




Introdução a Programação
2023.1

Estruturas de Controle Condicional

Prof^a. Giorgia de Oliveira Mattos

giorgiamattos@gmail.com

giorgia@ci.ufpb.br



Estruturas de Controle de Execução do Programa

Estruturas de Execução do Programa

- São estruturas/instruções que controlam a **sequência** e a **frequência de execução** de partes de um programa
 - Estrutura sequencial: as instruções são executadas em sequência, sem desvios
 - Estrutura condicional: as instruções que serão executadas são selecionadas em função do resultado de condições
 - **IF, IF-ELSE, SWITCH-CASE**
 - Estruturas de repetição: repetem uma sequência de instruções enquanto uma condição for satisfeita (verdadeira)
 - **FOR, WHILE, DO-WHILE**

Fluxo de execução do programa

Fluxo de
execução
natural do
programa

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

```
int main (){
```

```
    int n1, n2, divisao;
```

```
    printf ("Digite o primeiro numero: ");
```

```
    scanf ("%d", &n1);
```

```
    printf ("Digite o segundo numero: ");
```

```
    scanf ("%d", &n2);
```

```
    divisao = n1/n2;
```

```
    printf ("%d / %d = %d", n1, n2, divisao);
```

```
    return 0;
```

```
}
```

Executa uma instrução por vez, uma única vez e na ordem em que aparecem.

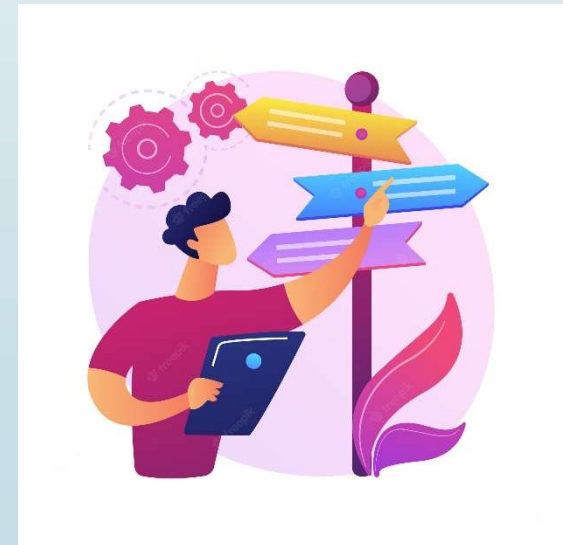
Isso pode ser um problema!

Dependendo do valor de n2, o programa executará operação inválida – divisão por zero – e será encerrado imediatamente.

O código fonte deve prever tais situações e indicar meios de contorná-las

Fluxo de execução do programa

- Um programa é mais do que uma lista de comandos
- Possíveis desvios e repetições podem fazer com que um programa tome rumos (caminhos) diferentes a cada execução
 - Mas a cada execução, em razão dos dados de entrada, o programa sempre percorrerá um caminho específico e definido dentre todos os que sejam possíveis





Um programa se comporta como um bairro com um conjunto de vias e sinalizações

O pedestre percorre caminhos diferentes de acordo com o seu destino:
Escola – sorveteria
Escola – banco – sorveteria

O programa escolhe o caminho a ser percorrido de acordo com os dados de entrada e deve estar pronto para funcionar em qualquer situação

O programa é um conjunto de instruções com ramificações internas para qualquer situação que possa ocorrer. São os valores de entrada passados para o programa que vão definir quais ramificações e qual conjunto de instruções serão executados naquela vez.

Fluxo de execução do programa

- Como o programa decide por qual caminho seguir?
 - Em cada desvio deve existir uma instrução especial que verifica alguma **condição**
 - De acordo com o valor da condição o programa seguirá um ou outro caminho
 - Então são os valores de entrada que vão definir o caminho seguido pelo programa, que a cada execução estará fazendo um caminho específico dentre todos os possíveis
 - Esses caminhos são definidos no código-fonte por estruturas especiais que podem criar desvios

Estrutura Condicional

- São utilizadas quando existe a necessidade de verificar condições para determinar quais instruções serão, ou não, executadas
- O comando **if** instrui o programa a tomar uma decisão baseada no valor da condição
- O comando **if** executará uma única instrução ou um conjunto de instruções se a condição for **verdadeira**

```
if (condição)  
    instrução 1;
```

```
if (condição)  
    instrução 1;  
else  
    instrução 2;
```


Estrutura Condicional - Simples

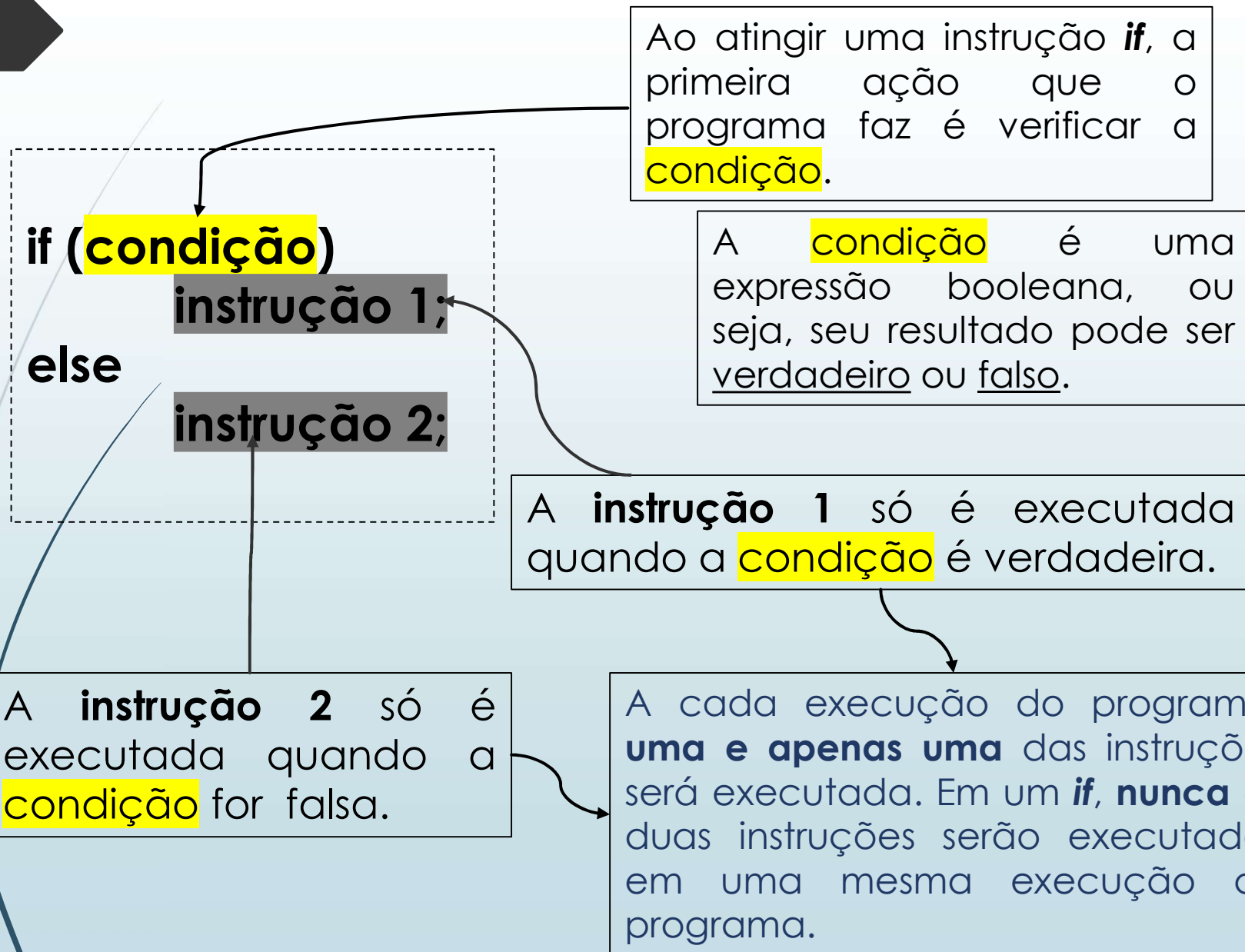
Ao atingir uma instrução **if**, a primeira ação que o programa faz é verificar a **condição**.

```
if (condição)  
    instrução 1;
```

A **condição** é uma expressão booleana, ou seja, seu resultado pode ser verdadeiro ou falso.

A **instrução 1** só é executada quando a **condição** é verdadeira.

Estrutura Condicional - Composta



Operadores e Expressões Relacionais

Operador relacional	Significado	Aplicação	Resultado
>	maior do que	$a > b$	1, se a é maior que b 0, caso contrário
>=	maior do que ou igual	$a \geq b$	1, se a é maior que ou igual b 0, caso contrário
<	menor do que	$a < b$	1, se a é menor que b 0, caso contrário
<=	menor do que ou igual	$a \leq b$	1, se a é maior que ou igual b 0, caso contrário
==	igual a	$a == b$	1, se a é igual b 0, caso contrário
!=	diferente de	$a != b$	1, se a é diferente de b 0, caso contrário

As expressões relacionais associam o uso dos operadores relacionais com valores numéricos. O resultado é valor booleano.

Exemplo 1 – condicional simples

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

```
int main () {  
    int n1, n2, divisao;
```

```
    printf ("Digite o primeiro numero: ");
```

```
    scanf ("%d", &n1);
```

```
    printf ("Digite o segundo numero: ");
```

```
    scanf ("%d", &n2);
```

```
    if (n2 != 0) {
```

```
        divisao = n1/n2;
```

```
        printf ("%d / %d = %d", n1, n2, divisao);
```

```
    }
```

```
    return 0;
```

```
}
```

Exemplo 1 – condicional composta

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

```
int main () {  
    int n1, n2, divisao;
```

```
    printf ("Digite o primeiro numero: ");
```

```
    scanf ("%d", &n1);
```

```
    printf ("Digite o segundo numero: ");
```

```
    scanf ("%d", &n2);
```

```
    if (n2 != 0) {
```

```
        divisao = n1/n2;
```

```
        printf ("%d / %d = %d", n1, n2, divisao);
```

```
    }
```

```
    else
```

```
        printf ("Erro: Divisao por zero!");
```

```
    return 0;
```

```
}
```

Exemplo 2 – condicional composta

```
#include <stdio.h>

int main () {
    int N1, N2 ;

    printf("Digite o primeiro numero: "); scanf("%d", &N1);
    printf("Digite o segundo numero: "); scanf("%d", &N2);
    if (N1 == N2)
        printf("Os numeros sao iguais!");
    else
        printf("O numeros sao diferentes!");
    return 0;
}
```

Estrutura Condicional

► Estrutura condicional encadeada

- É formada pela combinação de estruturas condicionais simples ou compostas, uma dentro da outra, não havendo limite para o número de combinações, podendo, em alguns casos, gerar um código bastante complexo

```
if (condição A)
    instrução 1;
else
    if (condição B)
        instrução 2;
    else
        instrução 3;
```

Observações

- O **else** sempre está associado ao **if** mais próximo, desde que ele não esteja utilizando as chaves
- Na dúvida, use as chaves!

Estrutura Condicional

➡ Exemplo

```
#include <stdio.h>

int main () {
    int N1, N2 ;
    printf("Digite o primeiro numero: "); scanf("%d", &N1);
    printf("Digite o segundo numero: "); scanf("%d", &N2);
    if (N1 == N2)
        printf("Os numeros sao iguais!");
    else
        if (N1 > N2)
            printf("Maior valor = %d", N1);
        else
            printf("Maior valor = %d", N2);
    return (0); }
```

Exercícios

- O que será impresso na tela quando x tiver os valores 10, -20 e 0?

```
scanf("%d", &x);  
if (x > 0){  
    printf("%d eh positivo\n", x);  
}else{  
    printf("%d nao eh positivo\n", x);  
}  
if (x < 0){  
    printf("%d eh negativo\n", x);  
}
```

```
10 eh positivo  
  
-20 nao eh positivo  
-20 eh negativo  
  
0 nao eh positivo
```

Exercícios

- O que será impresso na tela quando x tiver os valores 10, -20 e 0?

```
scanf("%d", &x);  
if (x > 0){  
    printf("%d eh positivo\n", x);  
}  
if (x < 0){  
    printf("%d eh negativo\n", x);  
}else{  
    printf("%d nao eh positivo\n", x);  
}
```

```
10 eh positivo  
10 nao eh positivo  
  
-20 eh negativo  
  
0 não eh positivo
```

Exercícios

- O que será impresso na tela quando x tiver os valores 10, -20 e 0?

```
scanf("%d", &x);  
if (x > 0){  
    printf("%d eh positivo\n", x);  
    if (x < 0 ){  
        printf("%d eh negativo\n", x);  
    }  
}else{  
    printf("%d nao eh positivo\n", x);  
}
```

```
10 eh positivo  
-20 nao eh positivo  
0 não eh positivo
```

Exercícios

- O que será impresso na tela quando x tiver os valores 10, -20 e 0?

```
scanf("%d", &x);  
if (x > 0){  
    printf("%d eh positivo\n", x);  
    if (x < 0 ){  
        printf("%d eh negativo\n", x);  
    }  
}else{  
    printf("%d nao eh positivo\n", x);  
}
```

*Para quais valores de x
essa instrução será
executada?*

A instrução nunca será alcançada, pois a condição do **if** interno nunca será verdadeira por este estar inserido na parte verdadeira do **if** externo, cuja condição é a oposta.

Exercícios

- O que será impresso na tela quando x tiver os valores 10, -20 e 0?

```
scanf("%d", &x);  
if (x > 0){  
    printf("%d eh positivo\n", x);  
}else{  
    if (x < 0){  
        printf("%d eh negativo\n", x);  
    }else{  
        printf("%d nao eh positivo\n", x);  
    }  
}
```

10 eh positivo
-20 eh negativo
0 nao eh positivo

Estrutura Condicional

- Diversos testes relacionais podem ser combinados em uma única condição de teste através do uso dos **operadores lógicos**
- Operadores lógicos:
 - && (E lógico – conjunção)
 - || (OU lógico – disjunção)
 - ! Negação lógica

Estrutura Condicional

```
if (condição1 && condição2)
```

```
    instrução;
```

A instrução somente será executada se as condições 1 e 2 forem simultaneamente verdadeiras.

```
if (condição1 || condição2)
```

```
    instrução;
```

A instrução será executada se pelo menos uma das condições for verdadeira.

Estrutura Condicional

```
if (!condição)  
    instrução;
```

A instrução somente será executada se a condição for falsa

- ✓ O operador de negação inverte o valor lógico da condição, ou seja, se a condição for falsa então a sua “negação” é o valor verdadeiro. Assim, a condição torna-se verdadeira e a instrução será executada.
- ✓ Em C, o valor 0 é considerado falso e diferente de 0 é considerado verdadeiro

Estrutura Condicional

```
#include <stdio.h>

int main () {
    if (2 > 3 || 3 < 7)
        printf (" Resultado da Condicional ||: VERDADEIRO");
    else
        printf (" Resultado da Condicional ||: FALSO");

    if (3 < 2 && 2 == 2)
        printf (" Resultado da Condicional &&: VERDADEIRO");
    else
        printf (" Resultado da Condicional &&: FALSO");

    return 0;
}
```

Estrutura Condicional

- Condicional de múltipla escolha: a estrutura funciona como um conjunto de opções. Pode-se representá-la usando a condicional encadeada ou a instrução **switch-case**
 - **SWITCH** é um comando em C que serve para fazer testes condicionais, testando igualdades
 - O conteúdo de uma variável é comparado com um valor constante, e caso a comparação seja verdadeira, um conjunto de instruções será executado.
- **Diferenças entre IF-ELSE e SWITCH**
 - Os **cases** não aceitam operadores lógicos. Isso limita o **case** a apenas valores definidos.
 - O **switch** executa seu bloco em cascata. Ou seja, se a variável indicar para o primeiro **case** e dentro do switch tiver 5 cases, o switch executará todos os outros 4 cases a não ser que utilizemos um comando para sair do switch, o break

Estrutura Condicional

```
switch (expressão){  
    case expressão-constante1: instrução1;  
    case expressão-constante2: instrução2;  
    ...  
    case expressão-constanteN: instruçãoN;  
    default: instrução;  
}
```

****** na instrução **switch**, todas as instruções seguindo a ramificação **case** selecionada são executadas, mesmo que algumas destas instruções façam parte de outra ramificação **case**. Este comportamento é evitado por meio do desvio incondicional **break**

Estrutura Condicional

```
#include <stdio.h>

int main () {
    int valor;
    printf ("Digite um numero de 1 a 7: "); scanf("%d", &valor);
    switch ( valor ) {
        case 1 : printf ("Domingo\n"); break;
        case 2 : printf ("Segunda\n"); break;
        case 3 : printf ("Terça\n"); break;
        case 4 : printf ("Quarta\n"); break;
        case 5 : printf ("Quinta\n"); break;
        case 6 : printf ("Sexta\n"); break;
        case 7 : printf ("Sábado\n"); break;
        default : printf ("Valor invalido!\n"); }
    return 0; }
```

Linguagem C - EXERCÍCIOS

1. Faça um programa para ler um número inteiro (considere que serão lidos apenas valores positivos e inteiros) e escrever se o número é par ou ímpar.
2. Escreva um programa que leia três valores e mostre qual é o maior valor e qual é o menor valor.
3. Faça um programa para ler 3 valores e escrever a soma dos 2 maiores. Considere que o usuário não informará valores iguais.
4. Escreva um programa que leia 3 números inteiros e mostre-os em ordem crescente.
5. Faça um programa que leia 2 valores inteiros e uma das seguintes operações a serem executadas (codificada da seguinte forma: 1.Adição, 2.Subtração, 3.Divisão, 4.Multiplicação). Calcular e escrever o resultado dessa operação sobre os dois valores lidos.

Linguagem C - EXERCÍCIOS

6. Faça um programa que leia a idade de 2 homens e 2 mulheres (considere que a idade dos homens será sempre diferente, assim como das mulheres). Calcule e escreva a soma das idades do homem mais velho com a mulher mais nova, e o produto das idades do homem mais novo com a mulher mais velha.
7. Faça um programa que mostre o menu de opções a seguir, receba a opção do usuário e os dados necessários para cada operação.

Menu de opções:

1 - Somar dois números;

2 - Raiz quadrada de um número.

Digite a opção desejada:

Linguagem C - EXERCÍCIOS

8. Dados três valores X, Y e Z, verificar se eles podem ser os comprimentos dos lados de um triângulo e, se for, verificar se é um triângulo equilátero, isósceles ou escaleno. Se eles não formarem um triângulo, escrever a mensagem “Os valores não formam um triângulo”. Propriedades e definições:

- Propriedade – o comprimento de cada lado de um triângulo é menor do que a soma dos comprimentos dos outros dois lados.
- Definição 1 – chama-se triângulo equilátero o triângulo que tem os comprimentos dos três lados iguais.
- Definição 2 – chama-se triângulo isósceles o triângulo que tem os comprimentos de dois lados iguais. Portanto todo triângulo equilátero é também isósceles.
- Definição 3 – chama-se triângulo escaleno o triângulo que tem os comprimentos de seus três lados diferentes.

Linguagem C - EXERCÍCIOS

9. Faça um programa que receba como entrada os coeficientes (a, b, c) de uma equação do 2º grau, calcule e mostre as raízes desta equação. As raízes de uma equação do 2º grau são calculadas pela fórmula
$$x = \frac{-b \pm \sqrt{D}}{2a}$$

onde $D = b^2 - 4ac$ é o discriminante da equação.

Para esse discriminante há três possíveis situações:

- a) Se $D < 0$ – não há solução real, pois não existe raiz quadrada de número negativo;
- b) Se $D = 0$ – há duas soluções iguais: $x_1 = x_2 = -b/2a$
- c) Se $D > 0$ – há duas soluções reais e diferentes: x_1 e x_2 .

Linguagem C - EXERCÍCIOS

10. Escreva um programa que calcule o Valor a Pagar pela conta de energia elétrica para uma determinada Classe Consumidora. Veja as informações a seguir:

- a) O programa deve receber como dados de entrada: o tipo da Classe Consumidora, (conforme definido na Tabela 1) e o Consumo em quilowatts/hora (KWh);
- b) O cálculo do Valor do Fornecimento (VF), em Reais (R\$), é definido pela seguinte fórmula: $VF = \text{Consumo} \times \text{Tarifa}$, onde a tarifa é definida na Tabela 1, conforme a classe consumidora;
- c) O Valor a Pagar (VP) é definido pela seguinte fórmula: $VP = VF + \text{ICMS}$, onde a taxa de ICMS é calculada aplicando uma alíquota de 30% ao valor do fornecimento ($\text{ICMS} = 0.3 \times VF$).

<u>Classe Consumidora</u>	<u>Tarifa (R\$)</u>
A	0.5
B	0.8
C	1.0

Linguagem C - EXERCÍCIOS

11. Escreva um programa que receba o código correspondente ao cargo de um funcionário e seu salário atual. Calcule e mostre o cargo, o valor do aumento e seu novo salário. Os cargos estão na tabela a seguir.

Código	Cargo	Percentual
1	Escriturário	50%
2	Secretário	35%
3	Caixa	20%
4	Gerente	10%
5	Diretor	Não tem aumento