

# MCTA028-15: Programação Estruturada

## Aula 2: Estruturas de Seleção

*Wagner Tanaka Botelho*

*wagner.tanaka@ufabc.edu.br / wagtanaka@gmail.com*

*Universidade Federal do ABC (UFABC)*

*Centro de Matemática, Computação e Cognição (CMCC)*

# Introdução

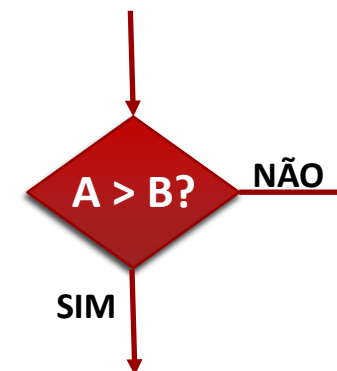
- Vamos estudar a estrutura condicional simples (`if`), composta (`if .. else`), aninhada e `switch`;
- Vamos utilizar (`if` e `if .. else`) os operadores **RELACIONAIS** e **LÓGICOS** que foram explicados na aula anterior:

## Relacionais

Operador	Significado	Exemplo
>	Maior do que	<code>x&gt;5</code>
>=	Maior ou igual a	<code>x&gt;=10</code>
<	Menor do que	<code>x&lt;5</code>
<=	Menor ou igual a	<code>x&lt;=10</code>
==	Igual a	<code>x==0</code>
!=	Diferente de	<code>x!=0</code>

## Lógicos

Operador	Significado	Exemplo
&&	Operador E	<code>(x &gt; 0 &amp;&amp; x &lt; 10)</code>
	Operador OU	<code>(a == 'F'    b != 32)</code>
!	Operador NEGAÇÃO	<code>!(x == 10)</code>



# Condiciona! Simples: if

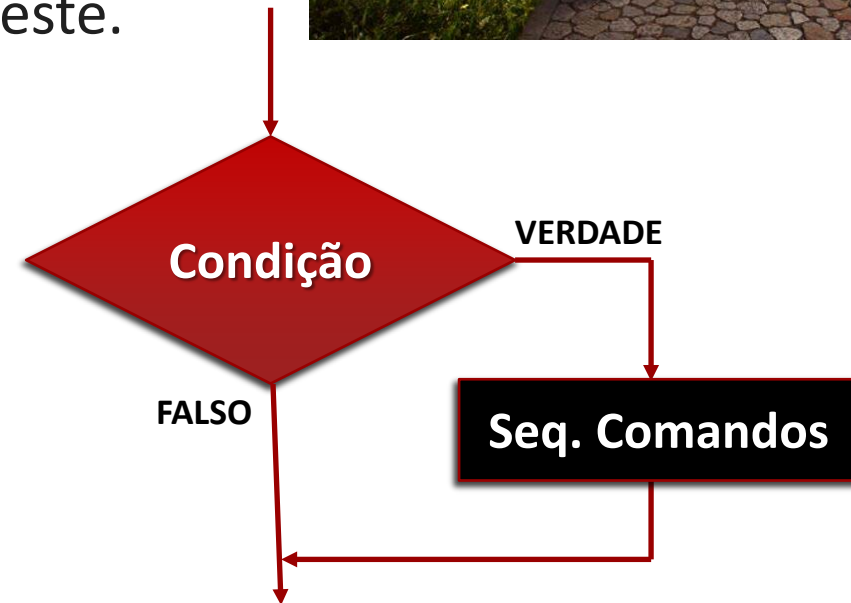
# Condicional Simple: `if`

- O `if` é usado sempre que :
  - É necessário **escolher** entre **dois caminhos** dentro do programa;
  - Quando se deseja executar **um** ou **mais** comandos que estejam **sujeitos** ao resultado de um teste.



- Sintaxe em C:

```
if(condição)
{
    sequência de comandos;
}
```

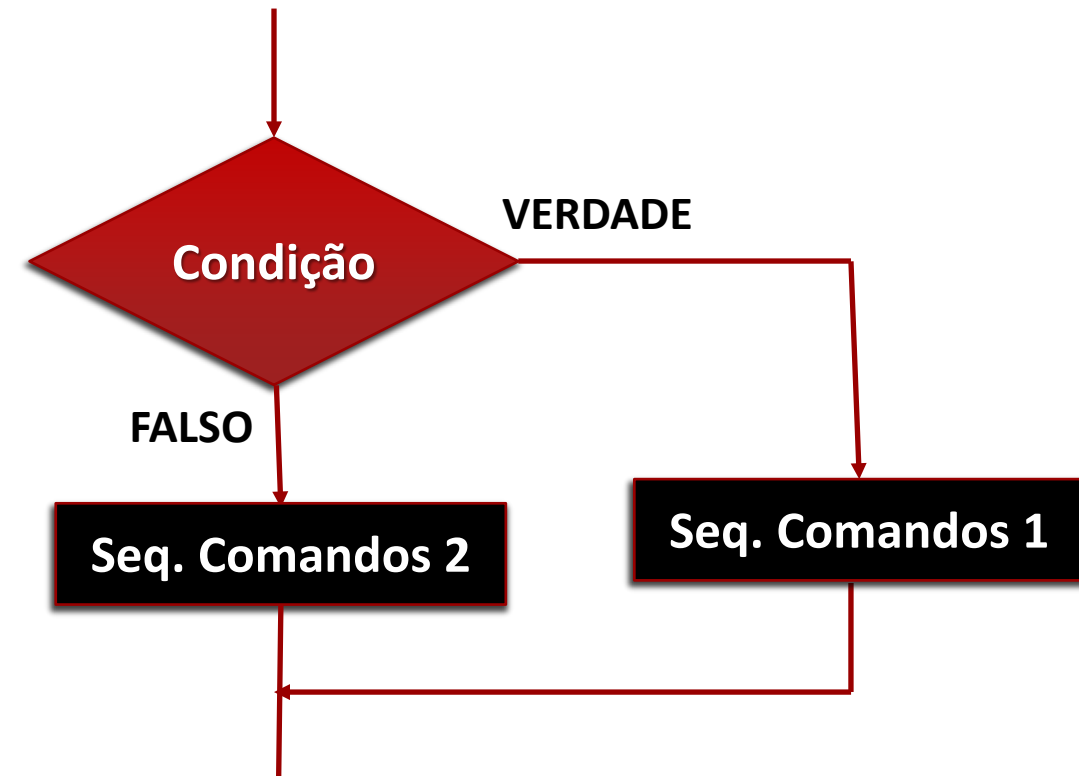


**Condiciona! Composta: if . . . else**

# Condicional Composta: `if...else`

- O comando `else` pode ser entendido como um **complemento** do comando `if`;
- O **objetivo** do `else` é auxiliar o comando `if` na tarefa de **escolher** entre os vários caminhos a serem seguidos dentro do algoritmo.
- Sintaxe em C:

```
if (condição)
{
    sequência de comandos 1;
}
else
{
    sequência de comandos 2;
}
```



Ex\_07.c X

```
1  #include <stdio.h>
2
3  void main()
4  {
5      int num;
6
7      printf("Digite um numero: ");
8      scanf("%d", &num);
9
10     if(num > 10)
11     {
12         printf("O numero eh MAIOR que 10\n");
13     }
14     else
15     {
16         printf("O numero eh MENOR que 10\n");
17     }
18 }
```

Ex\_08.c X

```
1  #include <stdio.h>
2
3  void main()
4  {
5      int num;
6
7      printf("Digite um numero: ");
8      scanf("%d", &num);
9
10     if(num > 10)
11         printf("O numero eh MAIOR que 10\n");
12     else
13         printf("O numero eh MENOR que 10\n");
14 }
```

# Estruturas `if...else` Aninhadas



# Estruturas if...else Aninhadas

- O objetivo é testar **vários casos**;
- Por exemplo, implemente um algoritmo para imprimir o **conceito final** de um aluno, considerando as seguintes informações:

**A: nota  $\geq 9.0$**

**B:  $7.5 \leq \text{nota} < 9.0$**

**C:  $6.0 \leq \text{nota} < 7.5$**

**D:  $5.0 \leq \text{nota} < 6.0$**

**F: nota  $< 5.0$**

```
*Ex_11.c X
1  #include <stdio.h>
2
3  void main()
4  {
5      float nota;
6
7      printf("Digite a nota do aluno: ");
8      scanf("%f", &nota);
9
10     if(nota >= 9){
11         printf("A");
12     }
13     else if(nota >= 7.5){
14         printf("B");
15     }
16     else if(nota >= 6){
17         printf("C");
18     }
19     else if(nota >= 5){
20         printf("D");
21     }
22     else{
23         printf("F");
24     }
25 }
```

```
1  #include <stdio.h>
2
3  void main()
4  {
5      float nota;
6
7      printf("Digite a nota do aluno: ");
8      scanf("%f", &nota);
9
10     if(nota >= 9){
11         printf("A");
12     }
13     else if(nota >= 7.5){
14         printf("B");
15     }
16     else if(nota >= 6){
17         printf("C");
18     }
19     else if(nota >= 5){
20         printf("D");
21     }
22     else{
23         printf("F");
24     }
25 }
```

D:\UFABC\Disciplinas\2021-2025\Q1\PE\Aulas\02\Codigos\Aula02\b

Digite a nota do aluno: 9,3

A

Process returned 65 (0x41) execution time : 3.110 s

Press any key to continue.

D:\UFABC\Disciplinas\2021-2025\Q1\PE\Aulas\02\Codigos\Aula02\b

Digite a nota do aluno: 6,7

C

Process returned 67 (0x43) execution time : 3.954 s

Press any key to continue.

D:\UFABC\Disciplinas\2021-2025\Q1\PE\Aulas\02\Codigos\Aula02\b

Digite a nota do aluno: 4,8

F

Process returned 70 (0x46) execution time : 4.121 s

Press any key to continue.

# Comando switch

# Comando switch

- O comando `switch` é muito mais limitado que o comando `if-else`;
- Enquanto o comando `if` pode testar **expressões lógicas** ou **relacionais**, o `switch` somente verifica se uma **variável** (do tipo `int` ou `char`) **é** ou **não igual** a certo valor **constante**.

- Sintaxe em C:

```
switch (variável)
{
    case valor1:
        sequência de comandos;
        break;
    case valor2:
        sequência de comandos;
        break;
    case valorN:
        sequência de comandos;
        break;
    default:
        sequência de comandos;
}
```

# Comando switch

## ➤ Sintaxe em C:

```
switch (variável)
{
    case valor1:
        sequência de comandos;
        break;
    case valor2:
        sequência de comandos;
        break;
    case valorN:
        sequência de comandos;
        break;
    default:
        sequência de comandos;
}
```

- O default é **opcional**, e sua sequência de comandos somente será executada se o **valor da variável** que está sendo testada pelo switch **NÃO** for **igual** a nenhum dos valores dos comandos case;
- O break é **opcional** e faz com que o switch seja **interrompido** assim que uma das sequências de comandos é executada.

```
1  #include <stdio.h>
2
3  void main()
4  {
5      char ch;
6
7      printf("Digite um simbolo de pontuacao: ");
8      scanf("%c", &ch);
9
10     switch(ch)
11     {
12         case '.':
13             printf("Ponto. \n");
14             break;
15         case ',':
16             printf("Virgula. \n");
17             break;
18         case ':':
19             printf("Dois pontos. \n");
20             break;
21         case ';':
22             printf("Ponto e virgula.\n");
23             break;
24         default:
25             printf("Nao eh pontuacao.\n");
26     }
27 }
```

D:\UFABC\Disciplinas\2021-2025\Q1\PE\Aulas\02\Codigos\Aula02\b

Digite um simbolo de pontuacao: ,  
Virgula.

Process returned 0 (0x0) execution time : 6.808 s  
Press any key to continue.

D:\UFABC\Disciplinas\2021-2025\Q1\PE\Aulas\02\Codigos\Aula02

Digite um simbolo de pontuacao: A  
Nao eh pontuacao.

Process returned 0 (0x0) execution time : 2.296 s  
Press any key to continue.

```
1  #include <stdio.h>
2
3  void main()
4  {
5      char ch;
6
7      printf("Digite um simbolo de pontuacao: ");
8      scanf("%c", &ch);
9
10     switch(ch)
11     {
12         case '.':
13             printf("Ponto. \n");
14         case ',':
15             printf("Virgula. \n");
16         case ':':
17             printf("Dois pontos. \n");
18         case ';':
19             printf("Ponto e virgula.\n");
20         default:
21             printf("Nao eh pontuacao.\n");
22     }
23 }
```

D:\UFABC\Disciplinas\2021-2025\Q1\PE\Aulas\02\Codigos\Aula02

```
Digite um simbolo de pontuacao: ,
Virgula.
Dois pontos.
Ponto e virgula.
Nao eh pontuacao.
```

```
Process returned 0 (0x0)   execution time : 4.253 s
Press any key to continue.
```

# Referências

- Slides do Prof. Luiz Rozante;
- SALES, André Barros de; AMVAME-NZE, Georges. Linguagem C: roteiro de experimentos para aulas práticas. 2016;
- BACKES, André. Linguagem C Completa e Descomplicada. Editora Campus. 2013;
- SCHILDT, Herbert. C Completo e Total. Makron Books. 1996;
- DAMAS, Luís. Linguagem C. LTC Editora. 1999;
- DEITEL, Paul e DEITEL, Harvey. C Como Programar. Pearson. 2011.