

# Algoritmos e Programação

Daniel de Sousa Moraes  
danielmoraes14@gmail.com

# Ementa

- Conceitos de Computação e Computadores
  - Origens da computação
  - A evolução dos computadores
  - Representação da informação
  - Arquitetura de um computador
- Introdução à Lógica de Programação
  - O que é Lógica
  - O que é um Algoritmo

# Ementa

- Algoritmos e fluxogramas
  - Aplicabilidade, propriedades
  - Tipos de dados
  - Variáveis
  - Expressões aritméticas e lógicas
  - Atribuição
  - Estruturas de Controle
  - Estruturas de Dados
  - Modularização e Recursividade
  - Manipulação de Arquivos

# Estruturas de Controle

- Como temos visto o fluxo de execução de um algoritmo é sequencial.
- As operações são executadas sequencialmente na ordem que são definidas no código.

# Estruturas de Controle

- Assim como nos fluxogramas que possui diferentes símbolos que expressam diferentes tipos de operação, podemos classificar as estruturas de controle de um código como:
  - **Estruturas sequenciais:** representam ações imperativas, sem nenhum tipo de decisão. A operação simplesmente é executada.
  - **Estruturas condicionais (ou decisão):** representam desvio condicionais no fluxo, dependendo do resultado de uma expressão lógica dada como condição.

# Estruturas de Controle

- **Estruturas de repetição:** representam a execução repetitiva de instruções existentes.

# Estruturas Sequenciais

- Os comandos (instruções) são executados imperativamente, sem desvio algum de caminho.

## Início

**inteiro** num

**leia** (num)

num  $\leftarrow$  num + 1

**escreve** (num)

## Fim

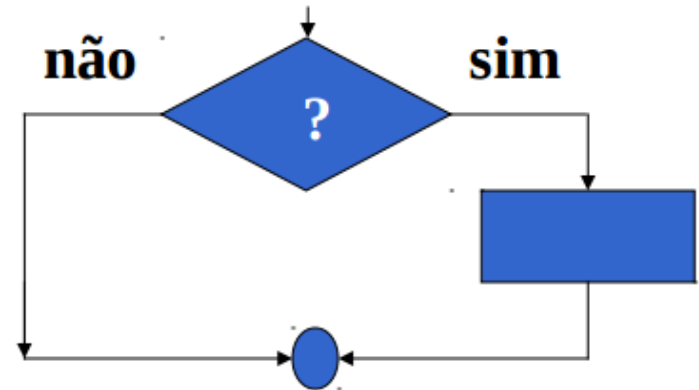
# Estruturas de Condição (Decisão)

- Permitem a tomada de uma decisão sobre qual o caminho a ser escolhido, de acordo com o resultado de uma expressão lógica.
- Existem três formas básicas: Seleção Simples, Composta e múltipla escolha.



# Estruturas de Decisão Simples

- Quando a condição avaliada é verdadeira, então a sequência de operações é executada. Caso contrário, há um desvio sem comando algum.
- Comandos:  
**Se <condição> então faça**  
    <instruções> ..  
**Fim Se**



# Estruturas de Decisão Composta

- Quando a condição avaliada é verdadeira, então a sequência de operações é executada. Caso contrário, há um desvio para um outro comando. Apenas UM comando é executado.
- Comandos:

**Se <condição> então faça**  
    <instruções>

**Senão faça**  
    <instruções>

**Fim Se**

**Se <condição1> então faça**

**Se <condição2> então faça**  
    <instruções>

**Fim Se**

**Fim Se**

# Exemplo – Condição Simples

Início

```
real nota1, nota2, nota3, media
```

```
leia (nota1, nota2, nota3)
```

```
media ← (nota1+nota2+nota3)/3
```

```
Se (media >= 7) então faça
```

```
    escreve ('Aprovado')
```

```
Fim Se
```

```
Se (media < 7) então faça
```

```
    escreve ('Reprovado')
```

```
Fim Se
```

Fim

# Exemplo – Condição Composta

Inicio

```
real nota1, nota2, nota3, media
```

```
leia (nota1, nota2, nota3)
```

```
media ← (nota1+nota2+nota3)/3
```

```
Se (media >= 7) então faça
```

```
    escreve ('Aprovado')
```

```
Senão faça
```

```
    escreve ('Reprovado')
```

```
Fim Se
```

Fim

# Exemplo – Condições aninhadas

Início

```
real nota1, nota2, nota3, media
```

```
leia (nota1, nota2, nota3)
```

```
media ← (nota1+nota2+nota3)/3
```

```
Se (media >= 7) então faça
```

```
    escreve ('Aprovado')
```

```
Senão faça
```

```
    Se (media >= 5) então faça
```

```
        escreve ('Recuperação')
```

```
    Senão faça
```

```
        escreve ('Reprovado')
```

```
    Fim Se
```

```
Fim Se
```

```
Fim
```

# Exemplo – Condições aninhadas

Início

```
real a, b, c,delta, x1, x2
leia (a, b, c)
Se (a != 0) então faça
    delta ← QUAD(b)-4*a*c
    Se (delta < 0) então faça
        escreve ('Não possui raiz')
    Senão faça
        Se (delta > 0) então faça
            x1 ← -b+RAIZ(delta)/2*a
            x2 ← -b-RAIZ(delta)/2*a
            escreve ('As raízes são',x1, x2)
        Senão faça
            x1 ← -b+RAIZ(delta)/2*a
            escreve ('A raíz é',x1)
    Fim Se
Fim Se
Senão faça
    escreve ('Não é equação do 2º grau')
Fim Se
```

Fim

# Exercícios

- Sabemos que das 6 da manhã a meio-dia desejamos 'Bom dia!', assim como de meio-dia a 6 da noite desejamos 'Boa tarde!' e daí até o amanhecer desejamos 'Boa noite!'. Faça um programa que pede ao usuário para informar a hora certa (de 0 a 24) e exibe uma das saudações acima como resultado.
- Faça um programa que leia 3 números reais e imprima o maior e o menor deles

# Estruturas de Múltipla Escolha

- Condições aninhadas (encadeadas) são bastante frequentes
- Para simplificar a escrita pode-se usar o comando **Escolha**
- Permite escolher um entre vários valores possíveis para uma variável

**Escolha** (expressão ou variável)

caso <valor 1>:

    <instruções>

caso <valor 3>:

    <instruções>

.....

senão

    <instruções>

**Fim Escolha**



# Estruturas de Múltipla Escolha

- Na seleção múltipla, a expressão é calculada e os comandos relacionados com o teste de mesmo valor da expressão são executados.
- Se não houver valor igual ao da expressão, os comandos subordinados à palavra senão são executados.
- A cláusula senão é opcional.
  - Se ela não existir e o valor da expressão não for igual a nenhum dos testes, nenhum comando da estrutura é executado.
- A seleção é exclusiva
  - No máximo uma das opções é executada.

# Exemplo - Escolha

**Inicio**

**inteiro** num

**leia** (num)

**Escolha** (num)

**caso 1: escreve** ('primeiro')

**caso 2: escreve** ('segundo')

**caso 3: escreve** ('terceiro')

**senão faça** ('fora do pódio')

**Fim Escolha**

**Fim**

# Exercício

- Um Jedi é um ser especial dotado de capacidade de controlar a força (vide Star Wars). Assim que um Jedi é descoberto, um mestre o guia em vários níveis: Faça um programa que leia do usuário um número inteiro representando seu nível e informa seu respectivo título.

Nível	Título
0	Descoberto
1	Padawan
2	Cavaleiro Jedi
3	Mestre Jedi

# Referências

SOUZA, Marco Antonio Furlan de et al. **Algoritmos e lógica de programação: um texto introdutório para Engenharia**. 2ª ed rev. e ampl. -- São Paulo: Cengage Learning, 2011. ISBN 978-85-221-1129-9

FORBELLONE, André Luiz Villar EBERSPACHER, Henri Frederico. **Lógica de programação: a construção de algoritmos e estruturas de dados**. 3ª ed – São Paulo: Prentice Hall, 2005.

MANZANO, José Augusto N. G.; OLIVEIRA, Jayr Figueiredo de. **Algoritmos: lógica para desenvolvimento de programação de computadores** . 21. ed. São Paulo: Érica, 2008. 240 p. ISBN 978-85-7194-718-4(broch)

Carlos de Salles Soares Neto – Notas de Aula da Disciplina de Algoritmos I - UFMA