



<http://www.nickgentry.com/>

## **Algoritmos e Estruturas de Dados**

### **Disciplina 301477**

Programa de Pós-graduação em  
Computação Aplicada

**Prof. Alexandre Zaghetto**  
<http://alexandre.zaghetto.com>  
[zaghetto@unb.br](mailto:zaghetto@unb.br)

Universidade de Brasília  
Instituto de Ciências Exatas  
Departamento de Ciência da Computação



O presente conjunto de *slides* não pode ser reutilizado ou republicado sem a permissão do instrutor.

# **Módulo 03**

## **Algoritmos com Alternativas**

---

## 1. Algoritmos com Alternativas

- Na vida real tomamos decisões a partir da análise de algumas condições.

✓ Exemplo: Se eu tiver pelo menos R\$ 50,00 (na conta corrente ou poupança), então irei ao cinema hoje à noite.

Trata-se de uma expressão lógica, uma vez que a pergunta: “Tenho R\$ 50,00 sobrando?” Pode ser respondida com um “Sim” ou com um “Não”.

➤ Analogamente, em linguagens de programação um determinado bloco básico de comandos **será executado ou não**, dependendo da avaliação de expressões lógico-aritmético-relacionais. A isso chamados de **algoritmo com alternativa**.

## 2. Relembrando o Tipo Lógico

- O tipo lógico não é um tipo básico em Linguagem C.
  - ✓ **0** está para **FALSO**.
  - ✓ Qualquer outro valor está para **VERDADEIRO**.

### 3. Operadores Lógicos

NOT

A	X
0	1
1	0

$\neg A$

AND

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

$A \&\& B$

OR

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

$A || B$

### 3. Operadores Lógicos

Operador	Ação
!	NAO (Negação)
&&	E (Conjunção)
	OU (Disjunção inclusiva)

! 0 && 0 || 0  
! (0 && 0) || 0

**ATENÇÃO:** O operador “!” tem mais alta precedência que qualquer operador aritmético. Já os operadores “&&” e “||” têm mais baixa precedência.

### 3. Operadores Relacionais

Operador	Ação
>	Maior que
>=	Maior ou igual a
<	Menor que
<=	Menor ou igual
==	Igual
!=	Diferente

$A > B+8$

**ATENÇÃO:** Esses operadores são menores em precedência que qualquer operador aritmético.



## 4. Precedência entre operadores lógicos e relacionais

Operador	Precedência
!	Mais alta
> >= < <=	.
== !=	.
&&	.
	Mais baixa

10 > 5 && !(10 < 9) || 3 <=4

## 5. Alternativa Simples

- Forma geral em C:

```
if (<expressão>) {  
    <comandos>  
}
```

## 5. Alternativa Simples

**Exemplo 1:** Escreva um algoritmo para calcular a área de um círculo, fornecido o valor do raio, **que deve ser positivo**. Em seguida o algoritmo deve imprimir o valor da área na tela do computador.



## 5. Alternativa Simples

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    float Area, Raio;

    printf("Informe o valor do raio:");
    scanf("%f", &Raio);

    if (Raio > 0) {
        Area = 3.14*pow(Raio,2);
        printf("A area eh: %.2f \n",Area);
    }
    if (Raio <= 0) {
        printf("Raio não pode ser nulo ou negativo! \n");
    }

    return 0;
}
```

## 6. Alternativa Composta

- Forma geral em C:

```
if (<expressão>) {  
    |  
    |    <comandos>  
    |  
} else {  
    |  
    |    <comandos>  
    |  
}
```



## 6. Alternativa Composta

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    float Area, Raio;

    printf("Informe o valor do raio:");
    scanf("%f", &Raio);

    if (Raio > 0) {
        Area = 3.14*pow(Raio,2);
        printf("A area eh: %.2f \n",Area);
    } else {
        printf("Raio não pode ser nulo ou negativo! \n");
    }

    return 0;
}
```

## 7. Alternativas Aninhadas

- Forma geral em C:

```
if (<expressão>) {  
    <comandos>  
    if (<expressão>) {  
        <comandos>  
    } else {  
        <comandos>  
    }  
    <comandos>  
} else {  
    <comandos>  
    if (<expressão>) {  
        <comandos>  
    } else {  
        <comandos>  
    }  
    <comandos>  
}
```

## 7. Alternativas Aninhadas

- Forma geral em C:

```
if (<expressão>) {  
|     <comandos>  
} else if (<expressão>) {  
|     <comandos>  
} else if (<expressão>) {  
|     <comandos>  
} else if (<expressão>) {  
|     <comandos>  
} else {  
|     <comandos>  
}
```



## 8. Alternativas de Múltiplas Escolhas

- Forma geral em C:

```
switch (variavel) { —————→ apenas int ou char  
  case valor1:  
    <comandos>  
    break;  
  case valor2:  
    <comandos>  
    break;  
  case valorN:  
    <comandos>  
    break;  
  default:  
    <comandos>  
}
```

## 8. Alternativas de Múltiplas Escolhas

**Exemplo 1:** Escreva um algoritmo que solicita ao usuário dois operandos e um código de operação (1 - Soma, 2 - Subtração, 3 - Divisão ou 4 - Multiplicação) e realiza a operação correspondente sobre os operandos fornecidos pelo usuário.

## 8. Alternativas de Múltiplas Escolhas

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    float operando1, operando2, resultado;
    int operacao;

    printf("Entre com o primeiro operando:");
    scanf("%f", &operando1);
    printf("Entre com o segundo operando:");
    scanf("%f", &operando2);
    printf("*****\n");
    printf("Menu\n");
    printf("*****\n");
    printf("1 - Soma\n");
    printf("2 - Subtração\n");
    printf("3 - Divisão\n");
    printf("4 - Multiplicação\n");
    printf("*****\n");
    printf("Entre com a opcao:");
    scanf("%d", &operacao);
```

## 8. Alternativas de Múltiplas Escolhas

```
switch (operacao) {  
    case 1:  
        resultado = operando1 + operando2;  
        printf("A soma eh: %f \n", resultado);  
        break;  
    case 2:  
        resultado = operando1 - operando2;  
        printf("A subtração eh: %f \n", resultado);  
        break;  
    case 3:  
        resultado = operando1 / operando2;  
        printf("A divisão eh: %f \n", resultado);  
        break;  
    case 4:  
        resultado = operando1 * operando2;  
        printf("A multiplicação eh: %f \n", resultado);  
        break;  
    default:  
        printf("Operação inválida!");  
}  
return 0;  
}
```

“(...) a novidade normalmente emerge apenas para aquele que, sabendo com precisão o que deveria esperar, é capaz de reconhecer que algo saiu errado.”

*Thomas S. Kuhn, Físico Teórico,  
em seu livro A Estrutura das Revoluções Científicas.*