



Linguagem C

Estruturas de Controle Condicionais

MsC. Douglas Santiago Kridi

Programação I - 2018.2

Bacharelado em Ciência da Computação

Universidade Estadual do Piauí

douglaskridi@gmail.com

Introdução

- Estrutura de controle (ou *fluxo de controle*) refere-se à ordem em que instruções, são executadas ou avaliadas em programas.
 - Um condicional é uma estrutura de controle de fluxo, executando um ou vários comandos se a *condição testada* for *verdadeira* e, em alguns casos, executando um ou vários comandos se a condição testada for *falsa*.

Introdução

- Condição: É uma *expressão lógica* testada pela unidade lógica aritmética, devolvendo como resposta: verdadeiro ou falso.
- Em C uma condição é nada mais que qualquer expressão, cujo valor resultante precisa ser necessariamente um número inteiro.
 - Se esse *valor for 0 (zero)*, então a expressão é interpretada como sendo *falsa*. Caso contrário, se a expressão resultar em qualquer outro *valor não nulo*, então a condição será tratada como *verdadeira*.

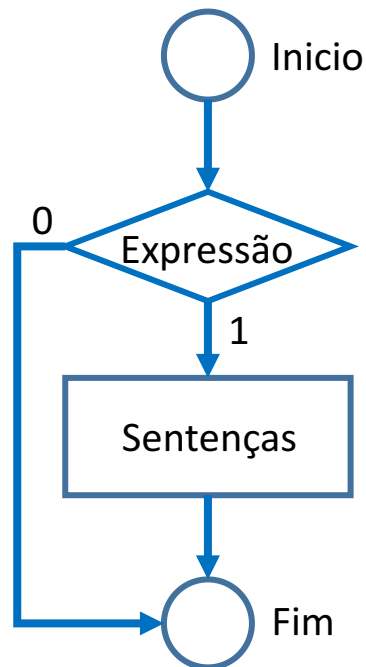
Introdução

- Convém recordarmos os operadores relacionais e lógicos:

Operadores Relacionais	Em C	Operadores Lógicos	Em C
Igual	==	Conjunção (E)	&&
Diferente	!=	Disjunção (OU)	
Maior que	>	Negação (Não)	!
Menor que	<		
Maior ou igual a	>=		
Menor ou igual a	<=		

Estrutura: if

- A estrutura if executa uma sentença ou bloco de sentenças somente se uma determinada condição for verdadeira.



```
if (expressão) {  
    sentença;  
    sentença;  
    ...  
}
```

Estrutura: if

- Exemplo: O código recebe a idade do usuário, e a partir dessa informação, ele imprime se a pessoa pode obter sua carteira de habilitação:

```
int idade;  
  
printf("Digite sua idade: ");  
scanf("%d", &idade);  
  
if (idade >= 18) {  
    printf("Ja pode obter habilitação!");  
}
```

- Se o resultado da expressão ($idade \geq 18$) for verdadeiro (diferente de zero), então o próximo bloco é executado.
- Caso contrário, a execução salta este bloco e passa para o comando após o if.

- Se o bloco de um if contém apenas uma sentença, pode-se omitir as chaves que delimitam o bloco.
 - Pode dificultar a leitura.

Estrutura: if

- Exemplo: Testando uma entrada qualquer:

```
int num;
printf ("Digite um numero: ");
scanf ("%d",&num);

if (num > 10)
    printf ("\n\n O numero é maior que 10");
if (num == 10) {
    printf ("\n\n O numero é igual a 10."); }
if (num < 10)
    printf ("\n\n O numero é menor que 10");
```

Estrutura: if

- Retorno da condição: Observe as duas situações abaixo:

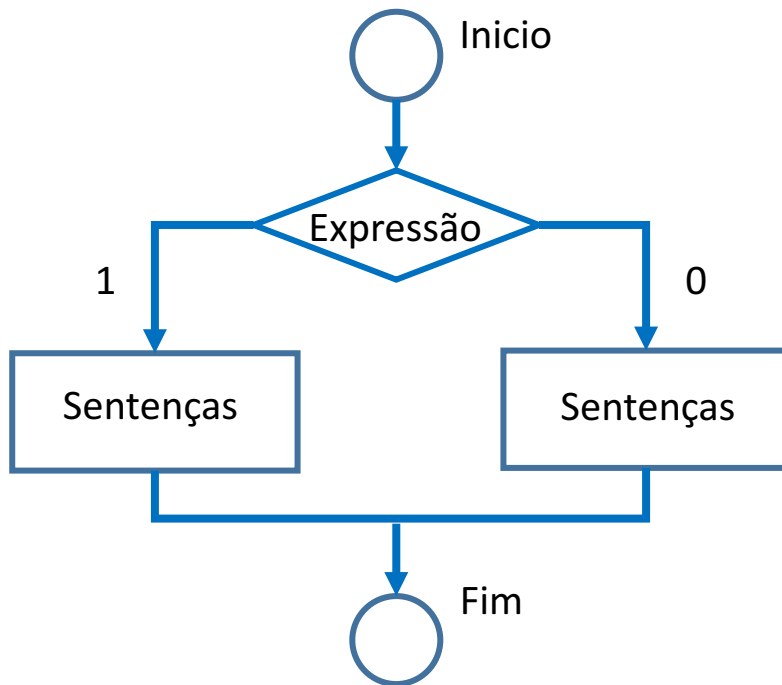
```
if (variavel != 0) {  
    ...  
}
```

```
if (variavel) {  
    ...  
}
```

- O que cada uma faz? Qual o retorno da expressão?
 - Ambas verificam se o valor de uma variável inteira é diferente de zero. São equivalentes!
 - Um valor não nulo sempre será interpretado como verdadeiro.
 - Se a variável apresentar qualquer valor diferente de zero, o if concluirá que esse valor é não nulo e, por consequência, o interpretará como verdadeiro.

Estrutura: if ... else

- A estrutura *if ... else* executa um bloco de sentenças tanto para uma condição verdadeira, quanto para uma condição falsa.



```
if (expressão) {  
    sentença;  
    sentença;  
    ...  
} else {  
    sentença;  
    sentença;  
    ...  
}
```

Estrutura: if ... else

- Se a condição da expressão do primeiro if for avaliada como verdadeira, então, obrigatoriamente, a expressão do segundo if é necessariamente falsa e vice-versa.
 - Ou seja, as duas condições são *mutuamente exclusivas* e apenas um dos blocos das estruturas condicionais do if é executado.

Estrutura: if ... else

- Exemplo: retomando o exemplo da carteira de habilitação, também desejamos expor uma mensagem negando a autorização:

```
int idade;

printf("Digite sua idade: ");
scanf("%d", &idade);

if (idade >= 18) {
    printf("Você ja pode obter habilitacao!");
} else {
    printf("Espere mais alguns anos!");
}
```

- Se a expressão for *verdadeira*, então apenas o *primeiro* bloco de sentenças é executado.
- se a expressão revelar-se *falsa*, então apenas o *segundo* bloco de sentenças é executado.

Estrutura: if ... else

- Exemplo: Testando uma entrada qualquer usando if .. Else:

```
int num;
printf ("Digite um numero: ");
scanf ("%d", &num);

if (num == 10) {
    printf ("\n\n Voce acertou!\n");
    printf ("O numero é igual a 10.\n");
} else {
    printf ("\n\n Voce errou!\n");
    printf ("O numero é diferente de 10.\n");
}
```

Estrutura: if ... else

- Pratique: Escrever um algoritmo que leia dois valores inteiros distintos e informe qual é o maior.

Operadores lógicos

- Podemos usar operadores lógicos, que unem duas expressões em uma condição maior e mais complexa.
- Lembre-se que C interpreta o valor 0 como falso e os demais valores como verdadeiro. Logo:

Ideia	Expressão	Resultado
E (and)	(cond1) && (cond2)	Resulta 1 se as duas condições forem verdadeiras. Resulta 0, se uma das duas for falsa
OU (or)	(cond1) (cond2)	Resulta 1, se uma das duas condições for verdadeira. Resulta 0, se as duas forem falsas.
NÃO (not)	! (condição)	Resulta 1, se a condição for falsa. Resulta 0, se a condição for verdadeira.

Operadores lógicos

■ Comportamento dos operandos:

&& (E)		Cond1	
		falso	verdade
Cond2	falso	falso	falso
	verdade	falso	verdade

(OU)		Cond1	
		falso	verdade
Cond2	falso	falso	verdade
	verdade	verdade	verdade

! (NÃO)	Resultado	
Condição	falso	verdade
(! Condição)	verdade	falso

Operadores lógicos

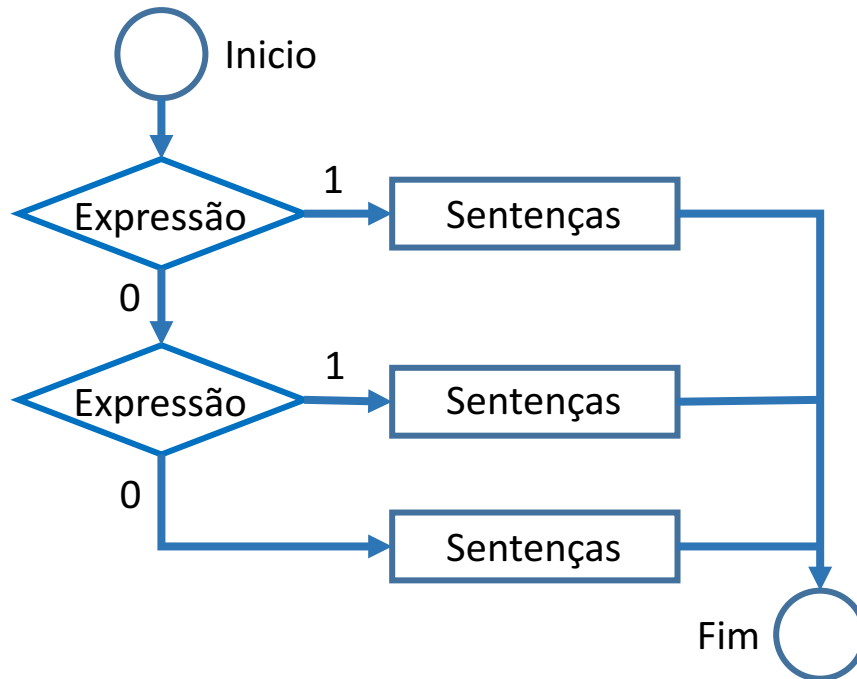
- Exemplo: Vamos validar a carteira de habilitação a partir da idade e do seu desempenho no exame de direção:

```
int idade;  
float media;  
  
printf("Digite sua idade: ");  
scanf("%d", &idade);  
printf("Digite sua media nos exames: ");  
scanf("%f", &media);  
  
if ( (idade >= 18) && (media >= 5.0) ) {  
    printf("Voce está aprovado!");  
} else {  
    printf("Ainda não aprovado!");  
}
```

- Cada expressão entre parênteses.
- Ambas precisam ser verdadeiras em função de &&

Estrutura: if aninhado

- A Estrutura if...else *aninhada*, permite realizar decisões múltiplas.
- A estrutura oferece vários blocos como alternativas para serem executados, cada qual associado à sua própria expressão condicional.



```
if (expressão) {  
    sentenças;  
    ...  
} else if (expressão) {  
    sentenças;  
    ...  
} else {  
    sentenças;  
    ...  
}
```

Estrutura: if aninhado

- Exemplo: Código que informa o período de renovação dos exames da carteira de habilitação.
 - Menores de idade não possuem carteira. Até 65 anos, os prazos são de 5 em 5 anos; depois, o exame precisa de renovação a cada 3 anos.

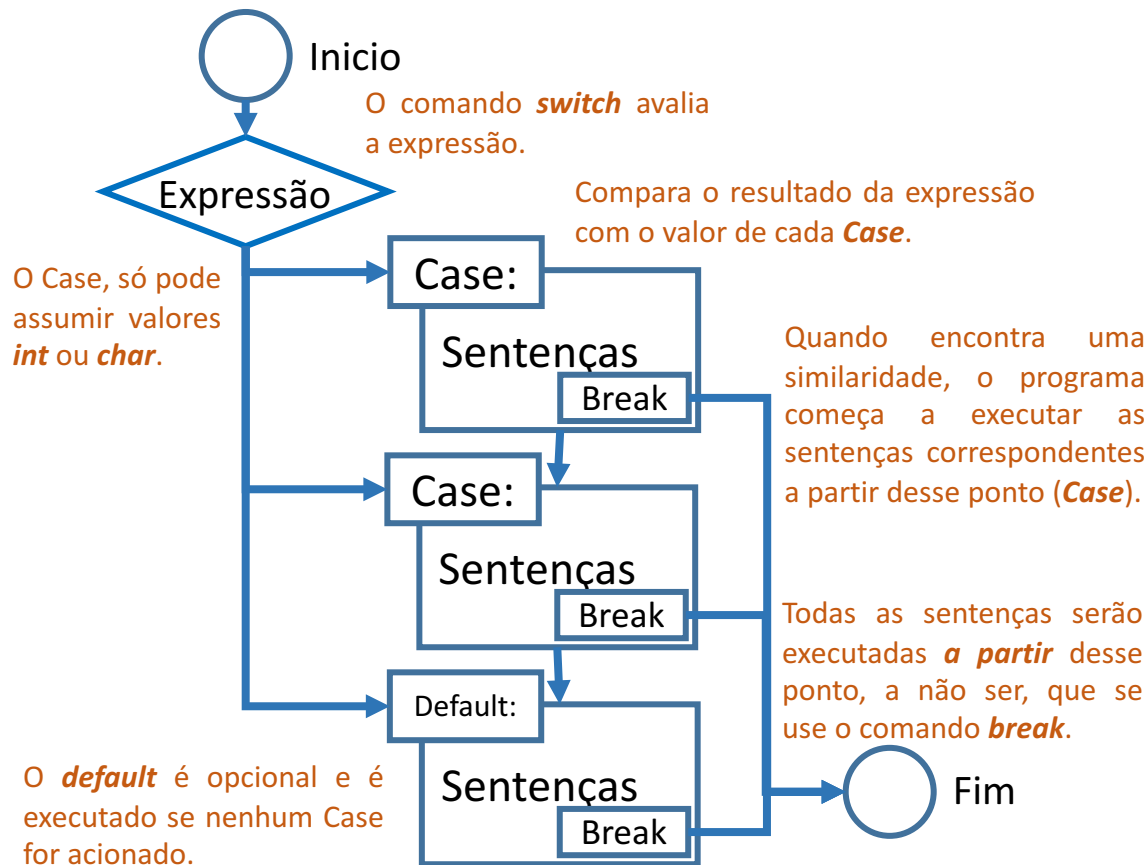
```
int idade;  
float media;  
  
printf("Digite sua idade: ");  
scanf("%d", &idade);  
  
if ( (idade >= 0) && (idade < 18) ) {  
    printf("Nao possui carteira de habilitacao.\n");  
} else if ( (idade >= 18) && (idade < 65) ) {  
    printf("Renove exames a cada 5 anos.\n");  
} else if (idade >= 65) {  
    printf("Renove exames a cada 3 anos.\n");  
}
```

Estrutura: if aninhado

- **Pratique:** Escrever um algoritmo que leia o nome e as três notas obtidas por um aluno durante o semestre.
 - Calcular a sua média (aritmética), informar o nome e sua menção aprovado (media ≥ 7), Reprovado (media ≤ 5) e Recuperação (media entre 5.1 a 6.9).

Estrutura: switch ... case

- O switch é uma construção de múltiplas possibilidades de decisão. Ele compara o resultado de uma expressão com uma série de valores constantes.



```

switch (expressão) {
    case valor1:
        sentenças;
        ...
    case valor2:
        sentenças;
        ...
    case valor3:
        sentenças;
        ...
    default:
        sentenças;
        ...
}
  
```

Estrutura: switch ... case

- Programa que recebe um numero e indica o dia da semana:

```
int valor;  
printf ("Digite um valor 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 ("Sabado\n");  
        break;  
    default :  
        printf ("Valor invalido!\n");  
}
```

Estrutura: switch ... case

- Programa que recebe um numero e emite uma mensagem:

```
int num;
printf ("Digite um numero: ");
scanf ("%d",&num);
switch (num)    {
    case 9:
        printf ("\n\nO numero e igual a 9.\n");
        break;
    case 10:
        printf ("\n\nO numero e igual a 10.\n");
        break;
    case 11:
        printf ("\n\nO numero e igual a 11.\n");
        break;
    default:
        printf ("\n\nO numero nao e nem 9 nem 10 nem 11.\n");
}
```

Estrutura: switch ... case

- Pratique: Um funcionário irá receber um aumento de acordo com o seu plano de trabalho, de acordo com a tabela abaixo:

PLANO	AUMENTO
A	10%
B	15%
C	20%

- Faça um programa que leia o plano de trabalho e o salário atual de um funcionário e calcula e imprime o seu novo salário. Use o comando switch.

Referências Bibliográficas



C: COMO PROGRAMAR

DEITEL, Harvey M.; DEITEL, Paul J. Editora Pearson - 6ª ed. 2011

Fundamentos da Programação de Computadores

Ascencio, Ana F. G., Campos, Edilene A. V. de, - Editora Pearson

2012



Lógica de Programação e Estrutura de Dados

Puga, Sandra. Riseti, Gerson. – Ed. Pearson - 2016

Lógica de Programação Algorítmica (Apostila)

Guedes, Sergio. - Editora Pearson/Ser - 2014

