

Fundamentos de Programação

Linguagem de Programação C
Estrutura de Seleção

Sumário

1 Estrutura

2 Operadores Relacionais

3 Seleção Simples - if

4 Seleção Composta - if-else

5 Operadores Lógicos

6 Estruturas Aninhadas

Estrutura

- Até agora, seu código é executado de forma **sequencial**, com todas as linhas sendo processadas em **ordem**.
- Agora utilizaremos uma nova estrutura que executa que permite executar uma linha apenas quando uma **condição** específica for atendida.
- Essa estrutura pode ser chamada de **estrutura de seleção, estrutura de decisão ou estrutura condicional**.

Exemplo - Fazer um pagamento através de Pix com validação:

- Ler a chave Pix.
- Ler o valor que deve ser enviado.
- Se o saldo for maior ou igual ao valor do pix:
 - Subtrair o valor informado do saldo do pagador.
 - Adicionar o valor informado ao saldo do recebedor.
 - Exibir uma mensagem de confirmação.
- Senão:
 - Exibir uma mensagem de negação.

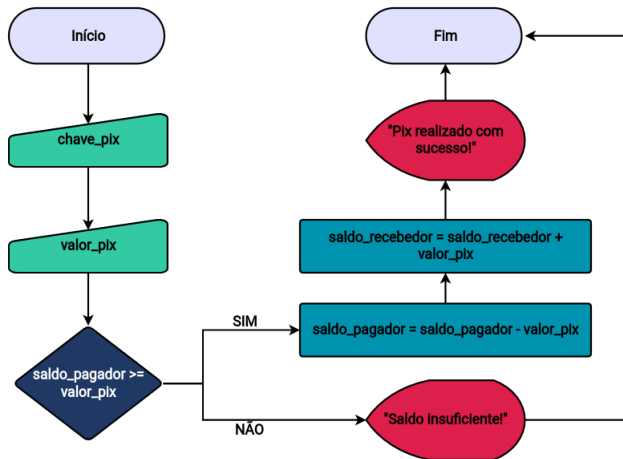


Figura 1: Exemplo de Pix em fluxograma com decisão.

```
1 INICIO
2   leia(chave_pix)
3   leia(valor_pix)
4   SE (saldo_pagador >= valor_pix) ENTAO
5       saldo_pagador = saldo_pagador - valor_pix
6       saldo_recebedor = saldo_recebedor + valor_pix
7       escreva("Pix realizado com sucesso!")
8   SENAO
9       escreva("Saldo insuficiente!")
10  FIMSE
11 FIM
```

Código 1: Exemplo Pix com decisão em pseudocódigo.

```
1 int main() {
2     scanf("%s", chave_pix);
3     scanf("%f", &valor_pix);
4     if (saldo_pagador >= valor_pix) {
5         saldo_pagador = saldo_pagador - valor_pix
6         saldo_recebedor = saldo_recebedor + valor_pix
7         printf("Pix realizado com sucesso!\n");
8     } else {
9         printf("Saldo insuficiente!\n");
10    }
11    return 0;
12 }
```

Código 2: Exemplo Pix com decisão em linguagem C.


```
1 if (condicao){  
2     printf("A condicao eh verdadeira!\n");  
3 }
```

Código 3: Estrutura básica seleção simples if.

- Ao utilizarmos o comando `if`, é necessário **definir uma condição**.
 - » Se essa condição for verdadeira, o código contido dentro do bloco `{}` será executado;
 - » Caso contrário, o bloco de código associado ao `if` será ignorado.
- As condições devem ser expressas como **comparações relacionais**, utilizando operadores semelhantes aos usados na matemática.
- Toda validação de uma condição retorna um valor que pode ser **VERDADEIRO** ou **FALSO**.

Operadores Relacionais

Operador	Significado
==	Igual
>	Maior
<	Menor
>=	Maior ou igual
<=	Menor ou igual
!=	Diferente

Tabela 1: Operadores relacionais.

Atenção!

- Trocar acidentalmente os operadores de igualdade (==) e de atribuição (=) é prejudicial, pois essa troca geralmente **não causa erros de sintaxe**. Em vez disso, as instruções com esses erros costumam ser compiladas corretamente, e os programas são executados até o fim, gerando, provavelmente, **resultados incorretos** devido a erros de lógica em tempo de execução.

Seleção Simples - if

```
1 #include <stdio.h>
2 int main(){
3     int idade_pessoa;
4     printf("Qual sua idade?\n");
5     scanf("%d", &idade_pessoa);
6
7     if (idade_pessoa >= 18){
8         printf("Voce pode dirigir!\n");
9     }
10    return 0;
11 }
```

Código 4: Exemplo maior de idade.

- Ao utilizarmos o comando `if-else`¹, é necessário **definir uma condição**.
 - Se a condição for verdadeira, o código contido no bloco `{}` associado ao `if` será executado;
 - Caso contrário, o código contido no bloco `{}` associado ao `else` será executado.
- Em resumo, todas as situações que não atendem à condição do `if` farão com que o bloco `else` seja executado.


```
1 if (condicao)(  
2     printf("A condicao eh verdadeira!\n");  
3 )else{  
4     printf("A condicao eh falsa\n");  
5 }
```

Código 5: Estrutura básica de seleção composta com if-else.

¹O operador condicional ?: pode ser usado como uma alternativa à estrutura if-else. Trata-se de um operador ternário, formado por três operandos: uma condição (primeiro operando), o valor retornado se a condição for verdadeira (segundo operando) e o valor retornado se a condição for falsa (terceiro operando), como no exemplo `printf("%s\n", x == y ? "V": "F");`.

Seleção Composta - if-else

```
1 #include <stdio.h>
2 int main(){
3     int idade_pessoa;
4     printf("Qual sua idade?\n");
5     scanf("%d", &idade_pessoa);
6
7     if (idade_pessoa >= 18){
8         printf("Voce pode dirigir!\n");
9     }else{
10         printf("Voce nao pode dirigir!\n");
11     }
12     return 0;
13 }
```

Código 6: Exemplo maior de idade.

```
1 #include <stdio.h>
2 int main(){
3     int n1, n2;
4     printf("Entre com o primeiro valor:\n");
5     scanf("%d", &n1);
6     printf("Entre com o segundo valor:\n");
7     scanf("%d", &n2);
8
9     if (n1 >= n2){
10         printf("O valor maior eh: %d\n", n1);
11     }else{
12         printf("O valor maior eh: %d\n", n2);
13     }
14     return 0;
15 }
```

Código 7: Exemplo número maior.

```
1 #include <stdio.h>
2 int main() {
3     int senha;
4
5     printf("Digite a senha de 3 dígitos:\n");
6     scanf("%d", &senha);
7
8     if (senha == 123) {
9         printf("Acesso permitido. Bem-vindo!\n");
10    } else {
11        printf("Senha incorreta!\n");
12    }
13    return 0;
14 }
```

Código 8: Exemplo senha.

```
1 #include <stdio.h>
2 int main() {
3     int senha;
4
5     printf("Digite a senha de 3 dígitos:\n");
6     scanf("%d", &senha);
7
8     if (senha != 123) {
9         printf("Senha incorreta!\n");
10    } else {
11        printf("Acesso permitido. Bem-vindo!\n");
12    }
13    return 0;
14 }
```

Código 9: Exemplo senha invertido.

Operadores Lógicos

- Em alguns casos, pode ser necessário analisar mais de uma condição. Para isso, utilizamos os operadores lógicos.
- Toda operação com operador lógico retorna um valor que pode ser **VERDADEIRO** ou **FALSO**.

Operador	Significado
&&	E lógico (AND)
	OU lógico (OR)
!	NÃO lógico (NOT)

Tabela 2: Tabela de Operadores Lógicos.


```
1 #include <stdio.h>
2 int main(){
3     float media_aluno, frequencia_aluno;
4     printf("Qual sua media?\n");
5     scanf("%f", &media_aluno);
6     printf("Qual sua frequencia?\n");
7     scanf("%f", &frequencia_aluno);
8
9     if(media_aluno >= 7.0 && frequencia_aluno >= 75.0){
10         printf("Aprovado!\n");
11     }else{
12         printf("Reprovado!");
13     }
14     return 0;
15 }
```

Código 10: Exemplo operador lógico &&.

```
1 #include <stdio.h>
2 int main(){
3     int idade_pessoa;
4     printf("Qual sua idade?\n");
5     scanf("%d", &idade_pessoa);
6
7     if (idade_pessoa >= 16 && idade_pessoa < 18){
8         printf("Voto facultativo!\n");
9     }else{
10         printf("Voto obrigatório!\n");
11     }
12     return 0;
13 }
```

Código 11: Exemplo operador lógico &&.

```
1 #include <stdio.h>
2 int main(){
3     float valor_total;
4     int quantidade_itens;
5
6     printf("Qual o valor total da compra?\n");
7     scanf("%f", &valor_total);
8     printf("Quantos itens na sua compra?\n");
9     scanf("%f", &quantidade_itens);
10
11     if(valor_total >= 299.00 || quantidade_itens >= 10){
12         printf("Frete grátis!\n");
13     }else{
14         printf("Não foi possível obter frete grátis!\n");
15     }
16     return 0;
17 }
```

Código 12: Exemplo operador lógico ||.

```
1  #include <stdio.h>
2  int main(){
3      char dia_desconto;
4      char estudante;
5
6      printf("Voce eh estudante?\n");
7      printf("Digite s, sem sim. \n Digite n, se se nao.\n");
8      scanf(" %c", &estudante);
9
10     printf("Hoje eh dia de desconto?\n");
11     printf("Digite s, sem sim. \n Digite n, se se nao.\n");
12     scanf(" %c", &dia_desconto);
13
14     if(estudante == 's' || dia_desconto == 's'){
15         printf("Voce tem direito a meia entrada!\n");
16     }else{
17         printf("Voce nao tem direito a meia entrada!\n");
18     }
19
20     return 0;
21 }
```

Código 13: Exemplo operador lógico ||.

- O operador lógico NOT (!) **inverte** o valor lógico de uma condição:
 - » Se a condição for verdadeira, o NOT a torna falsa;
 - » Se for falsa, o NOT a torna verdadeira.
- Para que o bloco `if` seja executado, a condição deve ser verdadeira. No entanto, se o operador lógico NOT for utilizado, o valor lógico da condição será invertido:
 - » Uma condição originalmente verdadeira será considerada falsa, impedindo a entrada no bloco `if`;
 - » Uma condição originalmente falsa será considerada verdadeira, permitindo a execução do bloco `if`.
- Em resumo, o operador NOT altera o valor lógico da condição, invertendo o comportamento esperado para a execução do bloco `if`.

```
1 if (!condicao)(  
2     printf("A condicao eh falsa!\n");  
3 )else{  
4     printf("A condicao eh verdadeira\n");  
5 }
```

Código 14: Exemplo operador lógico !.

```
1 #include <stdio.h>
2 int main() {
3     float media;
4
5     printf("Qual sua média?\n");
6     scanf("%f", &media);
7
8     if (media >= 7.0) {
9         printf("Aprovado!\n");
10    } else {
11        printf("Reprovado!\n");
12    }
13    return 0;
14 }
```

Código 15: Exemplo nota.

```
1 #include <stdio.h>
2 int main() {
3     float media;
4
5     printf("Qual sua média?\n");
6     scanf("%f", &media);
7
8     if (!media >= 7.0) {
9         printf("Reprovado!\n");
10    } else {
11        printf("Aprovado!\n");
12    }
13    return 0;
14 }
```

Código 16: Exemplo nota com operador lógico !.

A	B	A && B (AND)
V	V	V
V	F	F
F	V	F
F	F	F

Tabela 3: Tabela Verdade para o Operador Lógico AND.

A	B	A B (OR)
V	V	V
V	F	V
F	V	V
F	F	F

Tabela 4: Tabela Verdade para o Operador Lógico OR.

A	!A (NOT)
V	F
F	V

Tabela 5: Tabela Verdade para o Operador Lógico NOT.

Em C, verdadeiro é qualquer valor diferente de zero. Falso é zero. As expressões que usam operadores relacionais ou lógico devolvem zero para falso e 1 para verdadeiro.

(Schildt, 1996)

Operador	Retorno
Operador Aritmético	Número
Operador Relacional	Verdadeiro ou Falso
Operador Lógico	Verdadeiro ou Falso

Tabela 6: Retornos para Operadores.

Estruturas Aninhadas

```
1 #include <stdio.h>
2 int main(){
3     float media;
4     printf("Qual sua media?\n");
5     scanf("%f", &media);
6
7     if(media >= 5.0){
8         if (media >= 5.0 && media < 7.0){
9             printf("Reprovado!\n");
10        }else{
11            printf("Aprovado!\n");
12        }
13    }else{
14        printf("Reprovado!\n");
15    }
16    return 0;
17 }
```

Código 17: Exemplo nota com if encadeado.

```
1  if (condicao) {  
2      //instruções.  
3  } else if (outra_condicao) {  
4      //instruções.  
5  } else if (n_condicao) {  
6      //instruções.  
7  }
```

Código 18: Estrutura básica else if.

```
1 if (condicao) {  
2     //instruções.  
3 } else if (outra_condicao) {  
4     //instruções.  
5 } else if (n_condicao) {  
6     //instruções.  
7 } else {  
8     //instruções caso nenhuma condição seja atendida.  
9 }
```

Código 19: Estrutura básica else if.

```
1 #include <stdio.h>
2 int main() {
3     float media;
4     printf("Qual sua média?\n");
5     scanf("%f", &media);
6
7     if (media >= 7.0) {
8         printf("Aprovado!\n");
9     } else if (media >= 5.0) {
10        printf("Recuperação!\n");
11    } else {
12        printf("Reprovado!\n");
13    }
14    return 0;
15 }
```

Código 20: Exemplo nota com else if.

```
1 #include <stdio.h>
2 int main(){
3     int idade_pessoa;
4     printf("Qual sua idade?\n");
5     scanf("%d", &idade_pessoa);
6
7     if (idade_pessoa >= 18){
8         if(idade_pessoa >= 70) {
9             printf("Voto facultativo!\n");
10        }else{
11            printf("Voto obrigatório!\n");
12        }
13    }else{
14        if(idade_pessoa >= 16){
15            printf("Voto facultativo!\n");
16        }
17    }
18    return 0;
19 }
```

Código 21: Exemplo voto com if-else.


```
1 #include <stdio.h>
2 int main() {
3     int idade_pessoa;
4     printf("Qual sua idade?\n");
5     scanf("%d", &idade_pessoa);
6
7     if (idade_pessoa >= 70) {
8         printf("Voto facultativo!\n");
9     } else if (idade_pessoa >= 18) {
10        printf("Voto obrigatório!\n");
11    } else if (idade_pessoa >= 16) {
12        printf("Voto facultativo!\n");
13    }
14
15    return 0;
16 }
```

Código 22: Exemplo voto com else if.

(Deitel; Deitel, 2011) - Capítulo/Seção 2.6 e 3.5.



DE OLIVEIRA, J.F.; MANZANO, J.A.N.G. **Algoritmos: Lógica para Desenvolvimento de Programação de Computadores**. 16. ed. São Paulo: Editora Érica, 2004.

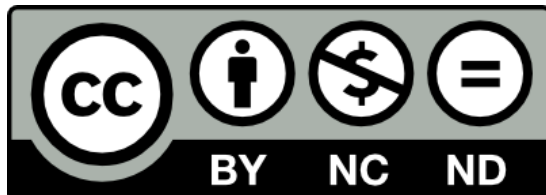
DE SOUZA, M.A.F. *et al.* **Algoritmos e Lógica de Programação**. São Paulo: Thomson Learning, 2004.

DEITEL, Paul; DEITEL, Harvey. **C: Como Programar**. 6. ed. São Paulo: Pearson Universidades, 2011.

MEDINA, M.; FERTIG, C. **Algoritmos e Programação – Teoria e Prática**. São Paulo: Novatec, 2005.

SCHILDT, Herbert. **C Completo e Total: O Guia Definitivo para Programação em C**. 3. ed. São Paulo: Makron Books, 1996. ISBN 978-8534606928.

Estes slides estão protegidos por uma licença Creative Commons



Este modelo foi adaptado de Maxime Chupin.

Fundamentos de Programação

Linguagem de Programação C
Estrutura de Seleção