



# ALGORITMOS E FLUXOGRAMAS

PROF. ANDRÉ BACKES



# INTRODUÇÃO

- Computadores = cérebros eletrônicos?
  - Computadores são máquinas e, por si só, não podem ser inteligentes.
  - Alguém as projetou e deu a ela todas as características que possuem.

# INTRODUÇÃO

- Computadores têm facilidade para lidar com um determinado assunto, uma familiaridade com alguma área do conhecimento.
- Exemplo
  - Um computador pode realizar um calculo 10 bilhões de vezes mais rápido que nosso cérebro.

# INTRODUÇÃO

- Por outro lado, nosso cérebro opera em paralelo, isto é, pode resolver vários problemas ao mesmo tempo.

# ALGORITMOS

- Para resolver um problema no computador é necessário que ele seja primeiramente descrito de uma forma clara e precisa.
- O conceito de algoritmo é frequentemente ilustrado pelo exemplo de uma receita.

# ALGORITMO: BOLO DE CHOCOLATE

- Aqueça o forno a 180 C
- Unte uma forma redonda
- Bater no liquidificador
  - 1 e 1/2 xícara de leite
  - 1 xícara de óleo
  - 3 ovos
  - 2 xícaras de açúcar
  - 1 xícara de chocolate
- Transferir para uma tigela e misturar com
  - 1 e 1/2 xícara de farinha
  - 1 e 1/2 xícara de aveia em flocos
  - 1 colher (sopa) de fermento
- Deite a massa na forma
- Leve ao forno durante 40 minutos

# ALGORITMOS

- Um algoritmo pode ser definido como uma sequência simples e objetiva de instruções para solucionar um determinado problema
  - A instrução é uma informação que indica a um computador uma ação elementar a executar
- A sequência de instruções deve ser
  - Finita
  - Não pode ser ambígua

# ALGORITMOS

- Por que NÃO ambíguo?
  - Cada instrução do algoritmo deve ser precisamente definida, sem permitir mais de uma interpretação de seu significado.
  - Os algoritmos devem se basear no uso de um conjunto de instruções bem definido, que constituem um vocabulário de símbolos limitado.



# ALGORITMOS

- Os algoritmos são capazes de realizar tarefas como:
  - Ler e escrever dados;
  - Avaliar expressões algébricas, relacionais e lógicas;
  - Tomar decisões com base nos resultados das expressões avaliadas;
  - Repetir um conjunto de ações de acordo com uma condição

# ALGORITMOS

- Como seria um algoritmo para as seguintes tarefas
  - Trocar um lâmpada
  - Apontar um lápis
  - Média de 2 números
  - Somar N números

# ALGORITMOS

- O algoritmo é a lógica do nosso problema. É a sequência de passos que eu faço na minha cabeça (ou no papel, quando for mais complexo) antes de escrever em uma linguagem de programação.
- Podem existir vários algoritmos diferentes para resolver o mesmo problema.
  - Exemplo: média de dois números

$$z = \frac{x + y}{2}$$

$$z = \frac{x}{2} + \frac{y}{2}$$

# ALGORITMOS

- Um algoritmo é um procedimento computacional definido composto de 3 partes
  - Entrada de dados
    - São os dados do algoritmo informados pelo usuário
  - Processamento de dados
    - São os procedimentos utilizados para chegar ao resultado
    - É responsável pela obtenção dos dados de saída com base nos dados de entrada
  - Saída de dados
    - São os dados já processados, apresentados ao usuário

# ALGORITMOS

- O algoritmo que usamos depende principalmente do tempo que ele demora pra ser executado e a memória que ele gasta no computador.
- Chamamos a isso de custo.
  - Exemplo: ordenar números
    - Quicksort, Mergesort, Bubblesort, etc

# ALGORITMOS

- Para escrever um algoritmo precisamos descrever a sequência de instruções, de maneira simples e objetiva. Algumas dicas:
  - Usar somente um verbo (imperativo) por frase
  - Imaginar que você está desenvolvendo um algoritmo para pessoas que não trabalham com computadores
  - Usar frases curtas e simples
  - Ser objetivo
  - Evitar palavras que tenham sentido dúbio

# PSEUDO-CÓDIGO

- Até aqui, os algoritmos foram descritos em linguagem natural
- Outra forma seria o uso de uma pseudo-linguagem ou pseudo-código
  - Emprega uma linguagem intermediária entre a linguagem natural e uma linguagem de programação usada para descrever os algoritmos
  - O pseudocódigo não requer toda a rigidez sintática necessária numa linguagem de programação, permitindo que o aprendiz se detenha na lógica do algoritmos e não no formalismo da sua representação

# PSEUDO-CÓDIGO

- Ex: ler dois número e imprimir o maior deles

```
Leia A;  
Leia B;  
Se A > B então  
    Imprima A;  
Senão  
    Imprima B;  
Fim Se
```



# PSEUDO-CÓDIGO

- Como seria um pseudo-código para as seguintes tarefas
  - Trocar um lâmpada
  - Apontar um lápis
  - Média de 2 números
  - Somar N números

# TIPOS DE PROCESSAMENTO

- Ao elaborar um algoritmo, devemos ter em mente qual o tipo de processamento será executado.
- Basicamente, existem 3 tipos de processamento
  - Processamento sequencial
  - Processamento condicional
  - Processamento com repetição
    - Repetição determinada
    - Repetição indeterminada

# TIPOS DE PROCESSAMENTO

- Processamento sequencial
  - As instruções são executadas uma após a outra
  - Não existe desvio na sequência das instruções
  - Cada instrução é executada uma única vez
- Exemplo
  - Imprimir a média aritmética de duas notas

```
Leia nota1  
Leia nota2  
media = (nota1 + nota2) / 2  
Imprima media
```

# TIPOS DE PROCESSAMENTO

- Processamento sequencial
  - A ordem das instruções é importante!

```
Leia nota1
Leia nota2
Imprima media
media = (nota1 + nota2) / 2
```



```
media = (nota1 + nota2) / 2
Leia nota1
Leia nota2
Imprima media
```



```
Leia nota1
Leia nota2
media = (nota1 + nota2) / 2
Imprima Media
```

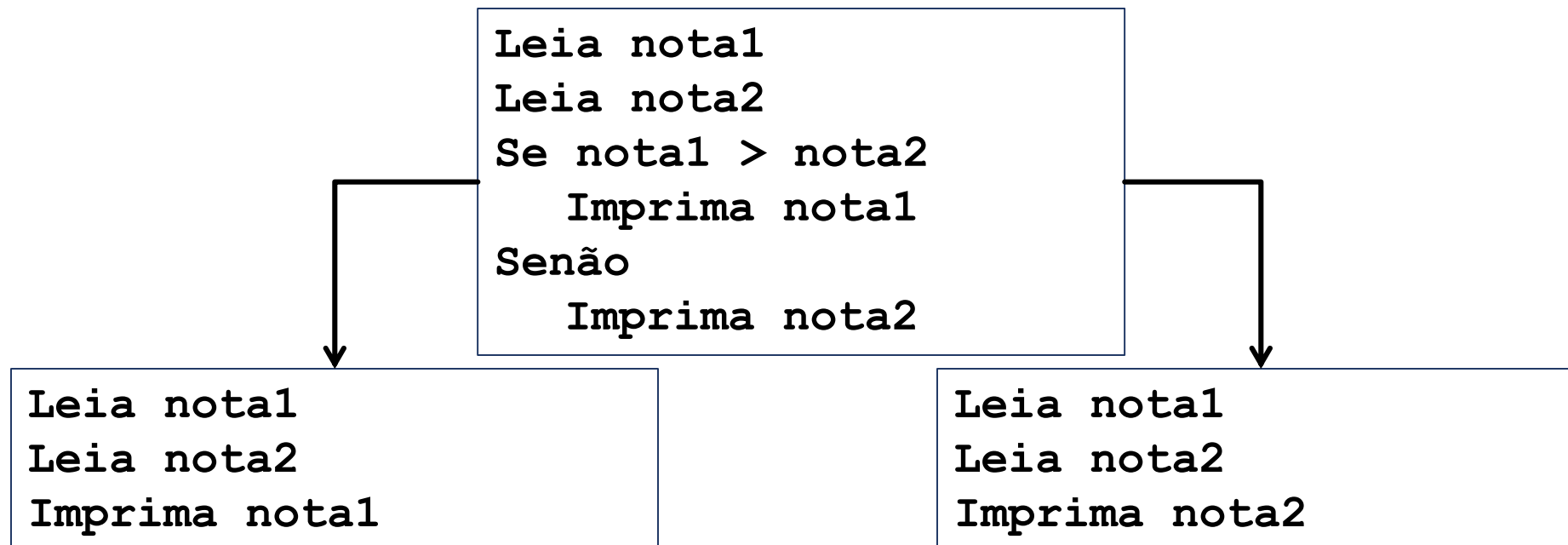


# TIPOS DE PROCESSAMENTO

- Processamento condicional
  - Um conjunto de instruções (pode ser apenas uma) pode ou não ser executado
  - Depende de uma condição
  - Se a condição testada for verdadeira, o conjunto de instruções é executado

# TIPOS DE PROCESSAMENTO

- Processamento condicional
  - As instruções executadas dependem da situação
- Exemplo
  - Imprimir a maior dentre duas notas lidas



# TIPOS DE PROCESSAMENTO

- Processamento com repetição
  - Um conjunto de instruções (pode ser apenas uma) é executado um número definido ou indefinido de vezes
  - Pode ser determinada por uma condição de parada
    - O conjunto de instruções é executado enquanto a condição for verdadeira
    - O teste da condição é realizado antes de qualquer operação

# TIPOS DE PROCESSAMENTO

- Processamento com repetição
  - Também chamado de laços condicionais
  - Repetem um conjunto de comandos em seu interior
- Exemplo
  - Imprimir a soma dos números inteiro de 1 a N
    - $Soma = 1 + 2 + 3 + \dots + N$
    - Necessidade de se identificar o que deve ser repetido no algoritmo

$$Soma = 1 \oplus 2 \oplus 3 \oplus \dots \oplus N$$



# TIPOS DE PROCESSAMENTO

- Processamento com repetição – Exemplo 1
  - Imprimir a soma dos números inteiro de 1 a N
    - $Soma = 1 + 2 + 3 + \dots + N$
    - Identificar: valor inicial ( $nro = 1$ ), valor final ( $N$ ), onde o resultado será armazenado ( $soma$ ), quando parar ( $nro \leq N$ ), variável (contador) que controla o número de repetições ( $nro$ ), etc.

```
Leia N
soma = 0
nro = 1
Enquanto nro <= N
    soma = soma + nro
    nro = nro + 1
Imprima soma
```

# TIPOS DE PROCESSAMENTO

- Processamento com repetição – Exemplo 2
  - Imprimir a média dos números positivos digitados. Parar quando um valor negativo ou zero for digitado
  - Problema
    - Não sabemos quantos números serão digitados!
    - Não tem como definir valor inicial ou final
    - A repetição é determinada por uma condição de parada (valor negativo ou zero)

# TIPOS DE PROCESSAMENTO

- Processamento com repetição – Exemplo 2
  - Imprimir a média dos números positivos digitados. Parar quando um valor negativo ou zero por digitado
    - Identificar: onde o resultado será armazenado (soma), quando parar (valor  $\leq 0$ ), variável (contador) que controla o número de repetições (valor), etc.

```
soma = 0
N = 0
Leia valor
Enquanto valor > 0
    soma = soma + valor
    N = N + 1
    Leia valor
Imprima soma/N
```

# TESTE DE MESA

- Após desenvolver um algoritmo é preciso testá-lo. Uma maneira de se fazer isso é usando o teste de mesa
  - Basicamente, esse teste consiste em seguir as instruções do algoritmo de maneira precisa para verificar se o procedimento utilizado está correto ou não
    - Tentar utilizar um caso onde se conhece o resultado esperado
  - Permite reconstituir o passo a passo do algoritmo

# TESTE DE MESA

- Criar uma tabela de modo que
  - Cada coluna representa uma variável
  - As linhas correspondem as alterações naquela variável (de cima para baixo)

valor	N	soma

# TESTE DE MESA

- Exemplo 1: imprimir a média dos números positivos digitados. Parar quando um valor negativo ou zero por digitado
  - Valores digitados: 4, 2, 3 e -1
  - Média é 3

```
soma = 0
N = 0
Leia valor
Enquanto valor > 0
    soma = soma + valor
    N = N + 1
    Leia valor
Imprima soma/N
```

valor	N	soma
4	0	0
2	1	4
3	2	6
-1	3	9

# FLUXOGRAMA

- Existem estudos que comprovam que o ser humano consegue gravar melhor uma mensagem, quando esta é acompanhada de imagens
- “Uma imagem vale mais do que mil palavras”

# FLUXOGRAMA

- Um fluxograma é um diagrama, escrito em uma notação gráfica simples, usado para representação visual de algoritmos.
  - Algoritmo -> texto
  - Fluxograma -> gráfico



# FLUXOGRAMA

- Representa uma seqüência de operações qualquer, de forma detalhada, onde todos os passos são visualizados.
- É utilizado também em outras áreas
  - Processos dentro de uma empresa, linha de produção, etc.

# FLUXOGRAMA

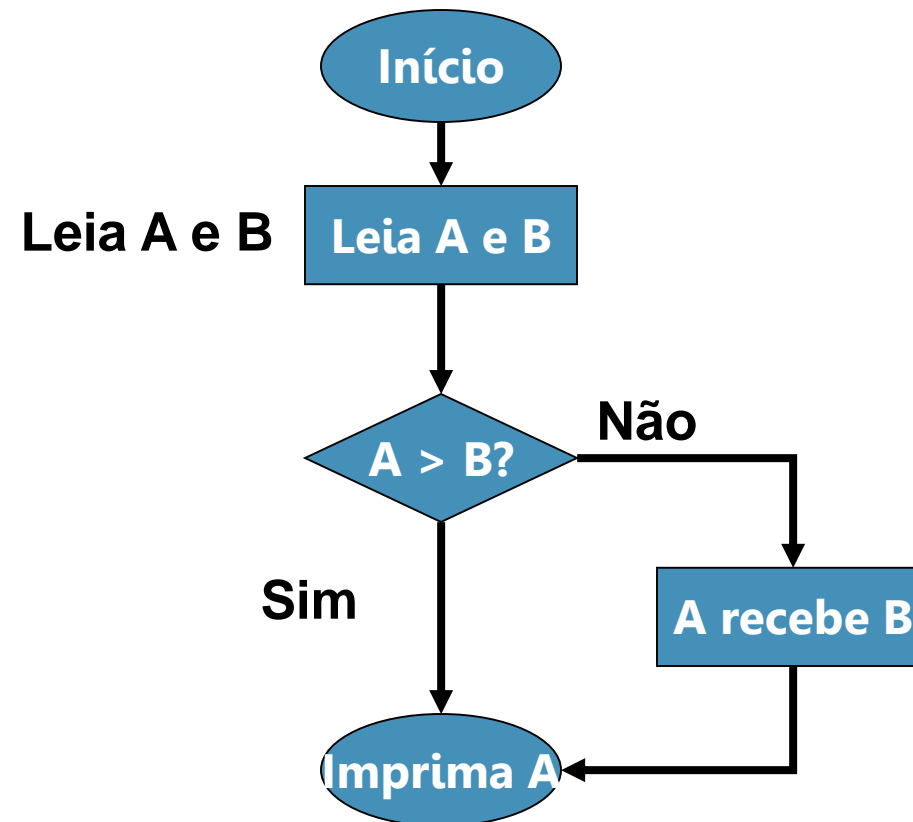
- É útil para compreensão de controle de fluxo nas fases iniciais de aprendizado de programação, ou quando a linguagem na qual os programas são escritos é muito primitiva.

# FLUXOGRAMA

- Vantagens
  - Padronização na representação;
  - Permite descrever com maior rapidez um conjunto de tarefas;
  - Facilita a leitura e o entendimento de uma atividade;

# EXEMPLO

- Imprimir maior valor lido



# FLUXOGRAMA - SÍMBOLOS

- Início e Fim
  - Podem ser círculos ou formas ovais
  - Normalmente contém as palavras "Início" ou "Fim", ou alguma expressão sinalizando o início ou fim do processo.



**Início**

**Fim**

**Imprima X**

# FLUXOGRAMA - SÍMBOLOS

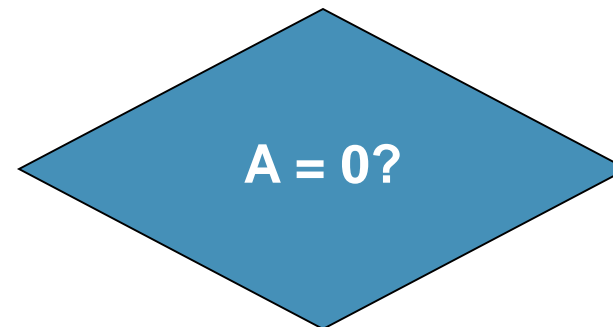
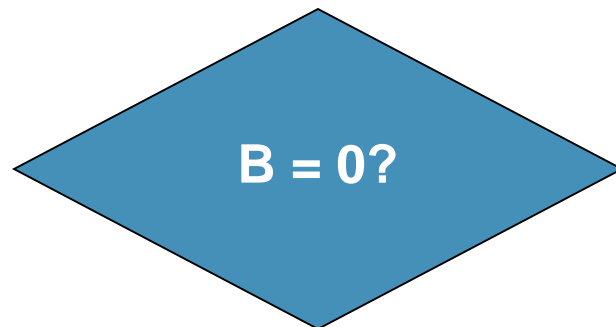
- Processo ou operação
  - Representados por retângulos.
  - Indicam uma tarefa a ser executada pelo programa.

**Somar + 1 a X**

**Multiplicar X por Y**

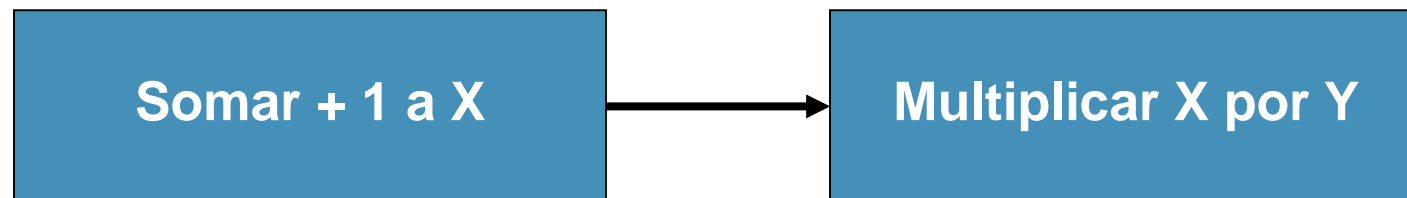
# FLUXOGRAMA - SÍMBOLOS

- Condição ou Decisão
  - Representado por losangos
  - Normalmente contém uma pergunta do tipo Sim/Não ou um teste de Verdadeiro/Falso.
  - Mudança no fluxo



# FLUXOGRAMA - SÍMBOLOS

- Setas
  - Conectam 2 símbolos quaisquer.
  - Definem o fluxo de controle.
  - Ordem das operações a serem realizadas.



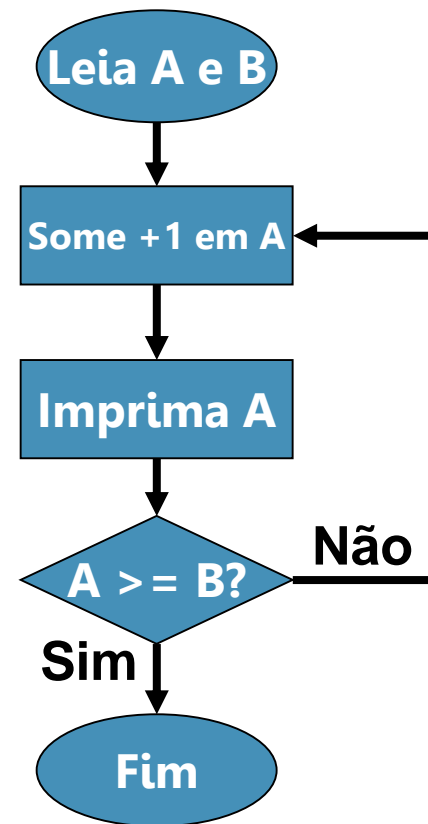


# FLUXOGRAMA

- Estrutura de decisão não necessariamente leva a uma caminho alternativo.
- Um processo pode ser repetido.

# EXEMPLO

- Listar números entre dois valores



# FLUXOGRAMA

- Como seria um fluxograma para as seguintes tarefas
  - Trocar um lâmpada
  - Apontar um lápis
  - Somar N números
  - Dividir 2 números

# METODOLOGIAS DE PROGRAMAÇÃO

- A resolução de um problema começa com a definição dos dados e tarefas básicas.
- Esta definição inicial é feita em nível bem alto e geral.
- Não há preocupação com os detalhes (refinamento).

# METODOLOGIAS DE PROGRAMAÇÃO

- Refinamentos Sucessivos (Top-Down )
  - Consiste em pegar um grande problema, de difícil solução, e dividi-lo em problemas menores que devem ser mais facilmente resolvidos
    - Decompor uma ou várias tarefas em sub-tarefas mais detalhadas
    - É um processo iterativo, isto é, sub-tarefas podem ser decompostas em sub-tarefas ainda mais detalhadas

# REFINAMENTOS SUCESSIVOS

- Exemplo: trocar um pneu furado
  - Levantar o carro parcialmente;
  - Retirar o pneu furado;
  - Instalar o novo pneu;
  - Abaixar o carro

# REFINAMENTOS SUCESSIVOS

- Exemplo: trocar um pneu furado
  - **Retirar o estepe;**
  - Levantar o carro parcialmente;
  - Retirar o pneu furado;
  - Instalar o novo pneu;
  - Abaixar o carro
  - **Apertar bem as porcas;**

# REFINAMENTOS SUCESSIVOS

- Exemplo: trocar um pneu furado
  - **Pegar as ferramentas no porta-malas;**
  - Retirar o estepe;
  - **Instalar o macaco;**
  - Levantar o carro parcialmente;
  - **Afrouxar os parafusos do pneu furado;**
  - Retirar o pneu furado;
  - Instalar o novo pneu;
  - Abaixar o carro
  - Apertar bem as porcas;
  - **Guardar o pneu furado e as ferramentas;**



# REFINAMENTOS SUCESSIVOS

- O algoritmo proposto pode ainda ser refinado de várias outras formas
  - O que fazer se o macaco não estiver no porta-malas?
  - O que fazer se o estepe também estiver vazio?
  - Deve-se sempre puxar o freio de mão antes de executar estas operações.
  - Limpar as mãos;
  - Consertar o pneu furado;
  - Etc