
Introdução ao Processamento de Dados

Algoritmos

**Instituto de Matemática e Estatística
(IME)
Universidade do Estado do Rio de Janeiro
(UERJ)**

Semestre 2015.2

Definição de Algoritmo

“Uma sequência de passos finitos, bem definidos, para a solução de um problema.”

Características de um Algoritmo

- Para que um computador possa desempenhar uma tarefa é necessário:
 - que ela seja detalhada passo a passo
 - numa forma compreensível pela máquina, utilizando aquilo que se chama de programa.
- Neste sentido, um programa de computador nada mais é que um algoritmo escrito numa LINGUAGEM compreensível pelo computador.

Características de um Algoritmo

Um algoritmo pode ser descrito através de:

- Descrição Narrativa
- Diagrama de Chapin
- Fluxograma
- Pseudocódigo

Descrição Narrativa

- Consiste em descrever, de forma narrativa, cada um dos passos para a resolução de problemas
- Exemplo: Troca de um pneu
 - 1) Afrouxar os parafusos
 - 2) Suspende o carro
 - 3) Retirar os parafusos e o pneu
 - 4) Colocar o pneu reserva
 - 5) Colocar e apertar ligeiramente os parafusos
 - 6) Abaixar o carro
 - 7) Terminar de apertar os parafusos

Descrição Narrativa

- Podemos usar a descrição narrativa para resolver qualquer problema
- Exercícios:
 - Fazer um algoritmo usando descrição narrativa para resolver o problema de atravessar um rio com um LOBO, uma OVELHA e um ALFACE.
 - Se o LOBO ficar sozinho com a OVELHA ele a devora, se a OVELHA ficar sozinha com o ALFACE, ela o devora!

Descrição Narrativa

- Problema do LOBO, OVELHA e ALFACE:
 - 1) Atravessar a ovelha para a margem 2
 - 2) Voltar vazio para a margem 1
 - 3) Atravessar o Lobo para margem 2
 - 4) Voltar com a ovelha para a margem 1
 - 5) Atravessar o alface para a margem 2
 - 6) Voltar vazio para a margem 1
 - 7) Atravessar a ovelha para a margem 2

Descrição Narrativa

- Podemos usar a descrição narrativa para resolver qualquer problema
- Exercícios:
 - Fazer um algoritmo usando descrição narrativa para atravessar 3 padres e 3 canibais.
 - Se em algum momento houver mais canibais do que padres, os canibais são jantados.
 - Jogo “Missionários e Canibais”. (<http://rachacuca.com.br/jogos/missionarios-e-canibais/>)

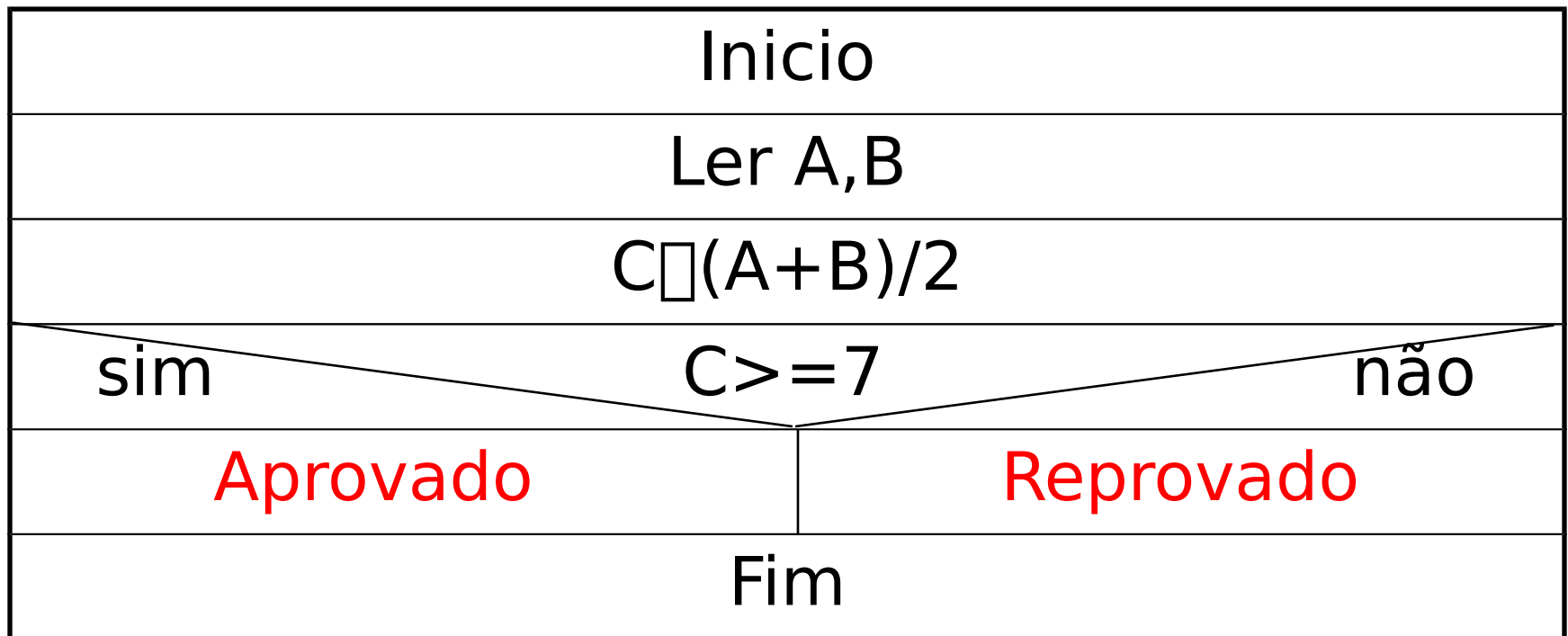
Descrição Narrativa

■ Problema dos PADRES e CANIBAIS

- 1) Atravessar 1 padre e 1 canibal a margem 2
- 2) Voltar com 1 padre para a margem 1
- 3) Atravessar 2 canibais para margem 2
- 4) Voltar com 1 canibal para a margem 1
- 5) Atravessar 2 padres para a margem 2
- 6) Voltar com 1 padre e 1 canibal para margem 1
- 7) Atravessar 2 padres para a margem 2
- 8) Voltar com 1 canibal para a margem 1
- 9) Atravessar 2 canibais para a margem 2
- 10) Voltar 1 canibal para a margem 1
- 11) Atravessar 2 canibais para a margem 2





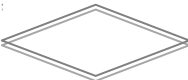

Diagrama de Chapin

- Ler duas notas e dizer se aluno foi aprovado ou reprovado.



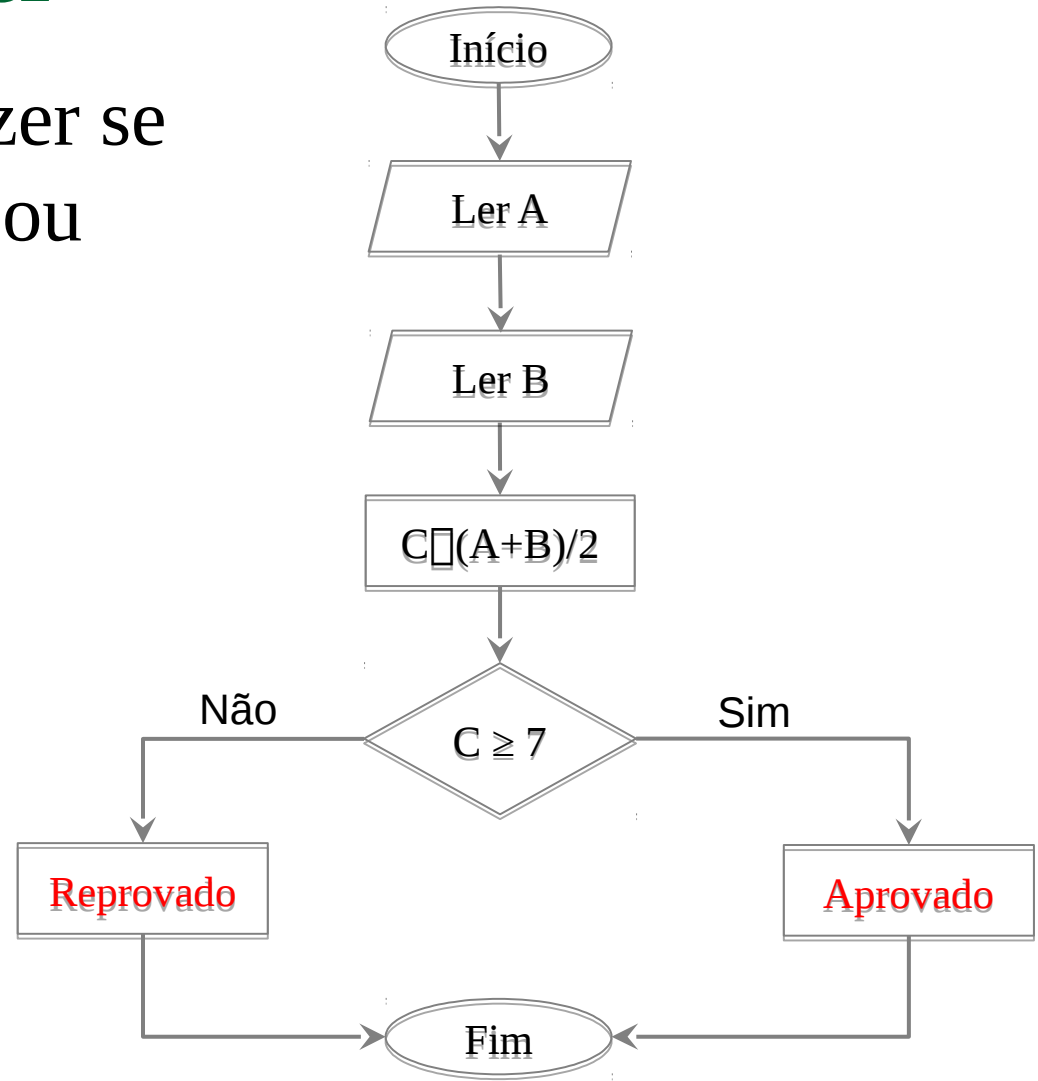
Fluxograma

- Representação esquemática de um algoritmo.
- Grafo dirigido composto por formas que representam diferentes ações (formas básicas):

Forma	Nome	Descrição
	Seta	Indica que o controle passa para a forma apontada
	Terminal	Representa o começo ou término do algoritmo
	Entrada/Saída	Representa entrada ou saída de dados
	Processo	Representa uma ação/cálculo/processo
	Decisão	Representa uma decisão
	Proc. Predefinido	Representa um outro fluxograma (aninhado)

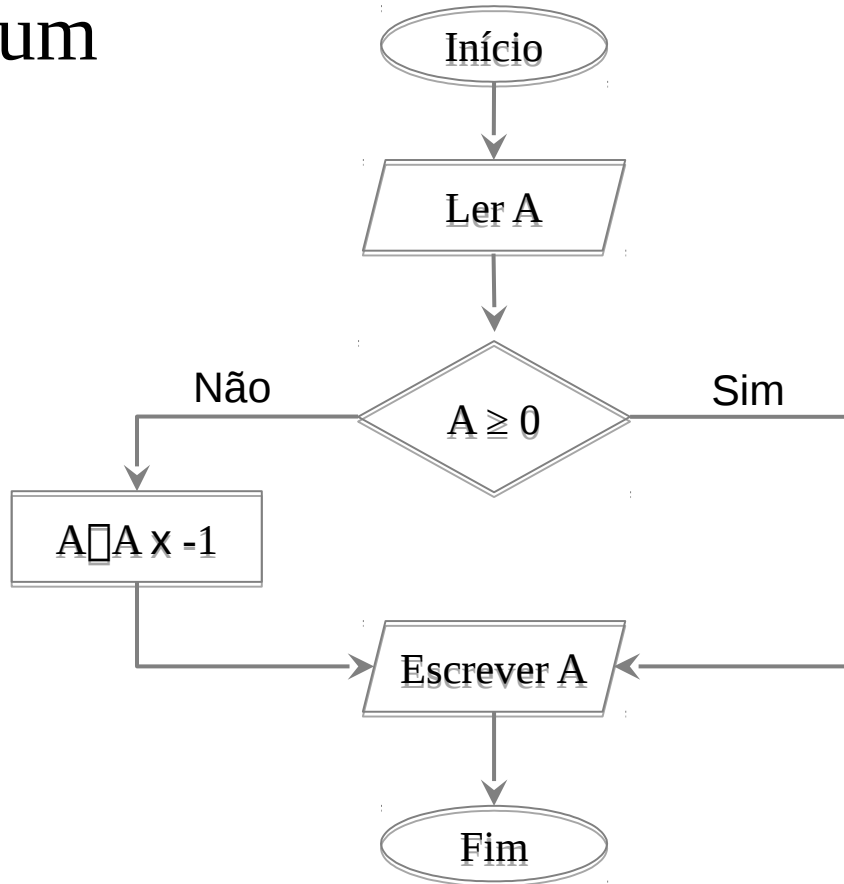
Fluxograma

- Ler duas notas e dizer se aluno foi aprovado ou reprovado.



Fluxograma

- Escrever o módulo (valor absoluto) de um número.



Pseudocódigo

- Forma genérica de escrever um algoritmo, utilizando uma linguagem simples, próxima da natural.
- Não há necessidade de conhecer a sintaxe de nenhuma linguagem de programação.
- Não pode ser executado num sistema real (computador), de outra forma deixaria de ser pseudo.

Pseudocódigo

- Há algumas propostas mais rígidas, em termos de sintaxe, de português estruturado: Portugol, G-Portugol, Portugol Viana.
- Formalismos não se justificam neste curso: nos concentraremos posteriormente na sintaxe da linguagem Pascal.

Qualidades do Algoritmo

- Clareza: refere-se à facilidade de leitura, entendimento do que cada etapa faz.
- Integridade: precisão da descrição/especificação das informações/dados manipulados.
- Elegância/Simplicidade: composto de um número de etapas pequeno, mas completo.

Qualidades do Algoritmo

- Modularidade: refere-se ao particionamento do problema em pequenos grupos de ações/etapas, que contribuem para um objetivo particular (sub-rotinas, funções).
- Generalidade: é interessante que seja tão genérico quanto possível, de forma a permitir a reutilização ou adaptação de seus componentes em outros projetos.

Tipos de Dados

Tipos básicos de dados:

- Numéricos
 - Inteiros
 - Reais
- Alfanuméricos
 - Caractere: um único caractere
 - Cadeia (String): uma cadeia de caracteres
- Lógicos ou Booleanos
 - Verdadeiro ou Falso

Variáveis

- Conceito familiar ao da matemática.
- Uma variável é uma entidade que possui um valor e é identificada por um nome particular.
- Pode receber muitos valores distintos no decorrer de um programa, mas num dado instante tem apenas um valor.
- Geralmente só pode ser de um tipo de dado (caso da linguagem Pascal).

Variáveis

- Por exemplo, comprimento dos lados de um triângulo: A, B e C
- Existem regras para se dar nome a uma variável:
 - Uma única letra, ou
 - Inicia com uma letra que pode ser seguida de dígitos ou letras, em qualquer quantidade
 - Não deve possuir caracteres que tenham funções específicas: + - * / = % “ ‘ ! ~ ? ; , () .
 - Não deve possuir espaços

Variáveis

- Existem regras para se dar nome a uma variável:
 - Sublinhado _ funciona como uma letra (ok)
 - Geralmente só se aceitam letras presentes na língua inglesa
 - Não pode ser um nome “reservado” numa linguagem, e.g., real, cadeia, inteiro, inicio, fim, se, não, verdadeiro, falso, ...

Variáveis

- Quais os nomes de variáveis válidos?

A

5B

A32B

x-y

A:B

KM/H

Caixa_Preta

b*d

E(2)

_NUM

Caixa Preta

A1

Endereço

A111111

média

ação

Programação Estruturada

Possui três tipos de comandos básicos:

- Sequência simples
 - Atribuição
 - Entrada e Saída
- Decisão
- Repetição

Atribuição

- Especifica o valor que será dado a uma variável
- Sintaxe: \leftarrow
- Exemplos:
 - $A \leftarrow 3$
 - Significa: variável A recebe o valor 3
 - $X \leftarrow 4 + 5$
 - Significa: X recebe a soma de $4 + 5$
 - $b \leftarrow b + 1$
 - Significa: b recebe o conteúdo do próprio b, mais 1

Operadores Aritméticos

+ soma

- subtração

* multiplicação

/ divisão real

// divisão inteira

% resto inteiro da divisão

() aninhamento/precedência

Operadores Aritméticos

Exemplos:

- $A \leftarrow 4$
- $C \leftarrow A + 5$
- $\text{mult} \leftarrow A * 2$
- $N1 \leftarrow 9 / 2$
- $N2 \leftarrow 9 // 2$
- $N3 \leftarrow 9 \% 2$
- $N4 \leftarrow (9 + 4) // 2$

Operadores Relacionais

- Resultado da operação é um valor lógico: verdadeiro ou falso

= = igual

!= diferente

> maior

>= maior ou igual

< menor

<= menor ou igual

Operadores Relacionais

Exemplos:

- $5 > 4$
- $A \leftarrow 5 > 4$
- $(5 + 6) > (7 + 8)$
- $(5 * 2) >= ((4 * 5) / 2)$

Operadores Lógicos

- Resultado da operação é um valor lógico: verdadeiro ou falso

e

ou

não

- Só fazem sentido se aplicados a dados (variáveis) com valor lógico (verdadeiro ou falso)

Operadores Lógicos

■ Tabela verdade:

C1	C2	C1 e C2	C1 ou C2	Não C1
V	V	V	V	F
F	V	F	V	V
V	F	F	V	F
F	F	F	F	V

Operadores Lógicos

Exemplos:

- $(5 > 4) \text{ e } (4 > 4)$
- $A \leftarrow (5 > 4) \text{ e } (4 > 4)$
- $B \leftarrow \text{não } A$
- $C \leftarrow B \text{ ou } A$
- $\text{não } ((5 + 6) > (7 + 8))$
- $((5 * 2) > ((4 * 5) / 2)) \text{ ou não } A$

Estrutura de um Algoritmo

algoritmo *nome*

declaração das variáveis

inicio

comandos

fim

Declaração de Variáveis

- É realizada na área de declaração de variáveis
- Coloca-se o tipo da variável, seguido de uma ou mais variáveis separadas por vírgulas

algoritmo primeiro

inteiro num

real R1, R2

inicio

fim

Declaração de Variáveis

Exercício:

- Criar um algoritmo com variáveis para guardar o nome de um aluno, sua idade e duas notas.
- Além disso, criar uma variável que indica se o aluno foi aprovado ou não.
- Atribuir valores iniciais a essas variáveis.

Declaração de Variáveis

Exercício:

```
algoritmo segundo
    cadeia nome
    inteiro idade
    real P1, P2
    logico aprovado
início
    nome ← 'Sebastião da Silva'
    idade ← 42
    P1 ← 6
    P2 ← 8.5
    aprovado ← verdadeiro
fim
```

Comandos de Entrada e Saída

- É necessário que haja comandos para ler os dados de entrada, fornecidos pelo usuário
- Os cálculos são de pouco valor se o usuário não pode ver seus resultados
- Para a entrada de dados usaremos o comando ***ler***
 - Exemplo: ler a,b
- Para mostrar os resultados usaremos o comando ***escrever***
 - Exemplo: escrever ‘resposta:’, a

Exercício 1

- Escrever um algoritmo para ler o valor de dois números e escrever a soma

```
algoritmo soma
    real a1, a2, resultado
inicio
    ler a1
    ler a2
    resultado ← a1 + a2
    escrever 'soma:'
    escrever resultado
fim
```

Exercício 1

- Alternativamente (sintaxe também válida):

```
algoritmo soma
    real a1, a2, resultado
inicio
    ler a1,a2
    resultado ← a1 + a2
    escrever 'soma:', resultado
fim
```

Exercício 2

- Criar um algoritmo para ler três números e escrever a média aritmética desses números

```
algoritmo media_aritmetica
    real n1, n2, n3, media
início
    ler n1, n2, n3
    media ← (n1 + n2 + n3)/3
    escrever 'media:', media
fim
```

Exercício 3

- Criar um algoritmo para ler temperatura em Celsius e escrever em Farenheit

```
algoritmo conversao_temperatura
    real c, f
inicio
    ler c
     $f \leftarrow (c/5)*9 + 32$ 
    escrever 'temp. em Farenheit:', f
fim
```


Exercício 4

- Criar um algoritmo para ler dois valores inteiros e trocar o conteúdo desses valores (imprimir os valores antes e depois da troca)

algoritmo troca

inteiro a, b, aux

inicio

ler a, b

escrever 'a:' a, ' b:', b

aux \leftarrow a

a \leftarrow b

b \leftarrow aux

escrever 'a:' a, ' b:', b

fim

Exercício 5

- Criar um algoritmo para ler o valor do tempo em segundos e imprimir e hora, minuto e segundos, e.g, 4000s = 1h 6min 40s

algoritmo converte_segundos

inteiro tempo, h, min, seg, resto

inicio

ler tempo

$h \leftarrow \text{tempo} // 3600$

$\text{resto} \leftarrow \text{tempo} \% 3600$

$\text{min} \leftarrow \text{resto} // 60$

$s \leftarrow \text{resto} \% 60$

escrever h, 'h ', min, 'min ', s, 's'

fim

Simulação (Chinês)

- Para verificar se um algoritmo funciona deve-se simular sua execução com dados, de forma que os resultados são previamente conhecidos
- Por exemplo, simule o funcionamento do algoritmo anterior para tempo = 10000
- Dica: crie uma tabela e vá acompanhando os valores que cada variável recebe depois de cada comando

Outras funções disponíveis

■ Existem várias funções disponíveis, como por exemplo:

- ❑ SQRT (X) – Calcula a raiz quadrada de X
- ❑ SQR (X) – Eleva ao quadrado o valor de X
- ❑ SIN (X) – Calcula o seno de X
- ❑ COS (X) – Calcula o cosseno de X
- ❑ RAND (X) – Produz um número aleatório de 0 até o valor de X

Exercício 6

- Criar um algoritmo para ler os valores de A, B e C e imprimir as raízes da equação do segundo grau correspondente

algoritmo raizes

real A, B, C, delta, R1, R2

inicio

ler A, B, C

delta \leftarrow SQRT(SQR(B)-4*A*C)

R1 \leftarrow (-B+delta)/(2*A)

R2 \leftarrow (-B-delta)/(2*A)

escrever 'R1:', R1, ' R2:', R2

fim

Exercício 7

- Criar um algoritmo para ler valores em hora, minuto e segundo e transformar tudo para segundos.

```
algoritmo converte_segundos2
    inteiro h, min, seg, total_seg
inicio
    ler h, min, seg
    total_seg ← h*3600+min*60+seg
    escrever 'tempo: ', total_seg, ' seg'
fim
```