



Programação de Computadores

Algoritmos

Professor: Roberto Rocha

Algoritmos

Sequência de passos finitos com o objetivo de solucionar um problema

Um algoritmo é uma descrição de um padrão de comportamento (o que fazer) expressa em termos de um repertório finito e bem inteligível de ações primitivas (como fazer), as quais, supõe-se, *a priori*, sejam possíveis de se executar.

Descreva seu procedimento em 5 passos como chegar a UNIVERSIDADE

Refine uma das etapas em outras 5 atividades

Refine uma das etapas anterior em outras 5 atividades

Quando parar?

Algoritmos

Sequência de passos finitos com o objetivo de solucionar um problema

O enigma é: você tem uma raposa, uma galinha e um milho.

- Você não pode deixar a galinha com o milho.
- Você não pode deixar a raposa com a galinha.
- Você tem que atravessar um rio em um barco. Este barco só pode levar você e um dos três.

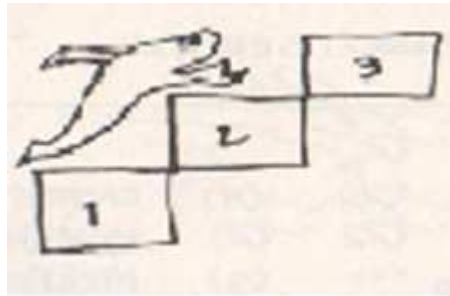
Em quantos passos você consegue?

A ordem que você tomá-los em toda é a seguinte:

- 1) Atravessar com a galinha,
- 2) voltar vazio,
- 3) Levar raposa,
- 4) Trazer de volta a galinha,
- 5) Levar o milho,
- 6) Votar sozinho,
- 7) Atravessar com a galinha.

Algoritmos

Um algoritmo é simplesmente um procedimento bem definido, passo a passo:
Um receita de bolo, se você quiser!



Passo a Passo

Significando que cada passo é completado antes que o próximo se inicie.

Bem Definido:

Significando que cada passo é completamente definido a partir da entrada atual e dos passos anteriores – não são permitidas ambiguidades!



Algoritmos

EXEMPLOS DE ALGORITMOS:



↳ ISTO É UM ALGORITMO PORQUE VOCÊ SEMPRE SABE O QUE FAZER:

1. VERIFIQUE SE AS OGIVAS ESTÃO CAINDO
2. SE SIM, DEITE-SE E CURTA!
3. SENÃO, VÁ PARA O SERVIÇO.



Algoritmos

EXEMPLOS NÃO-ALGORÍTMICOS

"SE AS OBIVAS
NUCLEARES ESTIVEREM
CAINDO COMO CHUVA
DE PEDRA, DEITE-SE
E TENTE CURTIR!"



NÃO HA' MENÇÃO SOBRE O QUE FAZER QUANDO NÃO ESTÃO
CAINDO BOMBAS... PORTANTO, NÃO ESTA' BEM DEFINIDO.

Algoritmos

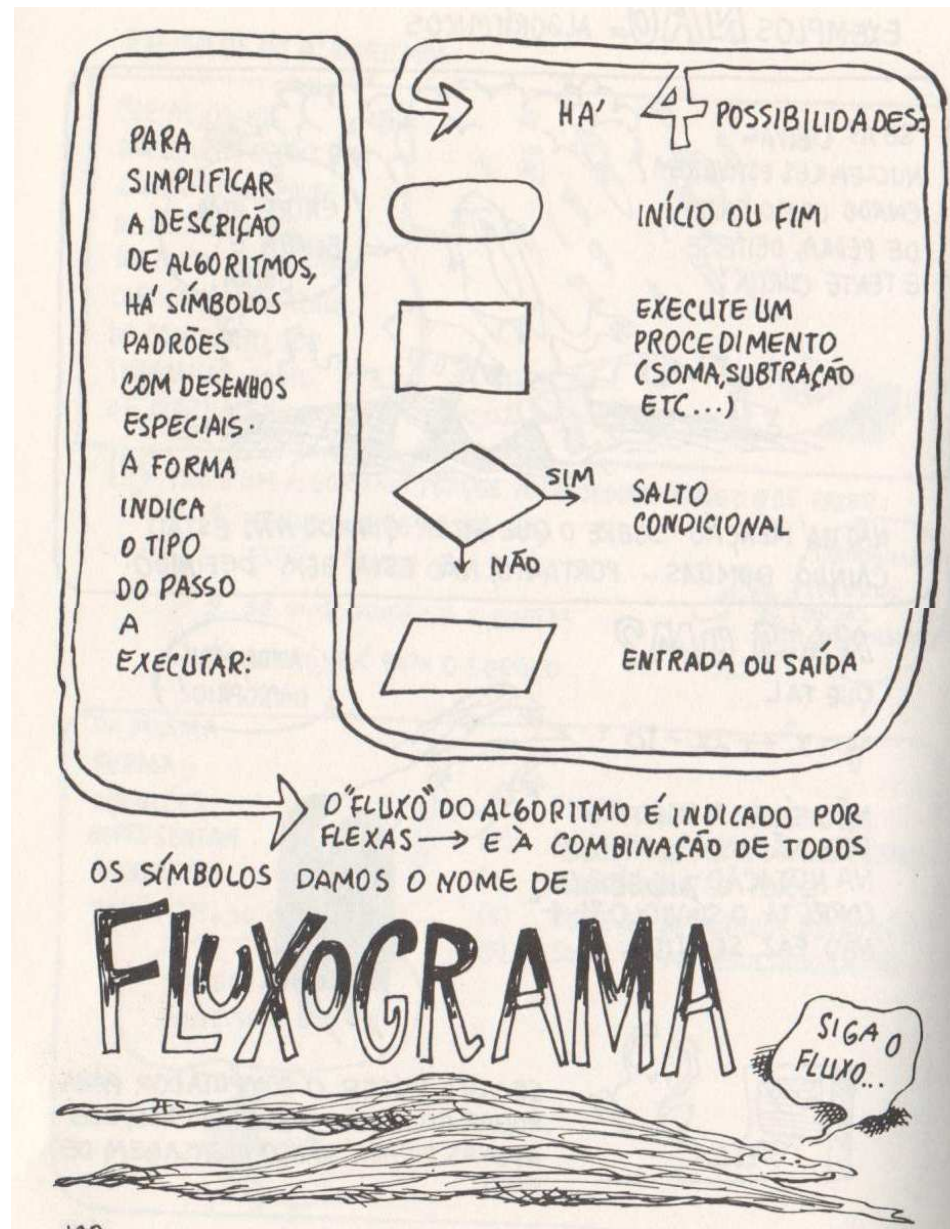


Algoritmos



Se você puser o computador para rodar algo não algoritmo, ele apenas ficará dando mensagem de erro!!!

Algoritmos - Fluxograma



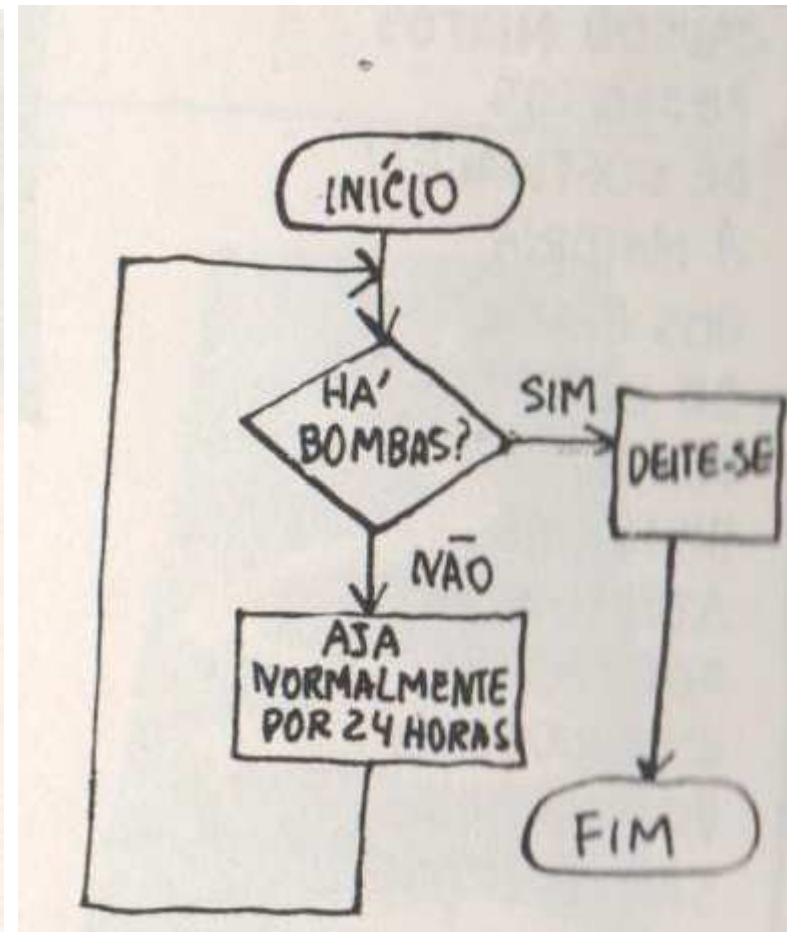
Algoritmos - Fluxograma



Algoritmos

TAMBÉM É POSSÍVEL, NO FLUXO DE ALGORITMOS, UM SALTO PARA A FRENTE OU PARA TRÁS. POR EXEMPLO, VAMOS REESCREVER O PRIMEIRO ALGORITMO:

1. SE ESTIVEREM CAINDO BOMBAS, VÁ PARA O PASSO 2. CASO CONTRÁRIO, VÁ PARA O PASSO 4.
2. DEITE-SE E CURTA!
3. VÁ PARA O PASSO 6.
4. LEVE UMA VIDA NORMAL POR 24 HORAS.
5. VÁ PARA O PASSO 1.



Algoritmos

Você poderá achar o FLUXOGRAMA mais fácil de “pegar” do que o “programa” escrito. Note que ele pode continuar indefinidamente!!!



Porém os FLUXOGRAMAS ajudam a descrever algoritmos simples e descrever algoritmos é o objetivo da PROGRAMAÇÃO DO COMPUTADOR!!

Algoritmos

O NÃO PENSAR
ALGORITMICAMENTE
CAUSOU MUITOS
PESADELOS
DE SOFTWARE!!
A MAIORIA
DOS PROJETISTAS
DE SOFTWARE
CONTA
HISTÓRIAS
ATERROIZANTES
SOBRE
USUÁRIOS
QUE NÃO
SABIAM O QUE
QUERIAM
COM
EXATIDÃO!!



Algoritmos

O primeiro passo para se escrever qualquer programa é a ANÁLISE da tarefa a ser feita deve-se descobrir como fazê-la algoritmicamente!!

Mais alguns exemplos ... Um pouco mais próximos de como comandar o computador fazer

“Contas de colegas de quarto”



Elisa e Sofia, que moram juntas, dividiam as despesas, ambas compravam alimentos e guardavam os recibos. No fim do mês, queriam saber quem devia para quem.

Algoritmos

PARA AS CONTAS DE COLEGAS
DE QUARTO, RACIOCINAMOS
NA FORMA:

SEJA S = DESPESAS DE SOFIA
 E = DESPESAS DE ELISA

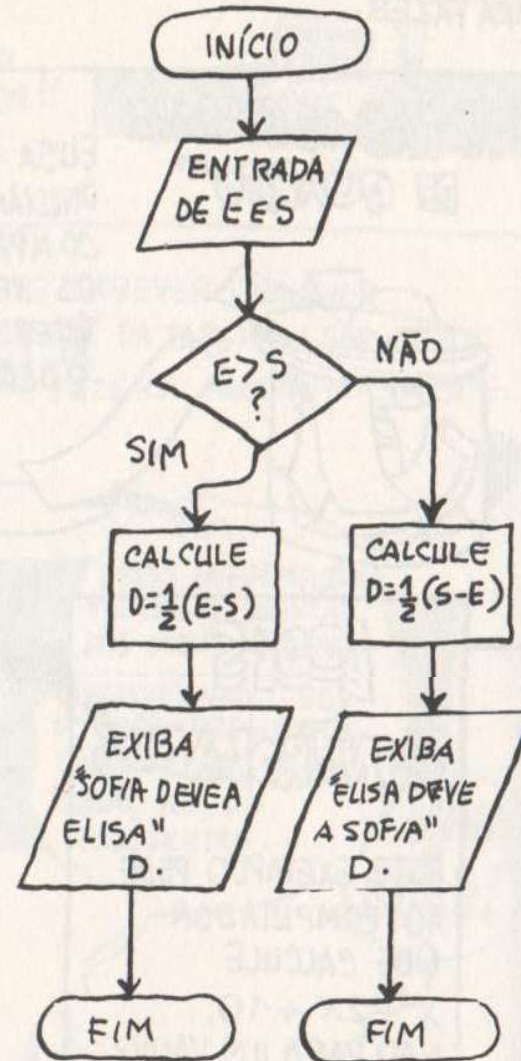
ENTÃO A DESPESA TOTAL É
 $S + E$ E CADA UMA DELAS
DEVERIA PAGAR

$$\frac{1}{2}(S + E).$$

SE ELISA GASTOU MAIS,
ENTÃO $E > S$ * E ,
PORTANTO, SOFIA DEVE A
ELISA $\frac{1}{2}(S + E) - S$, OU
 $\frac{1}{2}(E - S)$.

CASO CONTRÁRIO (QUANDO
 $S \geq E$ *), ELISA DEVE A SOFIA
 $\frac{1}{2}(S - E)$.

A SAÍDA DO ALGORITMO DEVE
DIZER QUEM DEVE E
QUANTO DEVE.



* $>$ QUER DIZER "É MAIOR DO QUE";
 \leq QUER DIZER "É MENOR OU IGUAL".

\geq QUER DIZER "É MAIOR OU IGUAL";
 $<$ QUER DIZER "É MENOR DO QUE".

Algoritmo:

Forbellone, 1999:

“Algoritmo é uma sequencia de passos que visa atingir um objetivo bem definido.”

Ascencio, 1999:

“Algoritmo é a descrição de uma sequencia de passos que deve ser seguida para a realização de uma tarefa.”

Salvetti, 1999:

“Algoritmo é uma sequencia finita de instruções ou operações cuja execução, em tempo finito, resolve um problema computacional, qualquer que seja sua instância.”

Manzano, 1997:

“Algoritmo são regras formais para obtenção de um resultado ou da solução de um problema, englobando formulas de expressões aritméticas.”

Exemplos:

Algoritmo 1 — Somar três números

- Passo 1 — Receber os três números.
- Passo 2 — Somar os três números.
- Passo 3 — Mostrar o resultado obtido.

Algoritmo 2 — Fazer um sanduíche

- Passo 1 — Pegar o pão.
- Passo 2 — Cortar o pão ao meio.
- Passo 3 — Pegar a maionese.
- Passo 4 — Passar a maionese no pão.
- Passo 5 — Pegar e cortar alface e tomate.
- Passo 6 — Colocar alface e tomate no pão.
- Passo 7 — Pegar o hambúrguer.
- Passo 8 — Fritar o hambúrguer.
- Passo 9 — Colocar o hambúrguer no pão.

Exemplos:

- Algoritmo 3** — Trocar uma lâmpada
- Passo 1 — Pegar uma lâmpada nova.
 - Passo 2 — Pegar uma escada.
 - Passo 3 — Posicionar a escada embaixo da lâmpada queimada.
 - Passo 4 — Subir na escada com a lâmpada nova na mão.
 - Passo 5 — Retirar a lâmpada queimada.
 - Passo 6 — Colocar a lâmpada nova.
 - Passo 7 — Descer da escada.
 - Passo 8 — Testar o interruptor.
 - Passo 9 — Guardar a escada.
 - Passo 10 — Jogar a lâmpada velha no lixo.



Exercícios:

- 1- Faça um algoritmo para os passos necessários para sacar dinheiro no caixa eletrônico
- 2 – Multiplicar 2 números
- 3 – Calcular a média aritmética de 3 números
- 4 – Faça um algoritmo para calcular o novo salário de um funcionário. A regra será a seguinte para os funcionários que recebem até \$5.000,00 o acréscimo será de 30% e para os demais será de 10%.
- 5 – Para os exercícios 2,3 e 4 indique quais dados serão de entrada e o que se espera que seja a saída.

Algoritmos

⚡ LOOPS CONTROLADOS

ESTE EXEMPLO PEDE
AO COMPUTADOR
QUE CALCULE
 $x^2 + 2x + 10$,
NÃO PARA UM VALOR
DE x , MAS PARA MUITOS
VALORES, A SABER
 $x = 0; 0,1; 0,2; 0,3; \dots$ E
ASSIM POR DIANTE...
ATE' 2,0.

EM "LOOPS CONTROLADOS", QUEREMOS AVALIAR UMA EXPRESSÃO,
 $x^2 + 2x + 10$, REPETITIVAMENTE, PARA VALORES DIFERENTES DE
 x (A SABER, $0,0; 0,1; 0,2; \dots; 1,9; 2,0$).

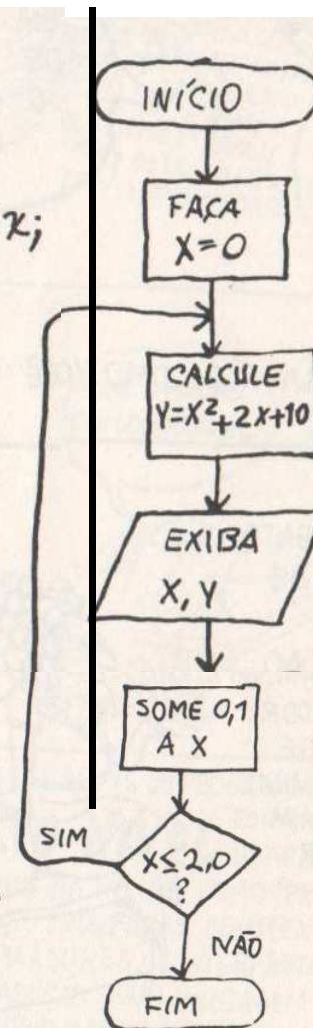
Algoritmos

O NÚCLEO DO ALGORITMO
É O LOOP:

1. CALCULE $x^2 + 2x + 10$
PARA O VALOR CORRENTE DE x ;
2. EXIBA O RESULTADO;
3. PEGUE O PRÓXIMO x ;
4. VOLTE AO PASSO 1.

TEMOS QUE DAR, TAMBÉM,
O VALOR INICIAL PARA x ,
O VALOR FINAL, O PONTO
DE PARADA E A FORMA
DE CALCULAR O
"PRÓXIMO x ".

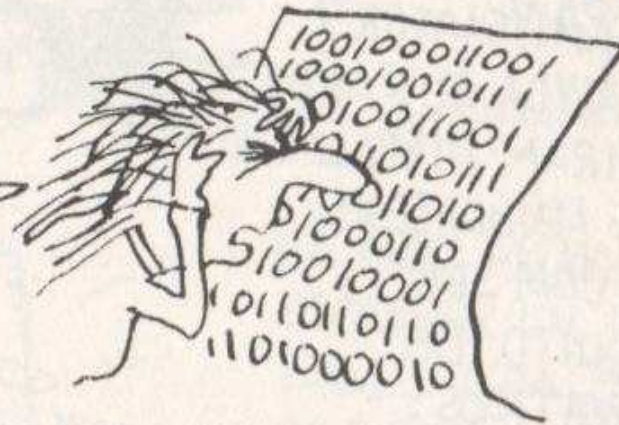
OBSERVE QUE O
FLUXOGRAMA
RETORNA, PEGANDO
VALORES SUCESSIVOS
DE x , ATÉ QUE ESTE
EXCEDA 2.



Algoritmos

NOS PRIMÓRDIOS DA COMPUTAÇÃO OS PROGRAMADORES ESCRIVIAM DIRETO EM "LINGUAGEM DE MÁQUINA" - CÓDIGO BINÁRIO - O QUE DAVA OBVIAMENTE GRANDES DORES DE CABEÇA!

PRECISAMOS
DE UM
COMPUTADOR
SÓ PARA CONTROLAR
A CONTA DA
ASPIRINA!



Algoritmos



Suponha as seguintes instruções para um computador

Mnemônico	Código	Significado
nop	0000 0000	passar adiante, não fazer nada
out	0000 0001	copiar o conteúdo do acumulador para o registrador de saída
inc	0010 0000	incrementar de uma unidade o valor no acumulador
dec	0011 0000	incrementar de uma unidade o valor no acumulador
load [xx]	0100 xxxx	carregar o acumulador com o conteúdo da memória na posição xxxx
store xx	0101 xxxx	carregar o conteúdo do acumulador na memória na posição xxxx
add [xx]	0110 xxxx	carregar o registrador B com o conteúdo da memória na posição xxxx, e somá-lo ao acumulador
sub [xx]	0111 xxxx	carregar o registrador B com o conteúdo da memória na posição xxxx, e subtrai-lo do acumulador
not	1000 0000	complementar o conteúdo do acumulador
or	1001 0000	disjunção do acumulador com o conteúdo da memória na posição xxxx
and	1010 0000	conjunção do acumulador com o conteúdo da memória na posição xxxx
xor	1011 0000	disjunção exclusiva do acumulador com o conteúdo da memória na posição xxxx
jump xx	1100 xxxx	carregar o apontador de instrução como endereço da posição xxxx e executar a instrução que ali estiver
jumpS xx	1101 xxxx	testar se o sinal do acumulador é negativo; se for, saltar para a instrução na posição xxxx; senão, executar a próxima instrução
jumpZ xx	1110 xxxx	testar se o acumulador é igual a zero; se for, saltar para a instrução na posição xxxx; senão, executar a próxima instrução
halt	1111 1111	parar a execução

Algoritmos

Escrever um programa capaz de calcular:
 $10 - 4 + 6 + 3$.

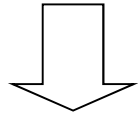
Mapeamento das ações na memória:

	Endereços	Conteúdo	Significado	Descrição
00	0000	0100 1100	load [12]	guardar 10
01	0001	0111 1101	sub [13]	subtrair 4
02	0010	0110 1110	add [14]	somar 6
03	0011	0110 1111	add [15]	somar 3
04	0100	0000 0001	out	mostrar resultado
05	0101	1111 1111	halt	parar
06	0110	0000 0000	nop	
07	0111	0000 0000	nop	
08	1000	0000 0000	nop	
09	1001	0000 0000	nop	
10	1010	0000 0000	nop	
11	1011	0000 0000	nop	
12	1100	0000 1010	[10]	valor = (10)
13	1101	0000 0100	[04]	valor = (04)
14	1110	0000 0110	[06]	valor = (06)
15	1111	0000 0011	[03]	valor = (03)

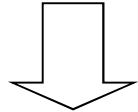
Uso de Linguagens Assembler

Desvantagens:

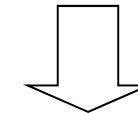
Programas com algum grau de dificuldade :



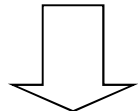
Número elevado de instruções



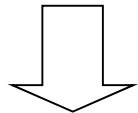
Grande o tempo de Codificação e testes



Dificuldade para examinar o programa e fazer modificações ou extensões



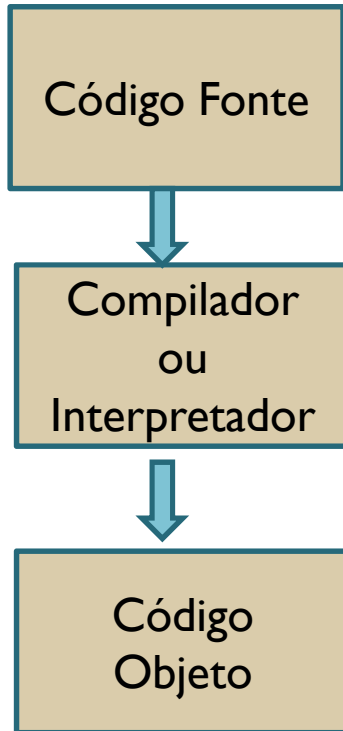
Dificuldade de entender a lógica do programa



Impossibilidade para transferir um programa de uma máquina para outra

Algoritmos

Finalmente:



As linguagens de programação de ALTO NÍVEL foram inventadas. Usaram palavras comuns do inglês, como “PRINT”, “READ”, “DO”.

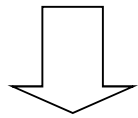
Programas complexos, chamados compiladores ou interpretadores, fazem a tradução para linguagem de máquina.

Os programas em linguagem de ALTO NÍVEL muitas vezes recebem o nome de “PROGRAMAS FONTE” e os convertidos em linguagem de máquina de “PROGRAMAS OBJETO”.

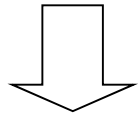
Linguagem de Alto Nível

Vantagens:

Tem um vocabulário que é similar à linguagem natural



Permite que se escrevem programas menores



É possível implementá-las em vários computadores

Algoritmos

A PRIMEIRA LINGUAGEM DE ALTO NÍVEL FOI **FORTRAN** ("FORMULA TRANSLATOR"), QUE ESTREOU AINDA NOS ANOS 50. DEPOIS, SURTIRAM LITERALMENTE CENTENAS DE LINGUAGENS, CADA QUAL COM SEU PRÓPRIO SEQUITO DE DEVOTOS FANÁTICOS!



Exemplo de programa:



O mesmo programa codificado em C++

```
#include <iostream>
using namespace std;
int main()
{
    int Soma=0, Valor;
    cout << "\nDigite um Valor:";
    cin >> Valor;
    while (Valor != 0)
    {
        Soma = Soma+Valor;
        cout << "\nDigite um Valor:";
        cin >> Valor;
    }
    cout << '\n' << "Valor da soma: " << Soma;
    cout << endl;
    system("PAUSE");
    return 0;
}
```



O mesmo programa codificado em
Fortran

C Este programa emite a soma de um
C conjunto de N valores fornecidos em
C cartões

```
INTEGER VALOR, SOMA
SOMA = 0
2  READ VALOR
   IF (VALOR .EQ. 0) GOTO 5
   SOMA = SOMA + VALOR
   GO TO 2
5  PRINT SOMA
END
```

Outras Linguagens de Alto Nível

PL/1 – Program Language 1

```
PSOMA: PROCEDURE OPTIONS (MAIN);  
  DECLARE (VALOR,SOMA)  
    DECIMAL FIXED (10);  
  SOMA = 0;  
  GET DATA (VALOR);  
  DO WHILE (VALOR  $\neq$  -1);  
    SOMA = SOMA + VALOR;  
    GET DATA (VALOR);  
  END;  
  PUT DATA (SOMA);  
END PSOMA;
```

Outras Linguagens de Alto Nível

BASIC

```
10 DIM VALOR,SOMA
20 LET SOMA=0
30 PRINT "ENTRE COM UM VALOR"
40 INPUT VALOR
50 IF VALOR>0 THEN 80
60   SOMA = SOMA + VALOR
70 GO TO 30
80 PRINT SOMA
90 END
```


Outras Linguagens de Alto Nível

ALGOL

```
BEGIN
  REAL VALOR,SOMA;
  SOMA:=0;
  READ(VALOR);
  WHILE (VALOR>0)
  DO BEGIN
    SOMA:=SOMA+VALOR;
    READ(VALOR);
  END;
  WRITE(SOMA);
END.
```

Outras Linguagens de Alto Nível

PASCAL

```
PROGRAM SOMA;  
  VAR VALOR,SOMA:REAL;  
  BEGIN  
    SOMA:=0;  
    READ(VALOR);  
    WHILE (VALOR>0)  
    DO BEGIN  
      SOMA:=SOMA+VALOR;  
      READ(VALOR);  
    END;  
    WRITE(SOMA);  
  END.
```

Outras Linguagens de Alto Nível

Cobol – COmmon Business Oriented Language

IDENTIFICATION DIVISION.

ENVIRONMENT DIVISION.

DATA DIVISION.

FD ARQ1

RECORDINE MODE IS F

BLOCK CONTAINS 50 TO 125 CHARACTERES

DATA RECORD IS REG-ARQ.

01 REG-ARQ.

02 CHAVE PICTURE 99.

02 COD PICTURE XXX.

PROCEDURE DIVISION.

UNICA SECTION.

INICIO.

READ ARQ1 AT END GO TO FIM.

MULTIPLY TOTAL BY TAXA GIVING LIQUIDO ROUNDED.

MOVE SPACES TO LINHA.

WRITE SAIDA AFTER ADVANCING 2.

INICIO-DO-PROCESSAMENTO.

OPEN INPUT ARQUIVO-DE-COMPRA OUTPUT RELATORIO.

LER-E-VERIFICAR.

READ ARQUIVO-DE-COMPRA AT END GO TO FIM.

IF NUMERO-DE-TRANSACOES IN ESTE-MÊS IS EQUAL TO ZERO, OR SALDO-ATUAL IS NEGATIVE

GO TO LER-E-VERIFICAR.

Outras Linguagens de Alto Nível

LISP

```
(  
  DEF FATORIAL(N)  
    ( COND((EQ N 0) 1)  
          (T (TIMES N  
              (FATORIAL (SUB N 1)  
                )  
              )  
            )  
          )  
    )  
  )  
)
```




Algoritmos

Fazer um algoritmo para o jogo da velha!

Algoritmos

Escrever um algoritmo para jogar o jogo da velha

início

enquanto (existir um quadrado livre) e (ninguém perdeu (ganhou) o jogo)

“?Espere a jogada do adversário?”

se (existir um quadrado livre)

então se (existirem 2 quadrados marcados com o mesmo símbolo em linha ou nas diagonais e o terceiro desocupado)

então jogue no quadrado desocupado

senão se o centro estiver livre

então jogue no centro

senão se (existe algum canto livre)

então se (for sua segunda jogada e o adversário possuir dois cantos marcados em diagonal)

então jogue num quadrado exceto os cantos

senão jogue no canto

fim se

senão jogue no local disponível

fim se

fim se

fim se

fim se

fim enquanto

fim



Algoritmos

A noção de um algoritmo, de uma ordem executável para o estabelecimento de um efeito final, é muito comum na vida cotidiana receitas, manuais, partituras etc.

Ao escrever um algoritmo, começa-se considerando o acontecimento como um processo, dividindo-o em uma sequência de (sub)ações que deverão ser realizadas sucessivamente.

É necessário o uso de um conjunto de mecanismos que permita o desenvolvimento de algoritmos, e que seja suficientemente conciso para evitar ambiguidades, ao mesmo tempo em que procure libertar o programador do rigor e das limitações de uma linguagem de programação específica.

Algoritmos

Exemplo:

Fazer um algoritmo para ler dois valores do teclado e mostrar a sua soma na tela.

Primeiro passo:

Identificar o objetivo do algoritmo.

início

“ ler dois valores inteiros e mostrar sua soma ”

fim.

Segundo passo:

Isolar processos.

início

// ler dois valores inteiros e mostrar sua soma

“ ler dois valores inteiros ”

“ calcular a soma dos dois valores”

“ mostrar a soma ”

fim.

Algoritmos

Terceiro passo:
Isolar ações consideradas primitivas.

início

// ler dois valores inteiros e mostrar sua soma

“ definir um local para armazenar o primeiro valor ”

“ definir outro local para armazenar o segundo valor ”

“ definir um local para armazenar a soma dos dois valores ”

“ ler um valor do teclado e armazená-lo ”

“ ler outro valor do teclado e armazená-lo ”

“ Calcular a soma dos dois valores ”

“ mostrar a soma ”

fim.

Algoritmos

Declaração de variáveis – Tipos básicos

Tipos possíveis:

inteiro: qualquer número inteiro, negativo, nulo, ou positivo: ex. -5, 0, 235

real: qualquer número real, negativo, nulo, ou positivo. Ex. -5, 30.5, 0, 40.

literal: qualquer conjunto de caracteres alfanuméricos.
Ex “AB”, “XYZ”, “Abacate”.

São exemplos de declaração de variáveis:

x1, soma, teste :**inteiro**

a, produto :**real**

nome, frase :**literal**

Regra para definição de variáveis:

- 1- Caracteres permitidos: letras, números e o _
- 2 – Primeiro caractere sempre letra ou o _
- 3 – não são permitidos espaços em branco e caracteres especiais (&,%+,#,...)
- 4 – não utilizar palavras reservadas (que pertençam à linguagem de programação)

Algoritmos

Comandos Básicos

Comando de Atribuição. ,

Para a atribuição de um valor a uma variável, símbolo \leftarrow

Sintaxe: variável \leftarrow expressão

Exemplos:

$x1 \leftarrow 5$

$a \leftarrow 1.5$

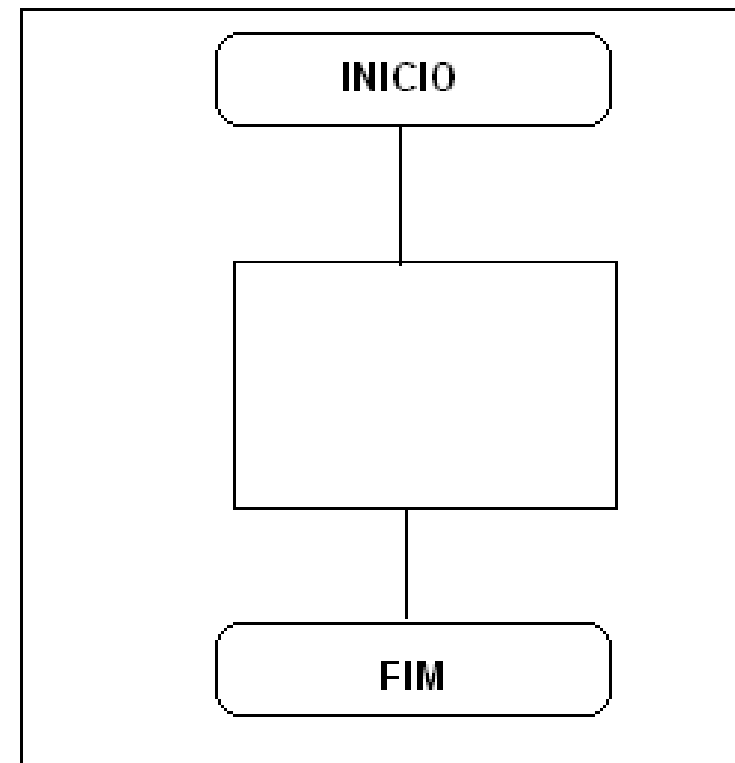
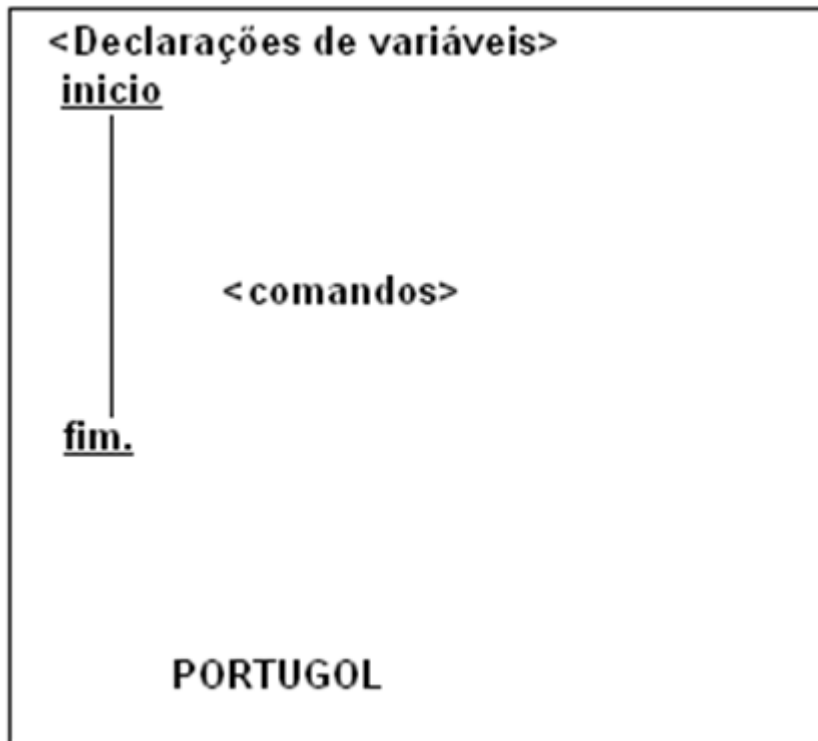
$soma \leftarrow 0$

$soma \leftarrow soma + a$

$resp \leftarrow \text{sen}(a) + x^2$

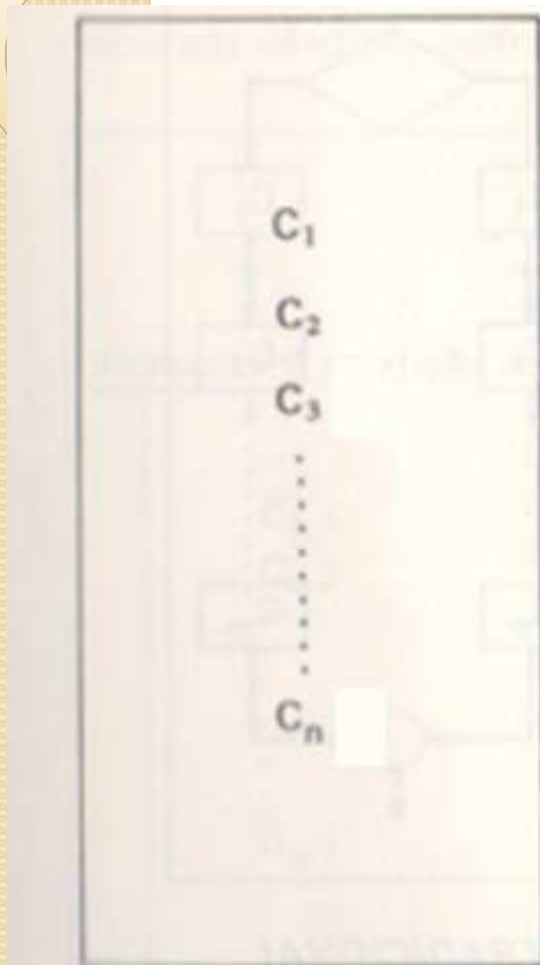
Algoritmos

Blocos

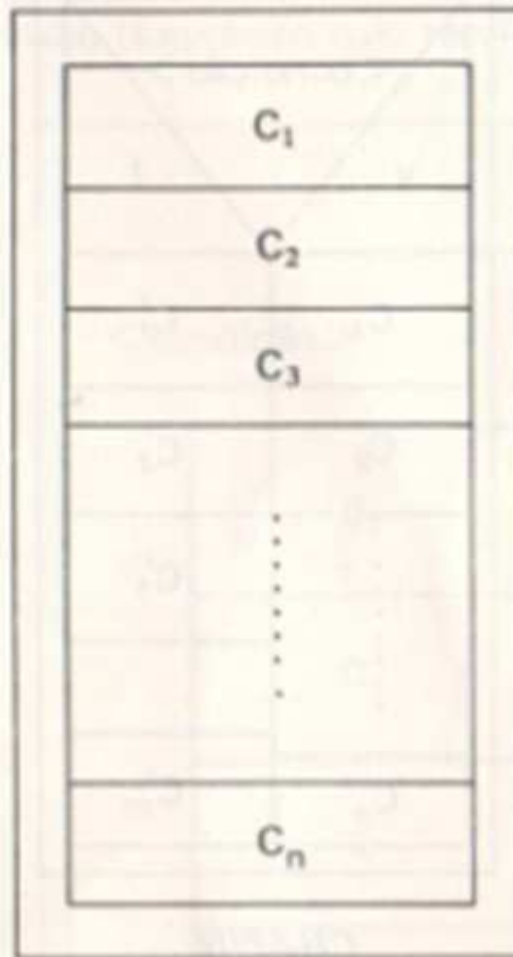


Algoritmos

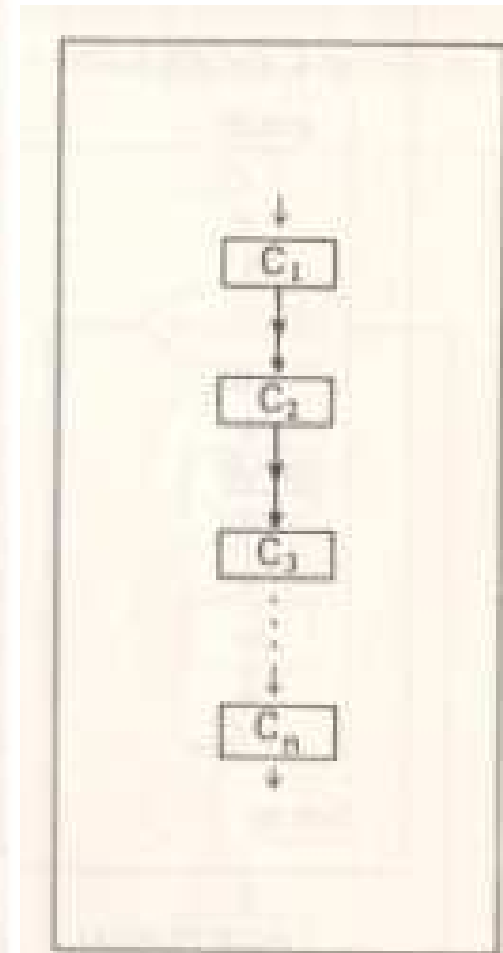
Sequência simples:



PORTUGOL



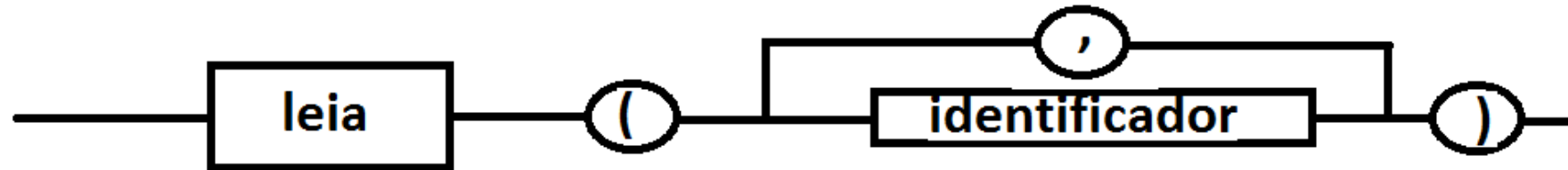
CHAPIN



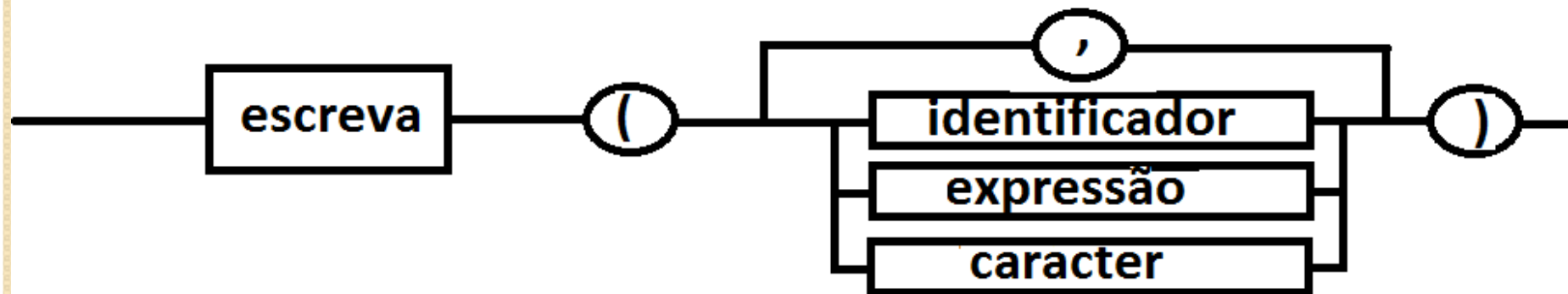
TRADICIONAL

Algoritmos

Entrada / Saída:



Exemplos:
leia (A, B)



Exemplos:
escreva ("Média = ", Media)
escreva ("Valor lido: ", N, " Resultado = ", N^2 + 5)

Algoritmos

Exemplo

algoritmo "Média"

// Função : calcular a média entre dois números

// Autor : Roberto Rocha

// Data : dd/mm/aaaa

// Seção de Declarações

var

a,b :inteiro

m:real

inicio

// Seção de Comandos

leia(a,b)

$m \leftarrow (a+b)/2$

escreva("A média de ",a," e ",b, " é igual a ",m)

fimalgoritmo

Algoritmos

```
var
  a,b :inteiro
  m:real
inicio
  // Seção de Comandos
  leia( a,b)
   $m \leftarrow (a+b)/2$ 
  escreva("A média de ",a," e ",b, " é igual a ",m)
finalgoritmo
```

Escreva um programa que solicite ao usuário dois números e imprima a soma deles.

Exercício:

1. Escreva um programa que solicite ao usuário dois números e imprima a soma deles.
2. Escreva um programa que solicite ao usuário dois números e imprima:
 - a) A soma deles
 - b) A Multiplicação destes números
 - c) A média aritmética destes números.
3. Escreva um programa que solicite ao usuário a altura e o raio de um cilindro circular e imprima o volume do cilindro. O volume de um cilindro circular é calculado por meio da seguinte fórmula:
$$\text{Vol} = \text{PI} * \text{raio}^2 * \text{altura}$$
4. Uma empresa contrata um encanador a R\$ 20.00 por dia. Crie um programa que solicite o número de dias trabalhados pelo encanador e imprima a quantia líquida que deverá ser paga, sabendo-se que são descontados 8% de impostos.
5. O cardápio de uma lanchonete é dado abaixo. Prepare um programa que leia a quantidade de cada item que você consumiu e calcule a conta final.

Hambúrguer.....	R\$ 30,00
Cheeseburger.....	R\$ 25,50
Fritas.....	R\$ 20,50
Refrigerante.....	R\$ 10,00
Milkshake.....	R\$ 30,00