



**UNIPAC**

Universidade Presidente Antônio Carlos

*Bacharelado em Ciência da Computação*

---

# *Introdução a Programação*

## **Material de Apoio**

*Parte IV*

*Prof. Nairon Neri Silva*

naironsilva@unipac.br

*1º sem / 2020*

*Material cedido pela professora Livia*

# Estruturas de Controle

---

- Para que o conjunto de ações (comandos a serem executados em um algoritmo) se torne viável, deve existir uma perfeita relação lógica intrínseca ao modo pelo qual essas ações são executadas, ao modo pelo qual é regido o ***fluxo de execução*** do algoritmo.
- Solução dos problemas através das estruturas básicas de controle do fluxo de execução: sequenciação, seleção e repetição. Bem como a combinação dessas estruturas.

# Estruturas de Seleção (Decisão/Condicionais)

---

- Com o conhecimento adquirido até aqui, ainda não é possível tomar decisões. Como escrever algoritmos que selecionem o que fazem?
- Um estrutura de seleção permite a escolha de um grupo de ações (bloco) a ser executado quando determinadas condições (representadas por expressões lógicas ou relacionais) são ou não satisfeitas.

# Estruturas de Seleção

---

- Os tipos de estruturas de seleção são:
  - Seleção simples
  - Seleção composta
  - Seleção múltipla

# Estrutura de Seleção Simples

---

- Na estrutura de seleção simples uma expressão lógica é avaliada, e antes de executar uma ação essa expressão é testada e analisada.
- Sintaxe:

```
if (condição)  
    <comando>;
```

# Estrutura de Seleção Simples

---

- Onde:
  - <condição> é uma expressão lógica que, quando inspecionada, pode gerar um resultado falso ou verdadeiro.
  - <comando> é a ação/comando a ser executado quando a <condição> tiver valor verdadeiro.
  - se a <condição> for verdadeira o comando será executado; caso contrário (<condição> for falsa) encerra-se a seleção.

# Estrutura de Seleção Simples

---

- Quando houver mais de um comando a ser executado quando a condição for verdadeira deve-se utilizar chaves ( { } ) para indicar o início e o término daquele bloco de comandos.

```
if (condição)
{
    <comando>;
    <comando>;
    ...
    <comando>;
}
```

*Caso haja somente um comando o uso de chaves é opcional!*

# Exemplo

---

- Elaborar um algoritmo para determinar se um aluno foi aprovado, onde a média para aprovação é 60%, e o nome e a nota do aluno serão informados pelo usuário.

```
void main()
{
    float nota;

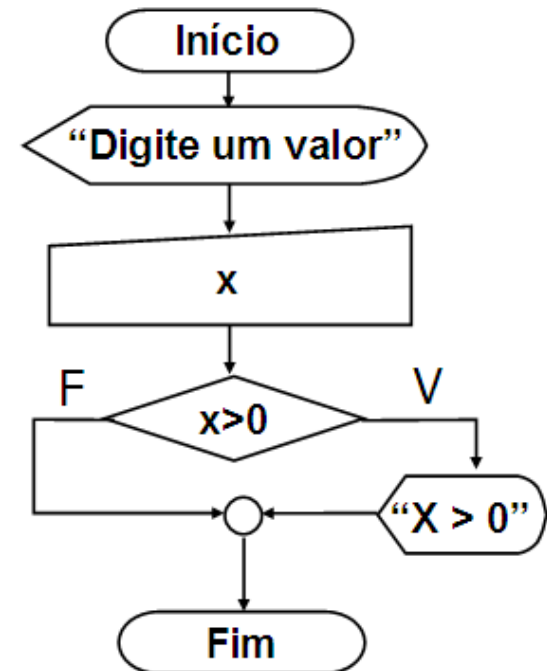
    scanf("%f", &nota);
    if (nota >= 60)
        printf("O aluno foi aprovado!");
}
```



# Exemplo

- Algoritmo para verificar se um número informado pelo usuário é maior que zero.

```
void main()  
{  
    int x;  
    printf("Digite um valor: ");  
    scanf("%d", &x);  
    if (x > 0)  
        printf("O n° é maior que 0!");  
}
```



# Estrutura de Seleção Composta

---

- Além de executar comandos se a condição for verdadeira, a estrutura de seleção composta também executa comandos se a condição for falsa.
- Sintaxe:

```
if (condição)
    <comando>;
else
    <comando>;
```

# Estrutura de Seleção Composta

---

- Quando houver mais de um comando a ser executado quando a condição for verdadeira ou falsa deve-se utilizar chaves ( { } ) para indicar o início e o término daquele bloco de comandos.

```
if (condição)
{
    <comando>;
    <comando>;
}
else
{
    <comando>;
    <comando>;
}
```

*Caso haja somente um comando o uso de chaves é opcional!*

# Exemplo

---

- Elaborar um algoritmo para determinar se um aluno foi aprovado ou reprovado, onde a média para aprovação é 60%.

```
void main()
{
    float nota;
    scanf("%d", &nota);
    if (nota >= 60)
        printf("O aluno foi aprovado!");
    else
        printf("O aluno foi reprovado!");
}
```

# Estrutura de Seleção Múltipla

---

- Comando **SWITCH**

Sintaxe:

```
switch (expressão) {  
    case constante1:  sequência de comandos  
                      break;  
    case constante2:  sequência de comandos  
                      break;  
    ...  
    default: sequência de comandos  
}
```

# Estrutura de Seleção Múltipla

---

- Comando **SWITCH**

Exemplo:

```
switch (i) {  
    case 0: printf("Voce informou o numero 0");  
            break;  
    case 1: printf("Voce informou o numero 1");  
            break;  
    default: printf("Numero diferente de 0 e 1");  
}
```

# Estrutura de Controle de Fluxo - Seleção

---

- Comando de seleção: ?

Pode-se utilizar o operador ? para substituir comandos if-else na forma geral.

Sintaxe: *Exp1 ? Exp2 : Exp3*

- Exemplo:

Utilizando IF:

```
if (x>9) y = 100;  
else y = 200;
```

Utilizando ?:

```
y = x > 9 ? 100 : 200;
```