

# Processamento de Dados aplicado à Geociências

1

## AULA 5: Princípios Básicos Estruturas Chaves Sequenciação Decisão (Estruturas de Seleção)

UNIVERSIDADE FEDERAL DE PELOTAS  
CENTRO DE DESENVOLVIMENTO TECNOLÓGICO  
CURSO SUPERIOR DE TECNOLOGIA EM  
GEOPROCESSAMENTO

Professor: Guilherme Tomaschewski Netto  
[guilherme.netto@inf.ufpel.edu.br](mailto:guilherme.netto@inf.ufpel.edu.br)



# Princípios Básicos



# Interação com o usuário

- Em um algoritmo é muito importante que se considere que um usuário irá interagir em algum momento
- Por isso, é fundamental que algumas mensagens sejam escritas para que o usuário entenda
  - o que está acontecendo durante a execução do algoritmo
  - o que está sendo fornecido como saída do algoritmo

# Interação com o usuário

## Algoritmo

**Declare** Nota1, Nota2, Media **Numerico**

**Escreva** 'Entre com a primeira nota'

**Leia** Nota1

**Escreva** 'Entre com a segunda nota'

**Leia** Nota2

Media := (Nota1+Nota2)/2

**Se** Media >= 7

**então** **Escreva** 'Aprovado com nota', Media

**senão** **Escreva** 'Reprovado com nota', Media

**fim\_se**

**fim\_algoritmo**

# Identação

- Outro conceito importante em algoritmos é o de identificação.
- A identificação deve ser utilizada para mostrar a estrutura lógica e a sequência de comandos em um algoritmo

# Identação

## Algoritmo

```
  Declare Nota1, Nota2, Media Numerico  
  Escreva 'Entre com a primeira nota'  
  Leia Nota1  
  Escreva 'Entre com a segunda nota'  
  Leia Nota2  
  Media := (Nota1+Nota2) / 2  
  Se Media >= 7  
  |   então Escreva 'Aprovado com nota', Media  
  |   senão Escreva 'Reprovado com nota', Media  
  |   fim_se  
fim_algoritmo
```

# Comentários

- Quando escrevemos um algoritmo, o ideal é que sejam usados comentários para que se possa descrever o funcionamento do mesmo
- Existem algumas formas de representar comentários, entre elas {}, //, /\* \*/
  - {isto é um comentário}
  - // isto é um comentário
  - /\* isto é um comentário \*/

# Comentários

## Algoritmo

```
//declaração de variáveis  
Declare Nota1, Nota2, Media Numerico  
Escreva 'Entre com a primeira nota'  
Leia Nota1  
Escreva 'Entre com a segunda nota'  
Leia Nota2  
Media := (Nota1+Nota2)/2 /*cálculo da média*/  
Se Media >= 7  
    então Escreva 'Aprovado com nota', Media  
    senão Escreva 'Reprovado com nota', Media  
fim_se  
fim_algoritmo
```

# Estruturas Chaves



## Estruturas Chaves da Construção de Algoritmos

- Existem 3 estruturas básicas de controle nas quais se baseiam os algoritmos:
  - Sequenciação
  - Decisão
  - Repetição
- A utilização combinada dessas 3 estruturas descritas vai permitir expressar, usando qualquer que seja a ferramenta, a solução para uma gama muito grande de problemas
- Todas as linguagens de programação oferecem representantes dessas estruturas

# Sequenciação

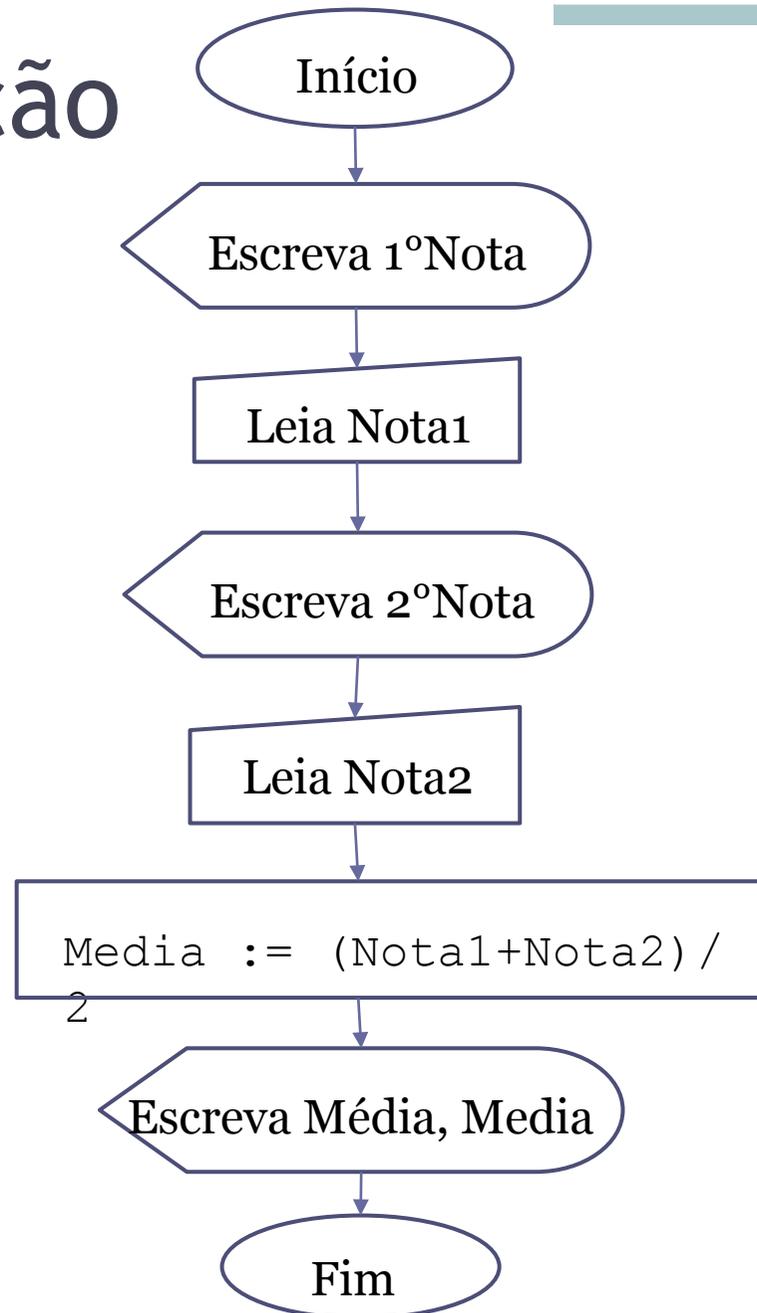
- Os comandos do algoritmo fazem parte de uma sequência, onde é relevante a ordem na qual se encontram os mesmos, pois serão executados um de cada vez, estritamente, de acordo com essa ordem
- A forma geral da sequenciação é dada por:
  - Algoritmo
    - Comando-1;
    - Comando-2;
    - :
    - Comando-n;
  - fim\_algoritmo

# Sequenciação

## Algoritmo

```
Declare Nota1, Nota2, Media Numerico  
Escreva 'Entre com a primeira nota'  
Leia Nota1  
Escreva 'Entre com a segunda nota'  
Leia Nota2  
Media := (Nota1+Nota2)/2  
Escreva 'Média final', Media  
fim_algoritmo
```

# Sequenciação



# Sequenciação

Início
Escreva 'Entre com primeira a nota'
Leia Nota1
Escreva 'Entre com a segunda nota'
Leia Nota2
$Media := (Nota1+Nota2)/2$
Escreva 'Média final', Media
Fim

# Estruturas de Decisão

- A estrutura de decisão executa uma sequência de comandos de acordo com o resultado de um teste
- É a estrutura mais flexível, podendo acontecer de várias formas, dependendo do fato de a condição ser satisfeita ou não e do fato do desvio de fluxo poder ou não ter outros comandos antes da finalização da estrutura
- Pode-se utilizar todos os operadores relacionais para as comparações. Entre as condições podem ainda ser utilizados operadores lógicos ( E, OU, NÃO)

# Estruturas de Seleção (condicional)

- Na vida real tomamos decisões a todo o momento baseadas em uma situação existente. Em um algoritmo, chamamos esta situação de condição. Associada a uma condição, existirá uma alternativa possível de ações.

## Exemplo

- “Se tiver R\$ 10,00 sobrando, então irei ao cinema hoje à noite.”
- A condição nesta frase é “tiver R\$ 10,00 sobrando”. Ela é uma expressão lógica, pois a pergunta “Tenho R\$ 10,00 sobrando?” Pode (tem que) ser respondida com “verdadeiro” ou “falso”.
- Se a condição for verdadeira, a ação a ser executada é “irei ao cinema”.

# Exemplo

- Vamos colocar agora a frase do exemplo anterior em outra forma, mais parecida com nosso Português Estruturado:
  - se “tiver R\$ 10,00 sobrando”
  - então “irei ao cinema”
  - fim\_se

# Estrutura de seleção simples

- Sintaxe:
  - Se <condição>
  - então <lista de comandos>
  - fim\_se
- onde <condição> é uma expressão lógica

# Estrutura de seleção simples

- Exemplo
- Comparação entre dois números
  - Se  $A > B$
  - então Escreva 'O número maior é', A

# Estrutura de seleção simples

- Exemplo
- Comparação entre três números
  - Se  $A > B$  E  $A > C$
  - então Escreva 'O número maior é', A

# Estrutura de seleção simples

## Algoritmo

```
Declare Nota1, Nota2, Media Numerico  
Escreva 'Entre com a primeira nota'  
Leia Nota1  
Escreva 'Entre com a segunda nota'  
Leia Nota2  
Media := (Nota1+Nota2) / 2  
Se Media >= 7  
    então Escreva 'Aprovado com nota', Media  
fim_se  
fim_algoritmo
```

# Estrutura de seleção composta

- O exemplo inicial poderia ser estendido para o caso do sujeito não ter dinheiro sobrando:
  - “Se tiver R\$ 10,00 sobrando, irei ao cinema hoje à noite, mas se não tiver ficarei vendo TV em casa”.
- Em português estruturado...
  - se “tiver R\$ 10,00 sobrando”
  - então “irei ao cinema”
  - senão “ficarei vendo TV em casa”
  - fim\_se

# Estrutura de seleção composta

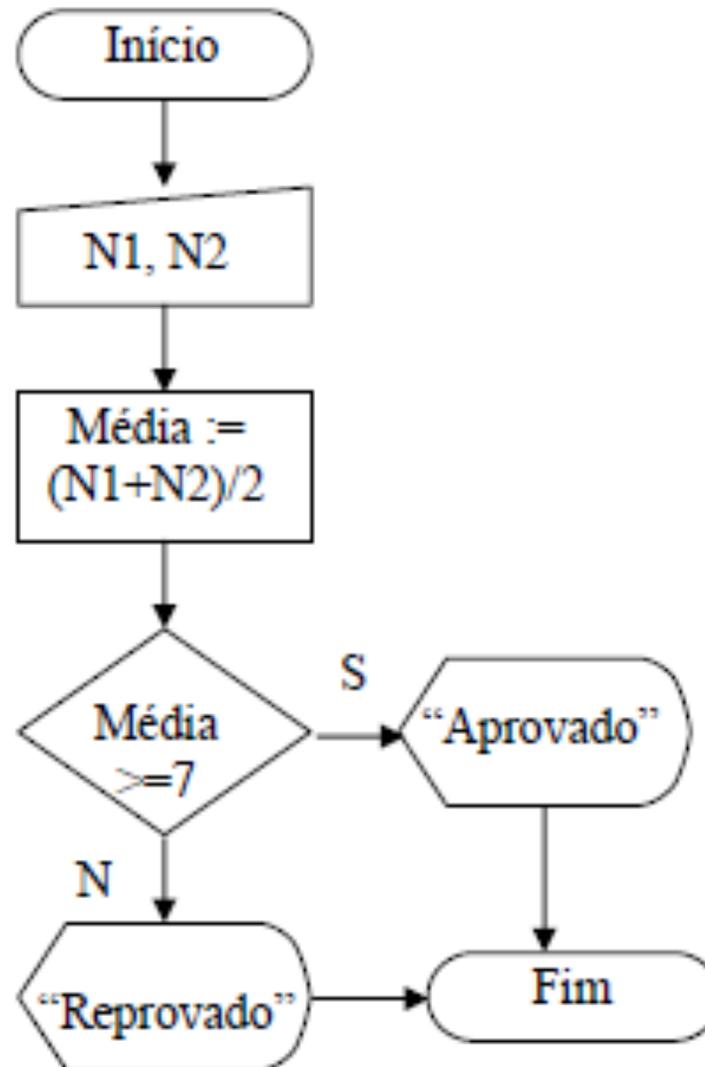
- Sintaxe:
  - Se <condição>
  - então <lista\_comandos>
  - senão <lista\_comandos>
  - fim\_se
- onde <condição> é uma expressão lógica

# Estrutura de seleção composta

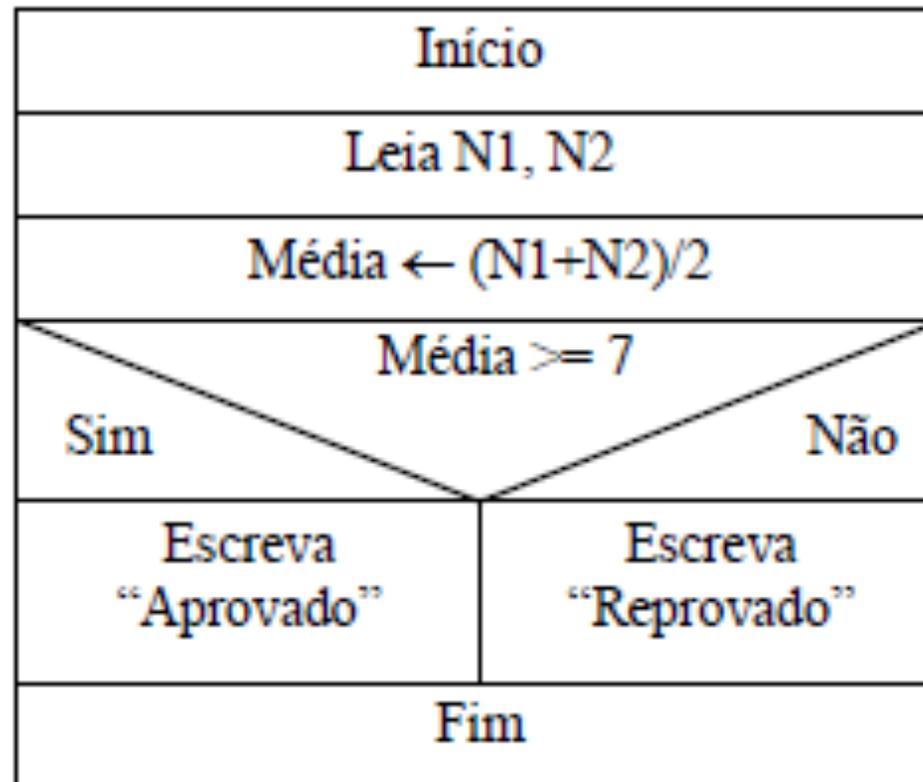
## Algoritmo

```
Declare Nota1, Nota2, Media Numerico  
Escreva 'Entre com a primeira nota'  
Leia Nota1  
Escreva 'Entre com a segunda nota'  
Leia Nota2  
Media := (Nota1+Nota2)/2  
Se Media >= 7  
    então Escreva 'Aprovado com nota', Media  
    senão Escreva 'Reprovado com nota', Media  
fim_se  
fim_algoritmo
```

# Estrutura de seleção composta



# Estrutura de seleção composta



# Estrutura de seleção encadeada

- A estrutura de seleção encadeada é também conhecida como “ninhos de se”
- Ocorre quando tem-se uma incidência de uma estrutura de seleção dentro de outra estrutura de seleção

# Estrutura de seleção encadeada

- Sintaxe:
  - Se <condição>
  - então <lista\_comandos>
  - senão Se <condição>
  - então <lista\_comandos>
  - senão <lista\_comandos>
  - fim\_se
  - fim\_se

# Exemplo

- Algoritmo que lê 3 números e escreve o maior

**Algoritmo**

**Declare A,B,C Numerico**

**Leia A,B,C**

**Se** A>B **E** A>C

**então Escreva** 'O maior é', A

**senão Se** B>A **E** B>C

**então Escreva** 'O maior é', B

**senão Escreva** 'O maior é', C

**fim\_se**

**fim\_se**

**fim\_algoritmo**

# Estruturas de Condição

- A estrutura de condição equivale a um ninho de “Se”
- Sintaxe
  - Faça\_caso
  - Caso <<Condição 1>>
  - <lista\_comandos>
  - Caso <<Condição n>>
  - <lista\_comandos>
  - Outros Casos
  - <lista\_comandos>
  - Fim\_de\_caso

# Estruturas de Condição

## Algoritmo

Declare A,B,C Numerico

Leia A,B,C

### Faça Caso

**Caso** A>B **E** A>C

**Escreva** 'O maior é', A

**Caso** B>A **E** B>C

**Escreva** 'O maior é', B

### Outros Casos

**Escreva** 'O maior é', C

**fim\_de\_caso**

**fim\_algoritmo**

# Exercícios

- Lista de exercícios 3