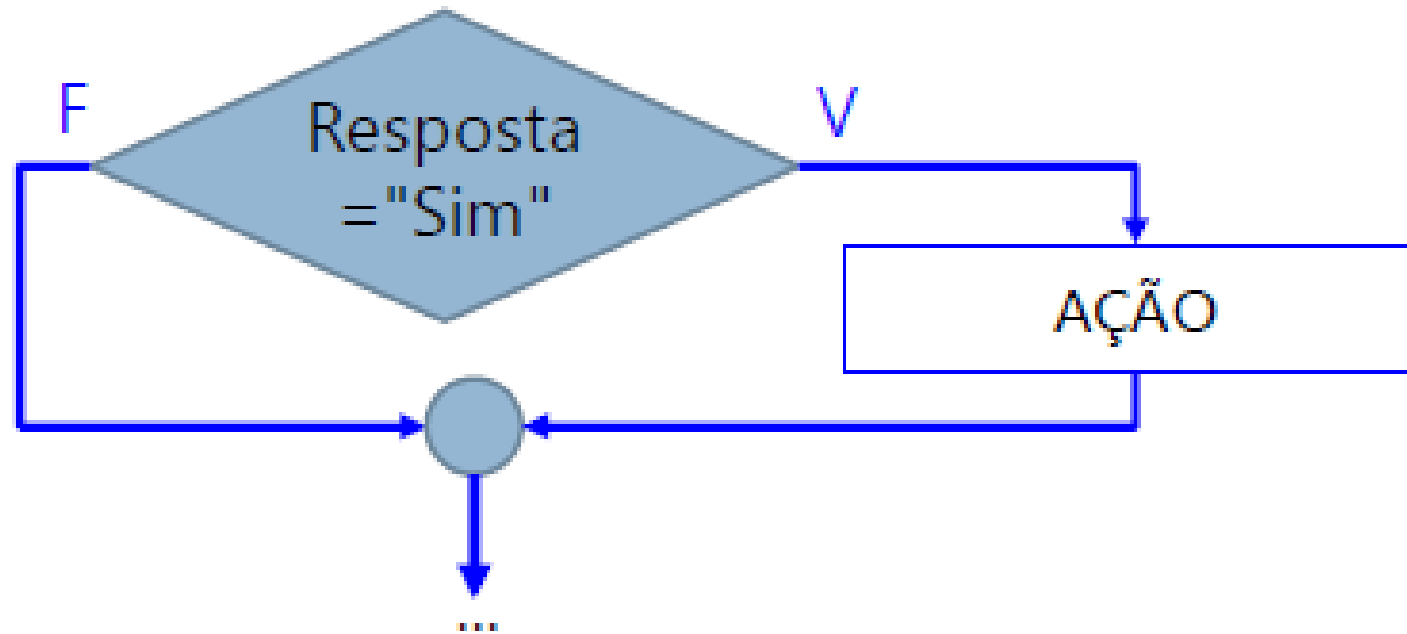
✓ Ação ou Processamento

- Identifica uma ação ou um processamento (exemplo: cálculo ou atribuição)



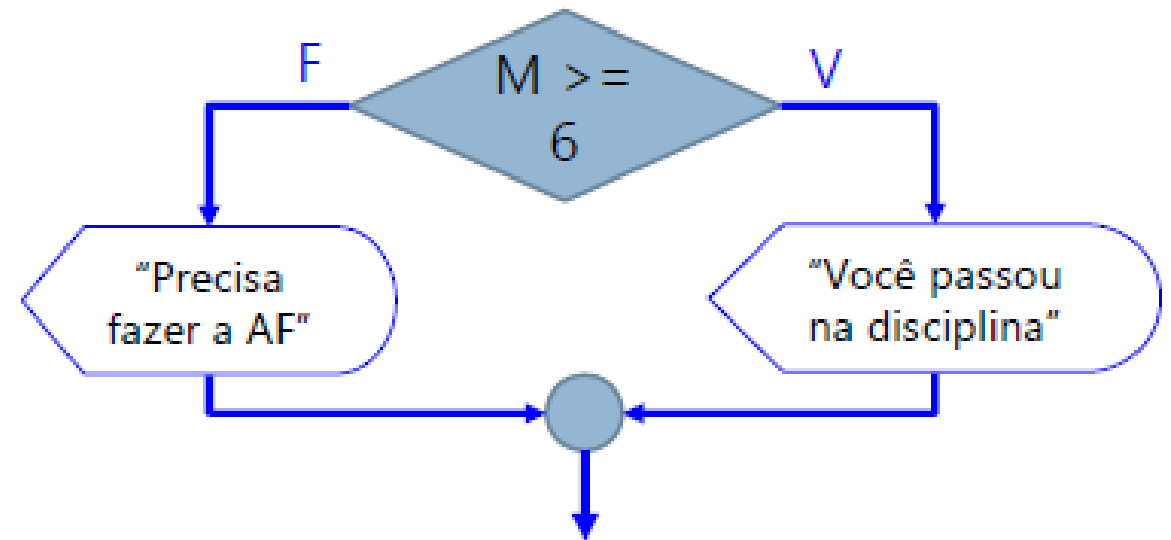
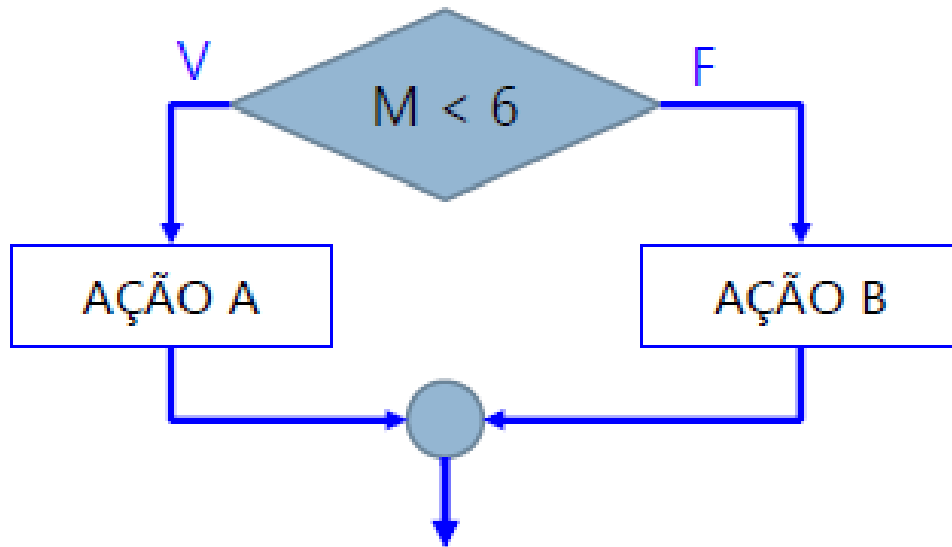
✓ Decisão

- Avalia uma determinada condição e executa uma ação.

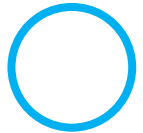


✓ Decisão

- Outra possibilidade é avaliar uma determinada condição e conforme o resultado, o fluxo segue por um caminho ou outro, executando ações diferentes.

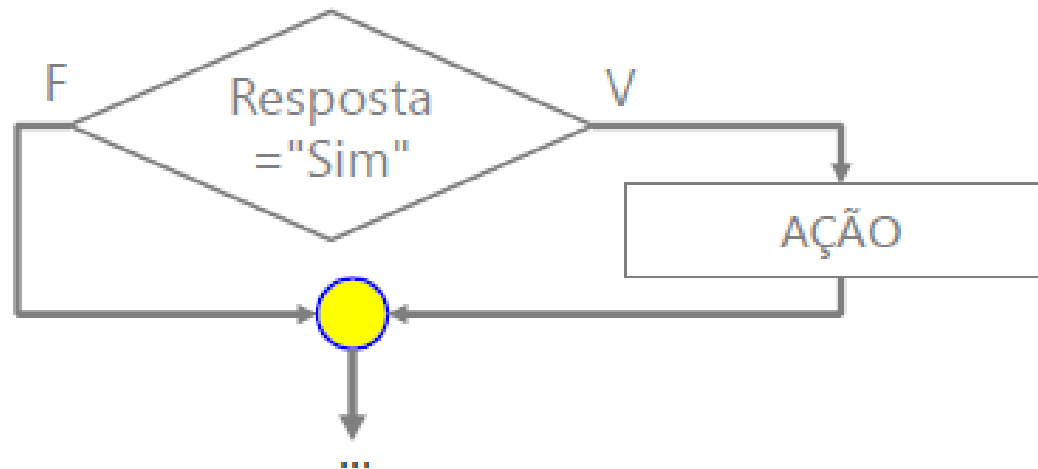


✓ Conector



Reunificação de fluxo

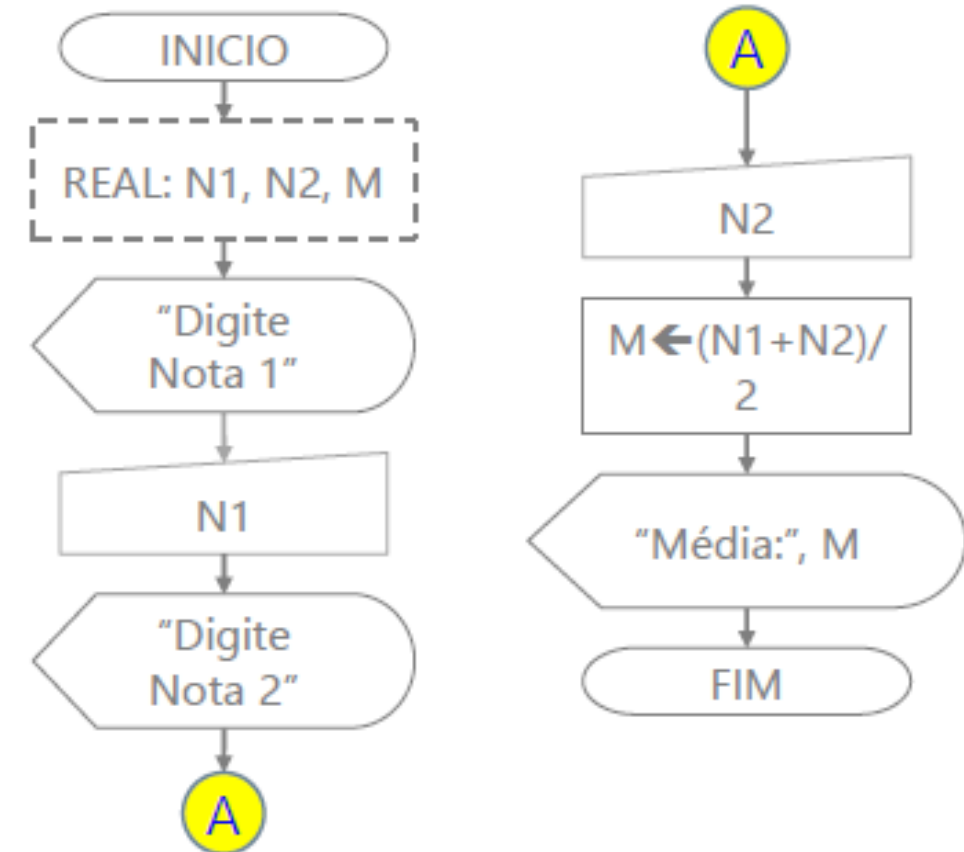
- Usado para conectar as opções de uma decisão de múltipla escolha para voltar ao mesmo fluxo.



✓ Conector

○ De mesma página

- Usado para conectar partes de um fluxograma.
- Quando a área disponível na mesma página não tem mais espaço, pode-se usar o conector para indicar o local, **na mesma página**, onde o fluxo continua.

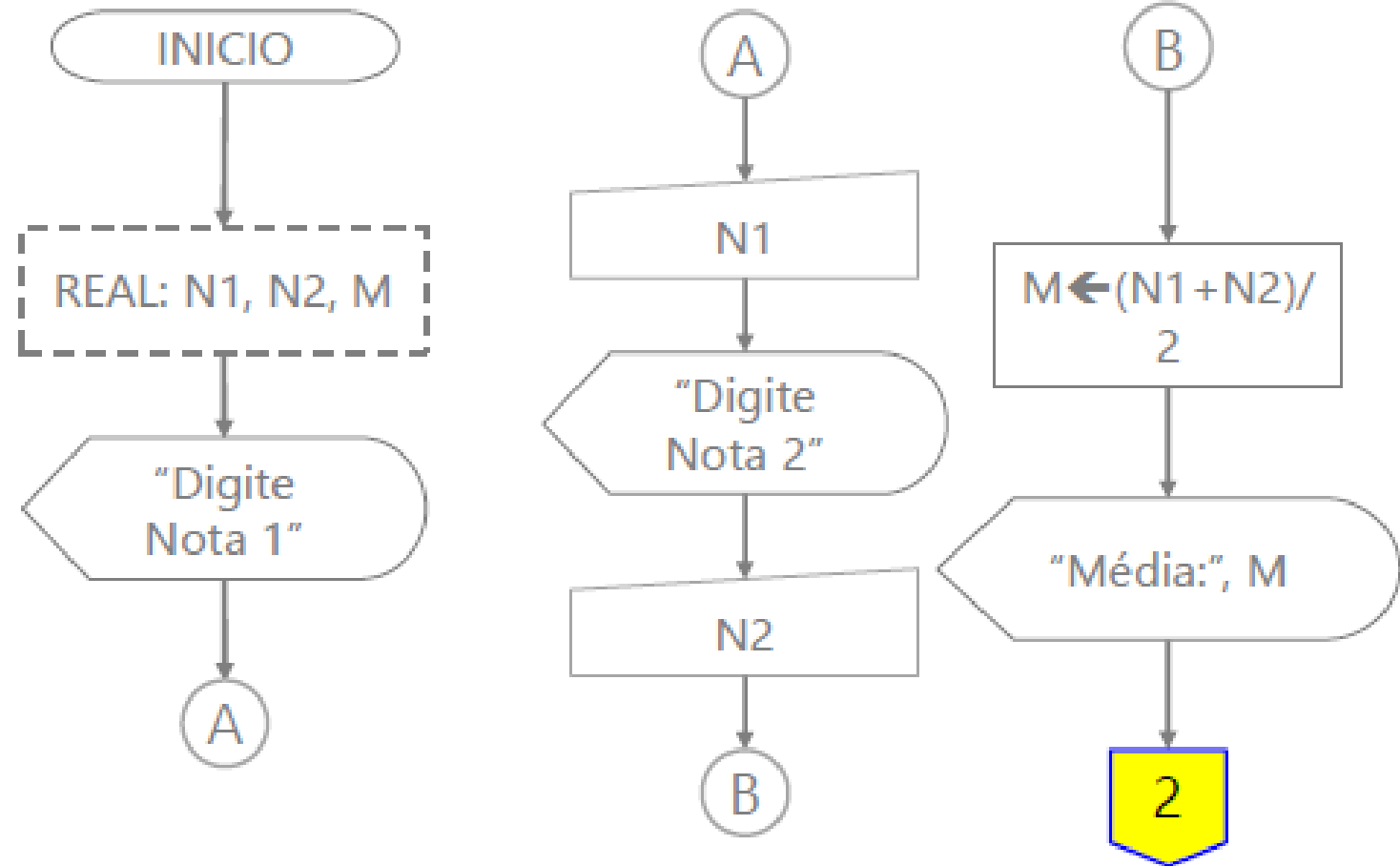


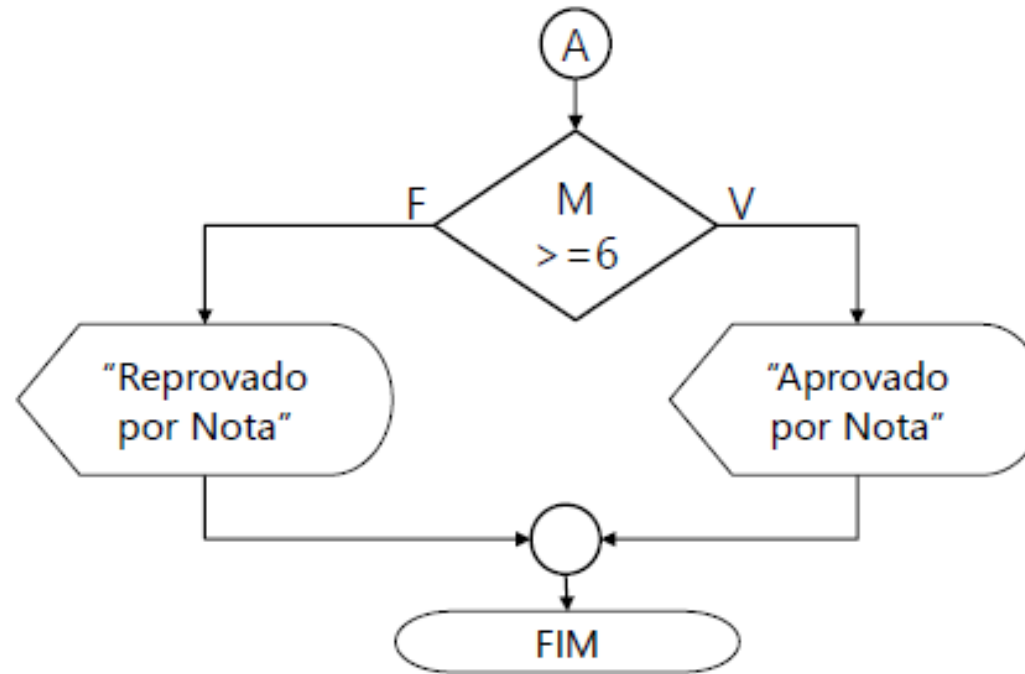
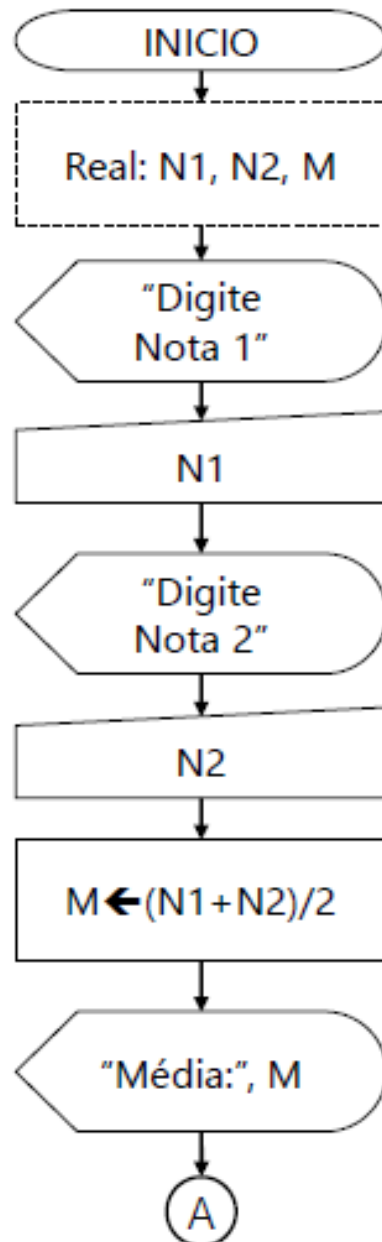
✓ Conector



De outra página

- Usado para conectar partes de um fluxograma.
- Quando a página não tem mais espaço disponível, pode-se usar o conector para indicar o local (**número de outra página**), onde o fluxo continua.





Este algoritmo apresenta os passos para se calcular a média de duas notas (N1 e N2) obtidas por um (a) aluno (a).

Em seguida, ele verifica se a média é igual ou maior que 6, e desta forma, informa se o (a) aluno (a) foi aprovado (a) ou reprovado (a) por nota, na disciplina.

✓ Pseudocódigo

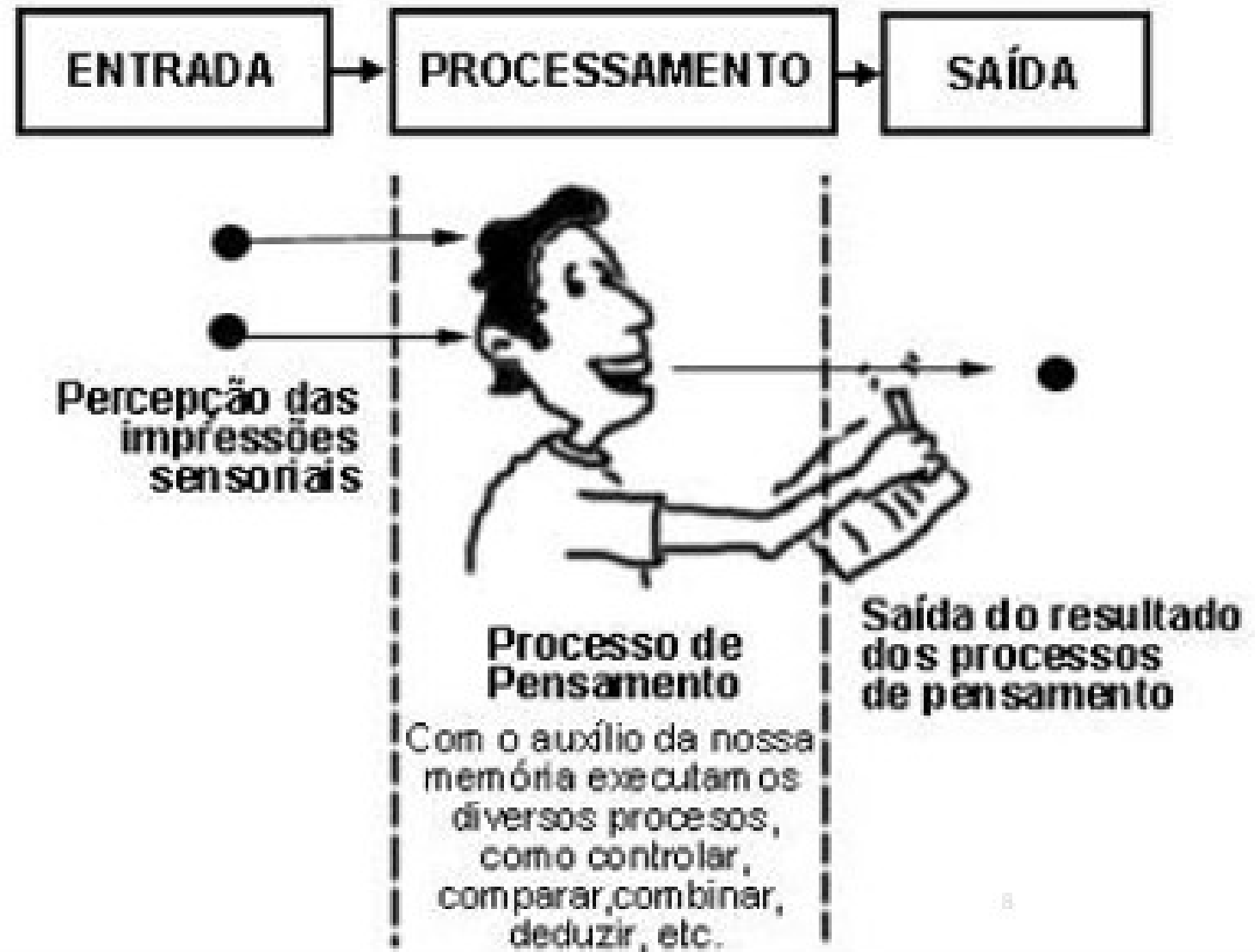
- Uma forma genérica de se escrever um algoritmo **sem necessidade de conhecer sintaxe** de nenhuma linguagem de programação;
- Utiliza uma linguagem **simples**;
- Ainda **incapaz** de ser executado num sistema real (computador):
 - Computadores **não entendem** pseudocódigos.
- Sua **transcrição** para qualquer linguagem de programação é quase que direta.
- É necessário aprender as **regras** do pseudocódigo.

✓ Exemplo de Pseudocódigo

```
algoritmo "media"
var N1, N2, M: real
inicio
    escreva ("Digite Nota 1:")
    leia(N1)
    escreva ("Digite Nota 2:")
    leia(N2)
    M <- (N1+N2) / 2
    escreva ("Média:", M)
finalgoritmo
```

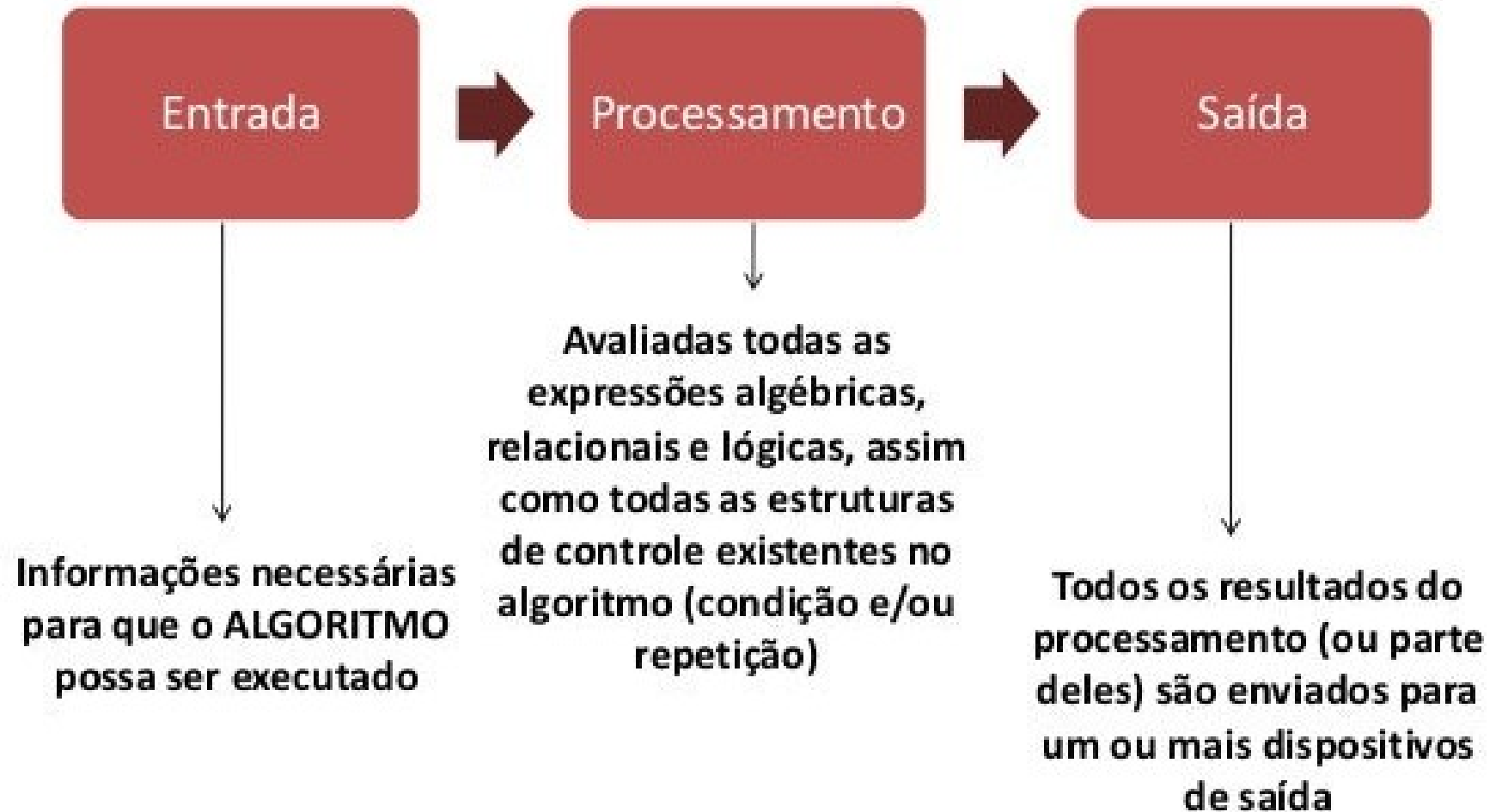
- VisualG (<http://visualg3.com.br/>)

Estrutura de um programa



Forma humana ...

No computador ...



✓ Um programa de computador é dividido em:

- Entrada de dados
- Processamento
- Saída de Dados

✓ De maneira simplificada

- Entrada >> Processamento >> Saída

Partes de um ALGORITMO




✓ Visão do Usuário

- Para o **usuário** do programa as regiões que mais importam são:
 - **entrada** de dados
 - **saída** dos dados

✓ Visão do Programador

- Para o **programador** além de entender qual o tipo de dado de entrada, faz-se necessário entender os **processos de transformação do dado (processamento)**, e qual a sua respectiva saída.

 testemesa.py

File Edit Format Run Options Window Help

```
numerol = int(input("Digite o primeiro numero:"))  
numero2 = int(input("Digite o segundo numero:"))  
resultado = numerol + numero2  
print("O resultado eh:", resultado)
```

usuário

programador

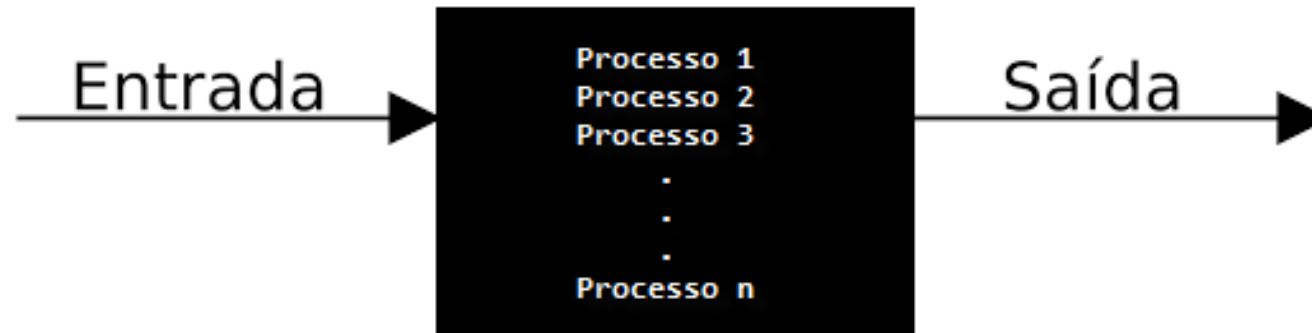
- ✓ Objetivo
- ✓ Recursos necessários – Reservar recursos.
- ✓ Ponto de Partida
 - 1° Passo que aproximam do objetivo final
 - 2° Passo que aproximam do objetivo final
 - 3° Passo que aproximam do objetivo final
 - ...
 - N passo que aproximam do objetivo final (onde N é o número total de passos)
- ✓ Objetivo final alcançado.
- ✓ Liberar recursos.

- ✓ Para o **usuário** do programa as regiões que mais importam são:
 - **entrada** de dados
 - **saída** dos dados
- ✓ Podemos dizer que a parte do processamento é considerada uma **caixa preta**.



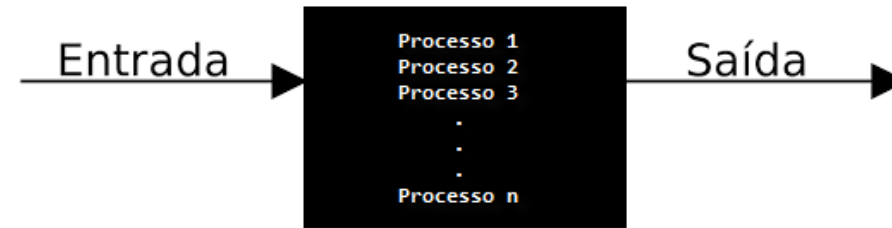
- ✓ No caso do processamento, o **usuário não entende** quais são os processos envolvidos. Apenas que dada uma entrada, ocorre um tipo de saída.
- ✓ Para o usuário, o importante é que dada uma entrada ocorre uma saída.
 - Exemplo: Apresentar o resultado de uma soma em uma calculadora.

- ✓ Para o programador, além de entender qual o tipo de dado de entrada, faz-se necessário entender os **processos de transformação do dado**, e qual a sua respectiva saída.
- ✓ Nesse caso o processamento deve ser entendido pelo programador e todos os seus processos descritos no **código fonte**.

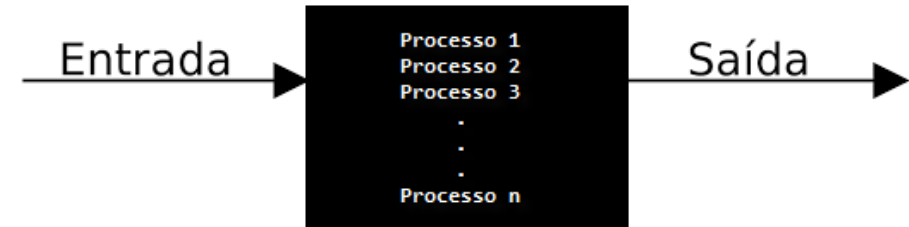


- ✓ É no **processamento** que escrevemos os **comandos** para o computador **realizar**.
 - Exemplo: Função somar dois números (Calculadora).

- ✓ O computador é uma máquina isenta de inteligência. Isto é, ela só entende 0 e 1 (ligado e desligado), ou em outras palavras entende o bit (***B**inary **D**igi**T***).



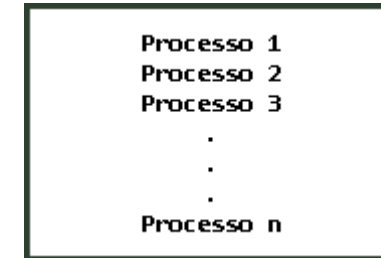
- ✓ Uma sequência de 8 bits é chamada de byte.
- ✓ Para facilitar a programação o homem desenvolveu as linguagens de programação.
- ✓ Já pensou escrever um programa somente em 0's e 1's ???



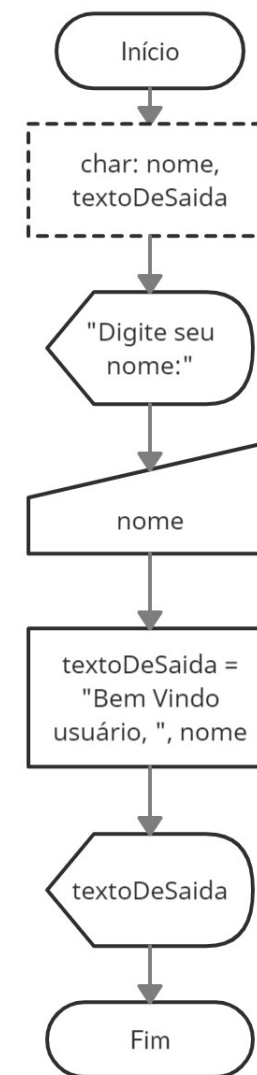
- ✓ A programação pode ser escrita em linguagens de:
 - Baixo nível - são as linguagens que se aproximam da linguagem de máquina.
 - Médio nível - são as linguagens intermediárias, não apresentam a linguagem humana mas também não estão no nível da linguagem de máquina utilizam um compilador para escrever o código em código de máquina.
 - Alto nível – linguagem próxima à da linguagem humana.

- ✓ Exemplo de programa em Alto Nível:

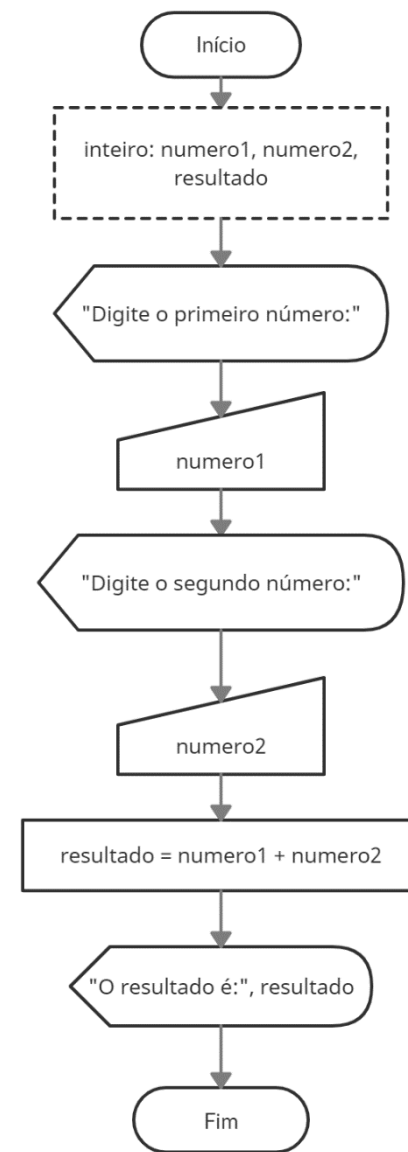
```
# Gerador de senha
tamanhoSenha = 3
def senha(num):
    qtd = len(num)
    if(qtd < tamanhoSenha):
        num = "0"+num
        senha(num)
    else:
        print(num,end="\n")
i = 0
while i < 10**tamanhoSenha:
    senha(str(i))
    i+=1
```



- ✓ **Exemplo:** Escreva o esquema em fluxograma de um algoritmo para apresentar uma mensagem de boas vindas.

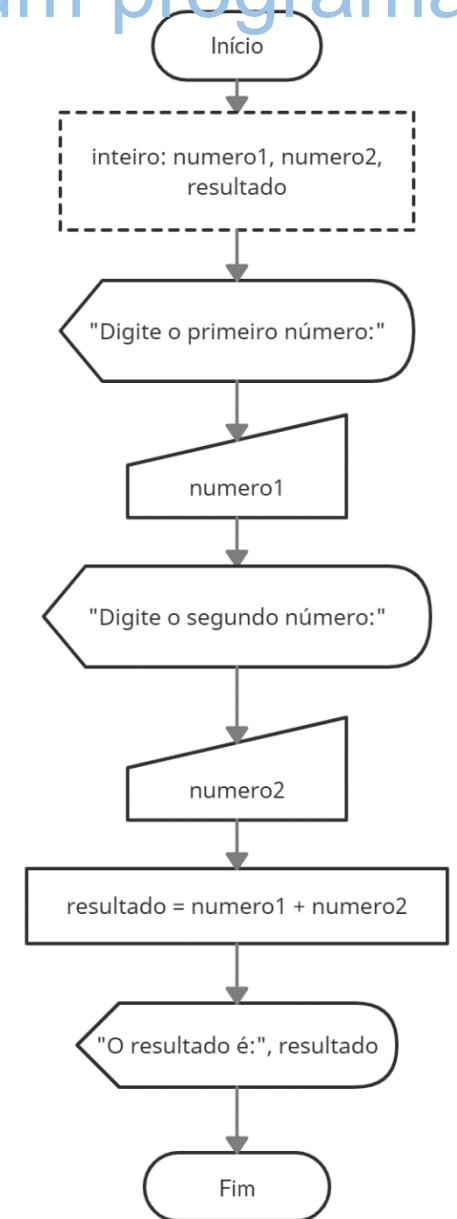


- ✓ **Exemplo:** Escreva o esquema de um algoritmo para somar dois números desenhando o respectivo fluxograma.



- ✓ **Exemplo:** Escreva o esquema de um algoritmo para somar dois números desenhando o respectivo fluxograma.

Testes de Mesa			
Entrada		Processamento	Saída
numero1	numero2	resultado	Tela
2	3	5	O resultado eh:5
101	250	351	O resultado eh: 351
10000	12345	22345	O resultado eh: 22345
A	5	???	???



Exercício 2.1:

Faça um algoritmo que leia o nome de uma pessoa, sua idade e o seu salário e ao final mostre essas informações.

Exercício 2.2:

Faça um algoritmo que leia 2 números e faça as 4 operações matemáticas e mostre esses resultados.

A large teal triangle is positioned on the left side of the slide, pointing towards the bottom-left corner.

*"O futuro depende das ações do
presente, boas ações geram bons
frutos"*



www.unicid.edu.br

R. Cesário Galeno, 475
03071 000
São Paulo SP Brasil
T F 55 11 2178 1212