

Marisangila Alves, MSc
marisangila.alves@udesc.com
marisangila.com.br



JOINVILLE
CENTRO DE CIÊNCIAS
TECNOLÓGICAS

UDESC
Universidade do Estado de Santa Catarina

2025/1

Algoritmo e Linguagem de Programação

Linguagem de Programação C
Estrutura de Repetição Definida

Sumário

1 Operadores Unários

2 Estrutura de Repetição

3 Contadora Múltipla

4 Variável Contadora

5 Variável Acumuladora

6 Condição de Parada Dinâmica

7 Estruturas de Controle

Operadores Unários

C inclui dois operadores, geralmente não encontrados em outras linguagens. São os operadores de incremento (++) e decremento(-) .

Incrementando!

Esse código:

1

```
x = x + 1;
```

é o mesmo que:

1

```
x += 1;
```

que é o mesmo que:

1

```
x++;
```

Acumulando!

Esse código:

1

```
x = x + 10;
```

é o mesmo que:

1

```
x += 10;
```

Operador	Exemplo de expressão	Explicação
++	++a	Incrementa 1 de a e depois usa o novo valor.
++	a++	Usa o valor atual e depois incrementa 1 de a.
-	-a	Decrementa 1 de a e depois usa o novo valor.
-	a-	Usa o valor atual de a e depois decrementa 1 de a.

Tabela 1: Operadores de incremento e decremento.

Operador	Exemplo e Resultado	
+=	a = 10; a += 2;	Resultado: 12
-=	a = 10; a -= 2;	Resultado: 8
*=	a = 10; a *= 2;	Resultado: 20
/=	a = 10; a /= 2;	Resultado: 5
%=	a = 10; a %= 2;	Resultado: 0

Tabela 2: Operadores aritméticos de incremento e decremento.

Estrutura de Repetição

- Quando é necessário que um bloco de instruções seja executado múltiplas vezes.
- É necessário usar uma estrutura de repetição ou laço de repetição, ou ainda *loop*.
- Quando o **número de repetições é definido** usamos o FOR.
- Nessa estrutura há uma variável de controle, chamada: **contadora** que deve ser **incrementada**.

```
1 #include <stdio.h>
2 int main()
3 {
4     int tabuada;
5     printf("Qual tabuada?\n");
6     scanf("%d", &tabuada);
7     printf("%d X 1 = %d\n", tabuada, tabuada * 1);
8     printf("%d X 2 = %d\n", tabuada, tabuada * 2);
9     printf("%d X 3 = %d\n", tabuada, tabuada * 3);
10    printf("%d X 4 = %d\n", tabuada, tabuada * 4);
11    printf("%d X 5 = %d\n", tabuada, tabuada * 5);
12    printf("%d X 6 = %d\n", tabuada, tabuada * 6);
13    printf("%d X 7 = %d\n", tabuada, tabuada * 7);
14    printf("%d X 8 = %d\n", tabuada, tabuada * 8);
15    printf("%d X 9 = %d\n", tabuada, tabuada * 9);
16    printf("%d X 10 = %d\n", tabuada, tabuada * 10);
17    return 0;
18 }
```

Código 1: Exemplo tabuada.

```
1 #include <stdio.h>
2 int main()
3 {
4     float nota1, nota2, nota3, nota4, nota5, total_notas, media;
5     total_notas = 5;
6     printf("Digite uma nota:\n");
7     scanf("%f", &nota1);
8     printf("Digite uma nota:\n");
9     scanf("%f", &nota2);
10    printf("Digite uma nota:\n");
11    scanf("%f", &nota3);
12    printf("Digite uma nota:\n");
13    scanf("%f", &nota4);
14    printf("Digite uma nota:\n");
15    scanf("%f", &nota5);
16    media = (nota1 + nota2 + nota3 + nota4 + nota5) / total_notas;
17    printf("Media: %f", media);
18    return 0;
19 }
```

Código 2: Exemplo notas.

```
1 #include <stdio.h>
2 int main()
3 {
4     for (int i = 0; i < 10; i++)
5     {
6         //Intruções.
7     }
8     return 0;
9 }
```

Código 3: Estrutura básica for.

 Atenção!

Por convenção a contadora deve ser inicializada com valor zero.

```
1 #include <stdio.h>
2 int main()
3 {
4     int contadora;
5     for (contadora = 0; contadora < 10; contadora++)
6     {
7         printf("Hello World!\n");
8     }
9     return 0;
10 }
```

Código 4: Exemplo hello world.

```
1 #include <stdio.h>
2 int main()
3 {
4     int tabuada;
5     printf("Qual tabuada?\n");
6     scanf("%d", &tabuada);
7     for (int i = 0; i < 10; i++)
8     {
9         printf("%d X %d = %d\n", tabuada, i, tabuada * i);
10    }
11    return 0;
12 }
```

Código 5: Exemplo tabuada for.

```
1 #include <stdio.h>
2 int main()
3 {
4     float nota, notas, media;
5     for (int i = 0; i < 10; i++)
6     {
7         printf("Digite uma nota:\n");
8         scanf("%f", &nota);
9         notas = notas + nota;
10    }
11    media = notas / 5;
12    printf("Media: %f", media);
13    return 0;
14 }
```

Código 6: Exemplo notas for.

```
1 int contadora;  
2 for (contadora = 0; contadora < 10; contadora++)  
3 {  
4     // instruções  
5 }
```

Código 7: Variável contadora declarada.

```
1 for (int contadora = 0; contadora < 10; contadora++)  
2 {  
3     // instruções  
4 }
```

Código 8: Variável contadora declarada no for.

Contadora Múltipla

```
1 #include <stdio.h>
2 int main()
3 {
4     for (int contadora = 0; contadora < 10; contadora += 2)
5     {
6         printf("Hello World!\n");
7     }
8     return 0;
9 }
```

Código 9: Variável contadora com incremento diferente de 1.

Variável Contadora

```
1 #include <stdio.h>
2 int main()
3 {
4     int numero;
5     int quantidade_pares = 0;
6     for (int i = 0; i < 50; i++)
7     {
8         if (numero % 2 == 0)
9         {
10             quantidade_pares = quantidade_pares + 1;
11         }
12     }
13     printf("Total de numeros pares: %d\n",quantidade_pares);
14     return 0;
15 }
```

Código 10: Exemplo contar números pares.

```
1 #include <stdio.h>
2 int main()
3 {
4     int numero;
5     int quantidade_pares = 0;
6     for (int i = 0; i < 50; i++)
7     {
8         if (numero % 2 == 0)
9         {
10             quantidade_pares++;
11         }
12     }
13     printf("Total de numeros pares: %d\n",quantidade_pares);
14     return 0;
15 }
```

Código 11: Exemplo contar números pares com operador unário.

Variável Acumuladora

```
1 #include <stdio.h>
2 int main(){
3     float nota, somatorio, media;
4
5     for(int i = 0; i < 5; i++){
6         printf("Informe uma nota:\n");
7         scanf("%f", &nota);
8         somatorio = somatorio + nota;
9     }
10    media = somatorio / 5;
11    printf("%.1f", media);
12    return 0;
13 }
```

Código 12: Variável acumuladora.

```
1 #include <stdio.h>
2 int main(){
3     float nota, somatorio, media;
4
5     for(int i = 0; i < 5; i++){
6         printf("Informe uma nota:\n");
7         scanf("%f", &nota);
8         somatorio += nota;
9     }
10    media = somatorio / 5;
11    printf("%.1f", media);
12    return 0;
13 }
```

Código 13: Variável acumuladora usando +=.

Condição de Parada Dinâmica

```
1 #include <stdio.h>
2 int main(){
3     float nota, somatorio, media;
4     int quantidade_notas;
5
6     printf("Quantas notas?\n");
7     scanf("%d", &quantidade_notas);
8
9     for(int i = 0; i < quantidade_notas; i++){
10         printf("Informe uma nota:\n");
11         scanf("%f", &nota);
12         somatorio += nota;
13     }
14     media = somatorio / quantidade_notas;
15     printf("%.1f", media);
16     return 0;
17 }
```

Código 14: Condição de parada em função de variável.

Estruturas de Controle

- A instrução `break` dentro de um `laço for` é usado para **interromper** a execução do `laço` imediatamente quando uma condição específica é atendida.
- Portanto, permite **sair** do `laço` **antes que ele seja concluído completamente**.

```
1 #include <stdio.h>
2 int main() {
3
4     for (int i = 1; i <= 10; i++) {
5         if (i == 5) {
6             printf("Encontrou 5! Saindo do loop...\n");
7             break;
8         }
9         printf("%d\n", i);
10    }
11    return 0;
12 }
```

Código 15: Exemplo encontrar número.

- A instrução `continue` é usado para **pular** a iteração atual de um laço e **avançar** para a próxima iteração.

```
1 #include <stdio.h>
2 int main() {
3     int i;
4
5     for (i = 1; i <= 10; i++) {
6         if (i % 2 == 0) {
7             continue;
8         }
9         printf("%d ", i);
10    }
11    return 0;
12 }
```

Código 16: Exemplo exibir apenas números ímpares.

(Deitel; Deitel, 2011) - Capítulo/Seção 3.12, 4.2, 4.4, 4.5, 4.6 e 4.9.



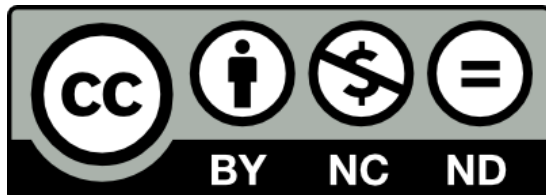
 DE OLIVEIRA, J.F.; MANZANO, J.A.N.G. **Algoritmos: Lógica para Desenvolvimento de Programação de Computadores**. 16. ed. São Paulo: Editora Érica, 2004.

 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.

Estes slides estão protegidos por uma licença Creative Commons



Este modelo foi adaptado de Maxime Chupin.

Marisangila Alves, MSc
marisangila.alves@udesc.com
marisangila.com.br



JOINVILLE
CENTRO DE CIÊNCIAS
TECNOLÓGICAS

UDESC
Universidade do Estado de Santa Catarina

2025/1

Algoritmo e Linguagem de Programação

Linguagem de Programação C
Estrutura de Repetição Definida