

# Algoritmos TADS

---

## Algoritmos – Conceitos Iniciais – Parte 1

Professor: Victor Hugo L. Lopes

# *Agenda*

---

- Tópicos de Lógica e lógica de Programação;
- Visão inicial sobre algoritmos;
- Sistemas e a TGS;
- Programação e os paradigmas de Programação;
- Norma ISO 5807:1985

# Lógica

- Inicialmente descrita por Aristóteles, que viveu entre 384 a 322 a.C..
- Diversas definições de lógica...
- “Conjunto de estudos que visam a determinar os processos intelectuais que são condição geral do conhecimento verdadeiro.”
- Lógica é a ferramenta para a obtenção do conhecimento, segundo Aristóteles.

# Lógica

A lógica é a área da filosofia que estuda raciocínios, isto é, como os raciocínios são estruturados. Neste sentido, ela fornece ferramentas que dizem quando se pode chegar a determinadas afirmações com base em outras informações. A lógica, portanto, estuda se é possível transpor de duas ou mais afirmações para uma outra informação conclusiva.

O **silogismo** de Aristóteles prevê o julgamento de no mínimo duas **premissas** para se afirmar algo conclusivo.

# Lógica - SILOGISMO

ex.:

- Todos os ursos da Sibéria são brancos; (1ª premissa)
- Asha é um urso da Sibéria; (2ª premissa)
- Logo, Asha é um urso branco. (conclusão)

Todo silogismo é baseado em um elemento central, no exemplo o urso da Sibéria. Se todos ursos da Sibéria pertencem ao conjunto dos ursos brancos, e se Asha faz parte do conjunto dos ursos da Sibéria, logicamente ela está inserida no conjunto dos ursos brancos.

# Lógica - SILOGISMO

ex.:

- Todo **cachorro** é mortal; (1ª premissa)
- Mequetrefe é um **cachorro**; (2ª premissa)
- Logo, Mequetrefe é mortal. (conclusão)

Note que nos exemplos a conclusão é verdadeira onde as duas premissas foram corretas, senão o raciocínio pode ficar comprometido: **Falácia**.

# Lógica - Falácia

Ex de falácia.:

- Se investirmos milhões, conseguiremos recuperar o meio ambiente; (1ª premissa)
- O meio ambiente foi recuperado; (2ª premissa)
- Logo, investimos milhões. (conclusão)

Não parece verdadeiro?

- Mas e se recuperarmos o meio ambiente com medidas que não necessariamente demandem de grandes investimentos?

.

# Lógica

Necessita de coerência interna e sequência de argumentos irrefutáveis e corretos, que possibilitem a chegada à uma conclusão correta:

- A velocidade da luz é a maior velocidade que existe;
- Se a velocidade de um corpo for superior à velocidade da luz ele viaja no tempo;
- ????





# E a Lógica na programação de Computadores?

# Lógica de Programação

- Como assim?
  - A lógica faz parte do nosso dia-a-dia;
    - » Fazer um bolo, tomar banho....
  - Vendo a lógica na prática:
    - Como exercício, descreva detalhadamente como você chegou no IFG hoje.

# Lógica de Programação

Início da tarefa. Problema: chegar ao IFG.

- Saí do apartamento, tranquei a porta, abri o portão eletrônico, destravei o carro, entrei no carro, liguei o carro, comecei a dirigir.
- Liguei a seta e virei à esquerda, segui em frente, liguei a seta e virei à direita ... segui em frente, liguei a seta e contornei a rotatória à esquerda, segui em frente, liguei a seta e me preparei para virar à direita.
- Aguardei a abertura do portão, entrei com o carro, estacionei, desliguei o carro, desci do carro, tranquei o carro e me dirigi à sala dos professores.

Fim da tarefa. Resultado: professor no destino correto.

# Lógica de Programação

- De agora em diante, façamos uma pausa nas questões computacionais aqui em andamento para entendermos um ponto básico:

Muito se fala em Sistemas:

- “O sistema está fora do ar”;
- “É culpa do sistema”;
- “Sistema respiratório humano”;
- ....
- Afinal, o que é um sistema?

# Sistemas e a TGS

O biólogo alemão Ludwig Von Bertalanffy começou a elaborar os primeiros traços de uma teoria, chamada mais tarde de Teoria Geral de Sistemas, em 1924.

A TGS foi concebida como uma teoria interdisciplinar, capaz de transcender aos problemas tecnológicos de cada uma delas e dispor de princípios gerais, bem como de modelos, também gerais, de tal forma que todas as ciências pudessem interligar as descobertas de todas de forma ampla e total.

# Sistemas e a TGS

Um **sistema** (do grego *sietemiun*), é um conjunto de elementos interconectados, de modo a formar um todo organizado.

É uma definição que acontece em várias disciplinas, como biologia, medicina, informática, administração.

Vindo do grego, o termo "sistema" significa "combinar", "ajustar", "formar um conjunto".

# Sistemas e a TGS

## Características de um SISTEMA:

- Todo sistema possui um objetivo geral a ser atingido;
- É um conjunto de órgãos funcionais, componentes, entidades, partes ou elementos interconectados;
- A integração entre os órgãos se dá por fluxo de informações, de matéria, de sangue, de energia.

# Sistemas e a TGS

A boa integração dos elementos componentes do sistema é chamada **sinergia**, determinando que as transformações ocorridas em uma das partes influenciarão todas as outras.

A alta sinergia de um sistema faz com que seja possível a este cumprir sua finalidade e atingir seu objetivo geral com eficiência; por outro lado se houver falta de sinergia, pode implicar em mau funcionamento do sistema, vindo a causar inclusive falha completa, morte, falência, pane, queda do sistema etc.



# Sistemas e a TGS

Tipos de sistemas:

Quanto à sua constituição:

- Físicos ou concretos;
- Abstratos ou conceituais;

Quanto à sua natureza:

- Fechados;
- Abertos;

# Sistemas e a TGS

Elementos de sistemas:

- Entrada ou insumo;
- Processo;
- Saída;


# Sistemas e a TGS

Hierarquia de sistemas:

- Sistema Total;
- Subsistema.

Exemplo:

- Sistema econômico mundial (ST);
  - Sistema econômico nacional (SS)(ST);
    - Sistema econômico municipal (SS)(ST);
      - » Rede bancária municipal (SS);



# Retornando aos Sistemas Computacionais

# Lógica de Programação

- A lógica de Programação é necessária para pessoas que desejam trabalhar com desenvolvimento de sistemas e programas. Ela permite definir a sequência lógica para o desenvolvimento.
- Lógica de programação é a técnica de encadear pensamentos para atingir determinado objetivo computacionais.

# Lógica de Programação

- Relembrando:
  - Programa é uma sequência de **instruções** codificadas em uma linguagem de programação que será interpretado pela máquina para a resolução de um problema computacional.

# Lógica de Programação

## Instruções

- Um conjunto de regras ou normas definidas para a realização ou emprego de algo;
- Em informática, é a informação que indica a um computador uma ação elementar a executar;
- Porém, uma ordem isolada não permite realizar o processo completo, para isso é necessário um conjunto de instruções colocadas em ordem sequencial lógica.

# Lógica de Programação

- Sistemas com vários módulos
  - Módulo de cadastro (cliente, funcionário, produtos);
  - Módulo de vendas;
  - Módulo de controle de estoque;
  - Módulo financeiro;
  - Módulo de Recursos Humanos (RH);
  - ...
- Para desenvolver um sistema: Análise de Sistemas!
- Projeto Lógico de um sistema!

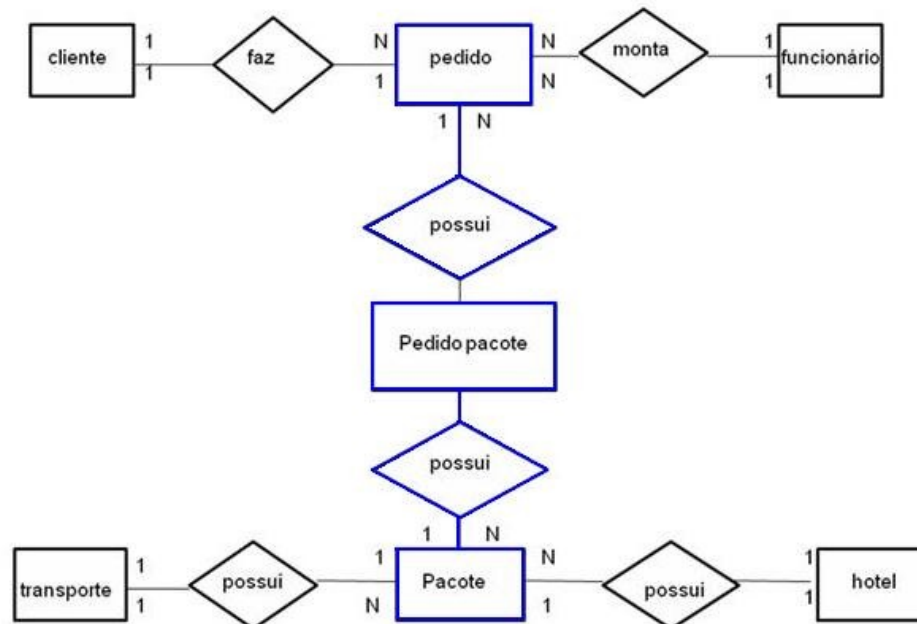


# Lógica de Programação

- Técnicas de Programação (método sistemático):
  - Análise do Problema;
  - Projeto do programa;
  - Implementação;
  - Testes;
  - Verificação e implantação.
- Processo de Software-> Engenharia de Software.

# Lógica de Programação

- Projeto lógico:
  - Ferramentas gráficas;
  - Ferramentas textuais;



# Lógica de Programação

- Projeto lógico – Ferramentas gráficas
  - Diagramas de Blocos (fluxogramas?!);
  - Diagramas de Quadros (Chapin);
  - Demonstrar de forma concreta a linha de raciocínio lógico utilizada para desenvolver um programa (elemento abstrato).
  - Concede facilidade em uma implementação posterior (transposição para linguagem de programação de alto nível);
  - Não diz respeito a como fazer, mas ao quê fazer!
  - Analogia com planta de uma edificação!

# Lógica de Programação

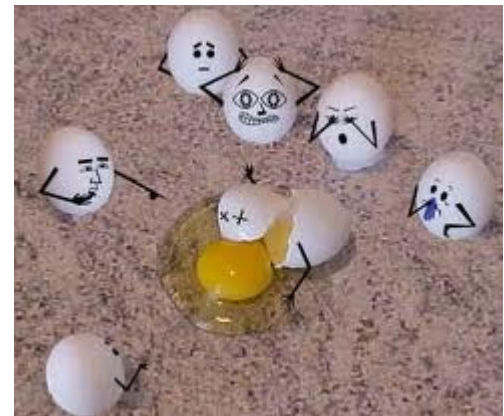
- Projeto lógico – Ferramentas textuais
  - Pseudocódigos ou metalinguagens;
  - Descrever de forma simples programas:
    - Linguagem natural;
    - Sem rigor técnico de linguagem de programação;
  - Técnica: PDL (Program Design Language);
    - Linguagem de Projeto de Programação e não Linguagem de programação!
  - Descrição Narrativa, Português estruturado ou Portugol.

# Lógica de Programação

- Descrição Narrativa:
  - Analisa o enunciado do problema e escreve, utilizando linguagem natural, os passos a serem seguidos para sua resolução.
  - **Vantagem:** conhece uma linguagem como o português?!
  - **Desvantagem:** ambiguidades e mau entendimento.

# Lógica de Programação

- Descrição Narrativa:
  - Ex.:
    - Problema: Ordenar a ida ao supermercado para a compra de ovos e leite.
    - “Pegue o dinheiro em cima da geladeira e vá até o supermercado da esquina. Compre 6 ovos, se tiver leite, traga dois. Se sobrar troco, traga farinha de trigo.”
  - Resultado?!?!?

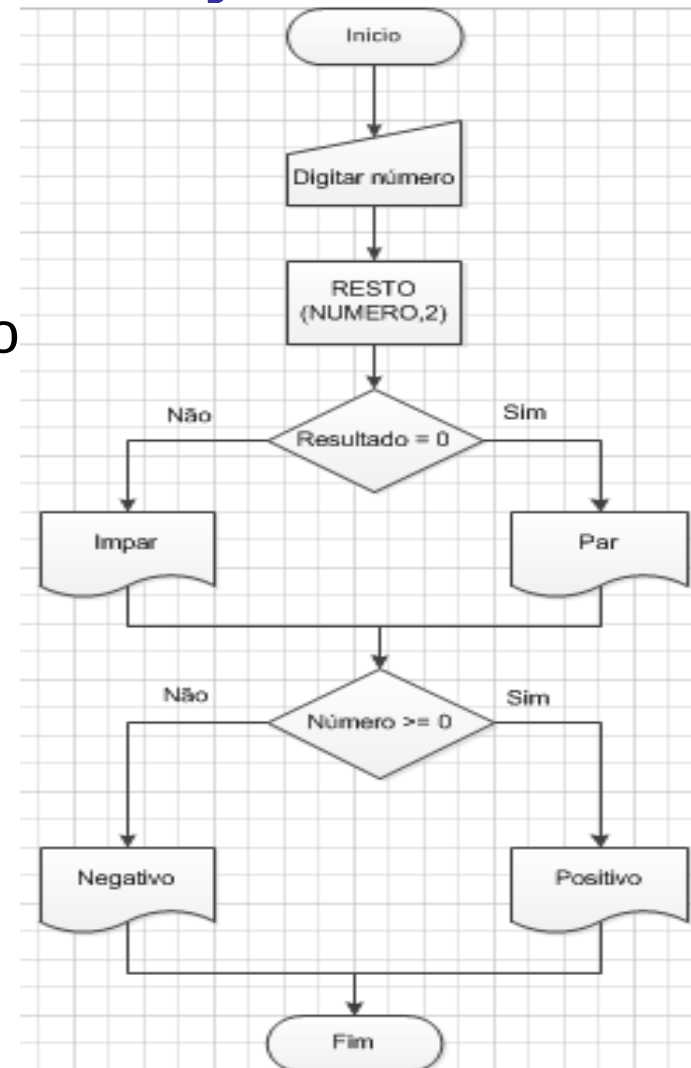


# Lógica de Programação

- Fluxograma/diagrama de blocos:
  - Analiza o enunciado do problema e escreve utilizando símbolos gráficos definindo os passos a serem seguidos.
  - Vantagem: Menor ambiguidade e maior facilidade de entendimento para o programador;
  - Desvantagem: muitos elementos gráficos e simbologia.

# Lógica de Programação

- Fluxograma/diagrama de blocos:
  - Problema: definir se um Número informado pelo usuário É par ou ímpar, positivo ou negativo.





# Lógica de Programação

- Pseudocódigo, português:
  - Analisa o enunciado do problema e escreve, por meio de regras definidas, os passos a serem seguidos para a solução.
  - Vantagem**: Linguagem muito próxima de uma linguagem de programação;
  - Desvantagem**: complexidade das regras do pseudocódigo.

# Lógica de Programação

- Pseudocódigo, portugol:  
--Ler do usuário 2 números e demonstrar o resultado da multiplicação dos dois.

ALGORITMO

DECLARE N1, N2, M NUMÉRICO

ESCREVA “DIGITE DOIS NÚMEROS”

LEIA N1, N2

M <- N1 \* N2

ESCREVA “MULTIPLICAÇÃO=” , M

FIM\_ALGORITMO

```
int main() {  
    int N1, N2, M;  
    scanf("%d%d",&N1,&N2);  
    M = N1 * N2;  
    printf("Multiplicacao = %d",  
M);  
    return (EXIT_SUCCESS);  
}
```

# Lógica de Programação

- Paradigmas de programação:
  - Estruturado;
  - Orientado à Objetos;
  - Lógico;
  - Funcional...
- Utilizaremos o paradigma estruturado!
  - O pensar humano é estruturado!
  - Bases para O. O.

Dica: pesquise sobre os paradigmas de programação.

# Lógica de Programação

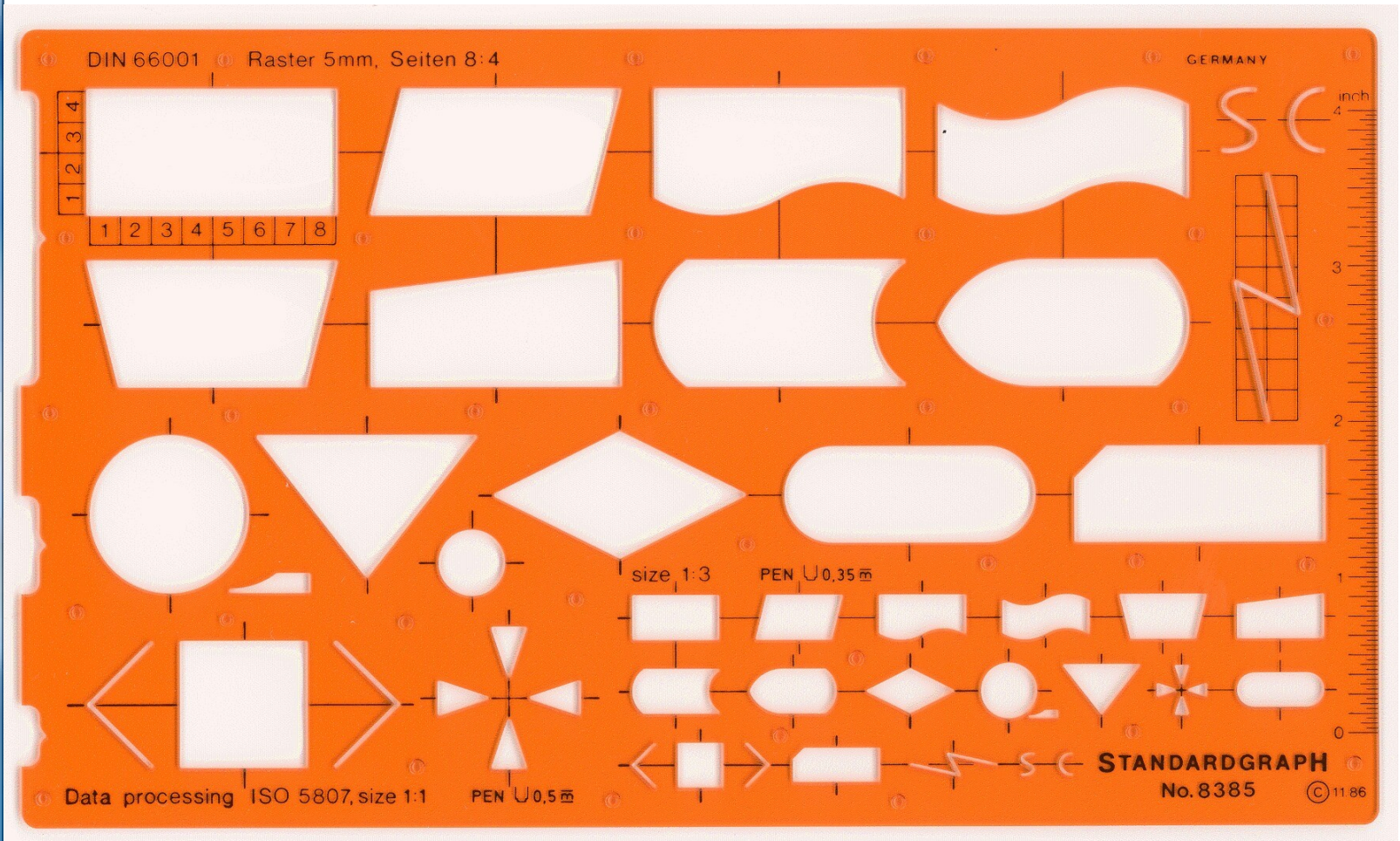
- Programação de computadores
  - Condicionar o pensamento humano às limitações técnicas do computador;
  - O que os computadores podem fazer?
- Analista de Sistemas → Arquiteto;
  - Nível Macro!
  - Planta do sistema.
- Programador?
  - Pedreiro altamente especializado!(mestre de obras, pedreiro e servente de pedreiro)
  - Nível Micro!

# Norma ISO\* 5807:1985

- Diagrama de Blocos (projeto) → Programação (implementação no computador);
- ISO 5807 = ISO 1028 + ISO 2636 (1973);
- Representação gráfica do comportamento do computador;
- Símbolos gráficos para demonstrar a linha de raciocínio do programador;



\*Organização Internacional para Padronização

# Norma ISO 5807:1985









# Norma ISO 5807:1985 – Principais Símbolos


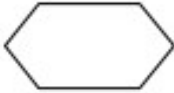


Símbolo	Significado	Descrição
	Terminal <i>Terminator</i>	Este símbolo representa a definição de início e fim do fluxo lógico de um programa (ISO 5807, 1985, p. 9). Também é utilizado na definição de sub-rotinas de procedimento ou função.
	Entrada manual <i>Manual input</i>	Este símbolo representa a entrada manual de dados, normalmente efetuada em um teclado conectado diretamente ao console do computador (ISO 5807, 1985, p. 3).

# Norma ISO 5807:1985 – Principais Símbolos

Símbolo	Significado	Descrição
	Processamento <i>Process</i>	Este símbolo representa a execução de uma operação ou grupo de operações que estabelecem o resultado de uma operação lógica ou matemática (ISO 5807, 1985, p. 3).
	Exibição <i>Display</i>	Este símbolo representa a execução da operação de saída visual de dados em um monitor de vídeo conectado ao console do computador (ISO 5807, 1985, p. 3).
	Documento <i>Document</i>	Este símbolo representa a execução da operação de saída de dados em um documento emitido por uma impressora na forma de relatório (ISO 5807, 1985, p. 3).
	Dados <i>Data</i>	Este símbolo representa oficialmente dados de uma forma genérica (ISO 5807, 1985, p. 2). Tipicamente é associado a operações genéricas de entrada e saída de dados, desde que identificados.



# Norma ISO 5807:1985 – Principais Símbolos

Símbolo	Significado	Descrição
	Decisão <i>Decision</i>	Este símbolo representa o uso de desvios condicionais para outros pontos do programa de acordo com situações variáveis (ISO 5807, 1985, p. 4).
	Preparação <i>Preparation</i>	Este símbolo representa a modificação de instruções ou grupo de instruções existentes em relação à ação de sua atividade subsequencial (ISO 5807, 1985, p. 4).
	Processo predefinido <i>Predefined process</i>	Este símbolo representa definição de um grupo de operações estabelecidas como uma sub-rotina de processamento anexa ao diagrama de blocos (ISO 5807, 1985, p. 4).
	Conector <i>Connector</i>	Este símbolo representa a entrada ou saída em outra parte do diagrama de blocos. Pode ser usado na definição de quebras de linha e na continuação da execução de decisões (ISO 5807, 1985, p. 9).

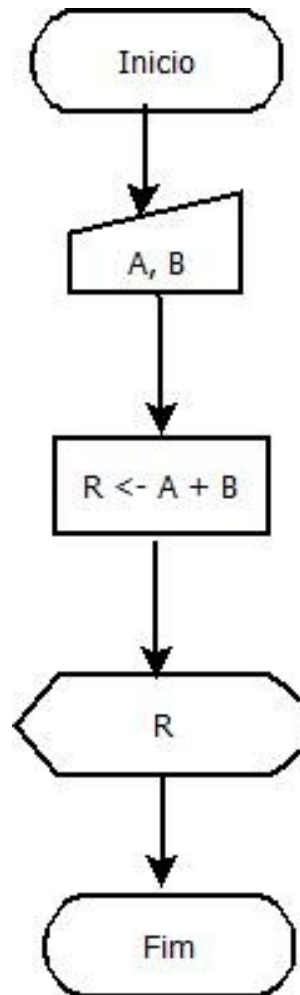
# Norma ISO 5807:1985 – Principais Símbolos

Símbolo	Significado	Descrição
-----	Linha tracejada <i>Dashed line</i>	Este símbolo representa uma alternativa na definição do relacionamento entre duas operações do diagrama de blocos. Também pode ser usado na definição de área de comentários (ISO 5807, 1985, p. 7).
—————	Linha <i>Line</i>	Este símbolo representa a ação de vínculo existente entre os vários símbolos de um diagrama de blocos. Normalmente possui a ponta de uma seta indicando a direção do fluxo de ação (ISO 5807, 1985, p. 6).

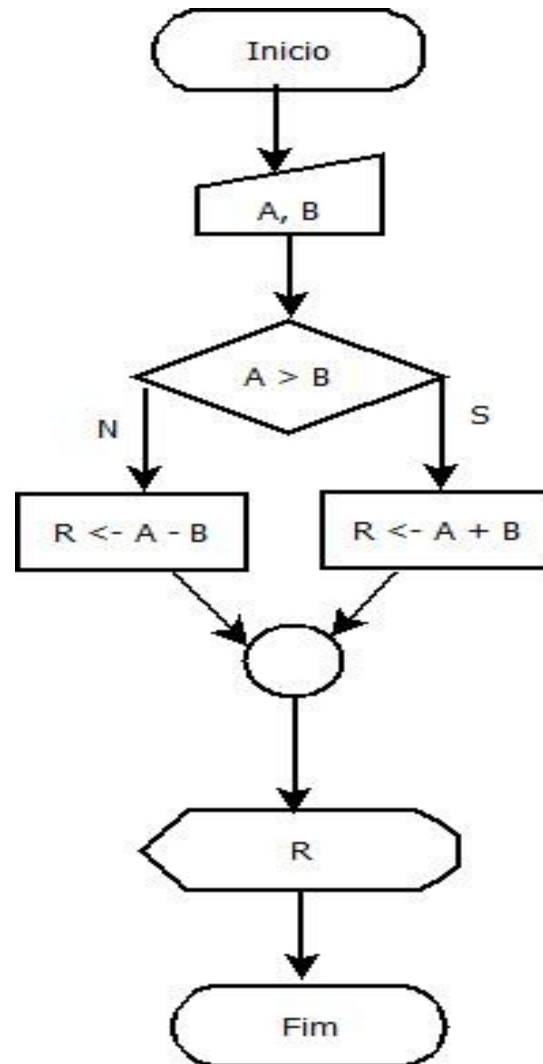
# Norma ISO 5807:1985

- Regras para o uso dos símbolos
  - Podem ser utilizados rótulos;
  - Os símbolos devem ser conectados uns aos outros por setas que ilustrem o fluxo de execução do programa;
  - Orientação: de cima para baixo, da esquerda para a direita;
  - Operação de entrada de dados: símbolo “entrada manual”;
  - Operação de saída de dados: símbolo “exibição”;
  - Processamento matemático e lógico: símbolo “processamento”;
  - Tomadas de decisão: símbolo “decisão” (condição lógica);
  - Repetições, “laços” ou loops: símbolo de “preparação”; 43

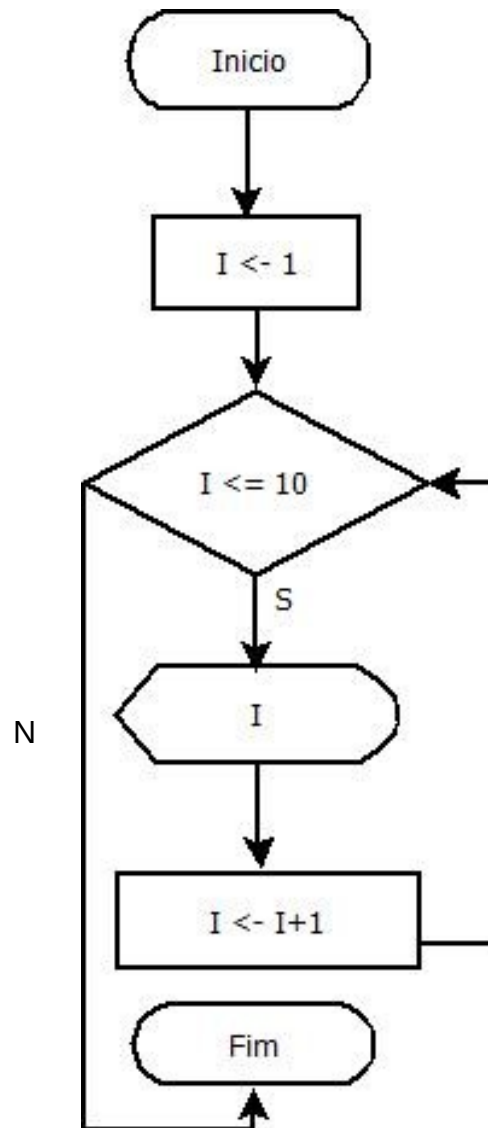
# Diagramas de Bloco - Exemplos



# Diagramas de Bloco - Exemplos



# Diagramas de Bloco - Exemplos



# Lógica de Programação

- Projeto lógico – Ferramentas textuais
  - Técnica: PDL (Program Design Language);
  - Português estruturado ou Portugol;
  - Comandos em idioma local;
  - Caracteres alfanuméricos e parênteses;
  - Escrever programas sem se preocupar com a linguagem de implementação;
  - Verbos no imperativo!
  - Problema: Falta de padronização entre profissionais!

# Compiladores, Interpretadores e Tradutores

- Ferramentas de tradução entre linguagens: Pascal → C;
- Interpretadores → Executam linha à linha o código fonte, em tempo real;
- Compiladores
  - Transformam código fonte de alto nível em linguagem de máquina!
  - Dica: Pesquise sobre compiladores e interpretadores.