

Algoritmo e Linguagem de Programação

Linguagem de Programação C
Estrutura de Repetição Indefinida

Sumário

1 Estrutura

2 Do While

Estrutura

- A estrutura de repetição `for` pode ser usada quando se sabe exatamente quantas iterações (repetições) são necessárias.
- No entanto, há outra forma de repetir instruções. A estrutura de repetição indefinida ou `while`.
- A estrutura `while` pode ser utilizada quando há uma condição de parada ou quando **quantidade de iterações não é conhecida**.
- Para validar a condição de parada do laço, operadores relacionais ou lógicos devem ser utilizados.

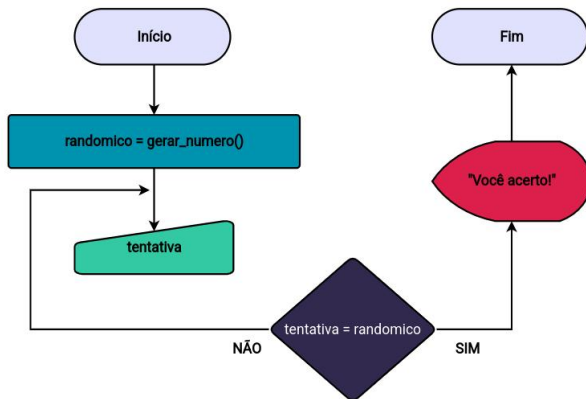


Figura 1: Exemplo acertar número randômico ¹.

- 1 O nome de uma variável de controle (ou contador do laço).
- 2 O valor inicial da variável de controle
- 3 O incremento (ou decremento) pelo qual a variável de controle é modificada cada vez que o laço é realizado.
- 4 A condição que testa o valor final da variável de controle (i.e., se o laço deve continuar).

```
1 while (/*condicao*/)
2 {
3     // instruções
4 }
```

Código 1: Estrutura básica while.

```
1 #include <stdio.h>
2 int main()
3 {
4     int contadora = 0;
5     while (contadora < 10)
6     {
7         printf("%d\n", contadora);
8         contadora++;
9     }
10    printf("Terminou!\n");
11    return 0;
12 }
```

Código 2: Exemplo de incremento de contador.

```
1 #include <stdio.h>
2 int main()
3 {
4     int numero = 10;
5     while (numero >= 0)
6     {
7         printf("Ainda nao terminou!\n");
8         numero--;
9     }
10    printf("Terminou!\n");
11    return 0;
12 }
```

Código 3: Exemplo de decremento de contador.

```
1 #include <stdio.h>
2 int main()
3 {
4     int numero;
5     printf("Escolha um numero\n");
6     scanf("%d", &numero);
7     while (numero < 50)
8     {
9         printf("Escolha outro!\n");
10        scanf("%d", &numero);
11    }
12    return 0;
13 }
```

Código 4: Exemplo de parada com entrada.

```
1 #include <stdio.h>
2 int main()
3 {
4     char opcao;
5     printf("Deseja continuar?\n");
6     printf("Digite s para continuar\n");
7     scanf(" %c", &opcao);
8     while (opcao == 's')
9     {
10         printf("Deseja continuar?\n");
11         printf("Digite s para continuar.\n");
12         scanf(" %c", &opcao);
13     }
14     return 0;
15 }
```

Código 5: Exemplo sair.

```
1 #include <stdio.h>
2 int main()
3 {
4     int numero = 10;
5     while (numero >= 0)
6     {
7         printf("Ao infinito e alem!\n");
8     }
9     return 0;
10 }
```

Código 6: Exemplo de laço infinito.

Atenção!

É fundamental garantir que a variável que controla o laço seja alterada corretamente, para que **a condição de parada se torne verdadeira** em algum momento, evitando a ocorrência de um *loop* infinito.

¹Os números gerados por computadores não são verdadeiramente aleatórios, mas sim pseudoaleatórios, pois são produzidos por algoritmos determinísticos que dependem de um estado inicial ou "semente". 12/23

Do While

- A estrutura de repetição `do/while` é similar à estrutura `while`.
- Na estrutura `do/while` a condição de continuação é testada **depois** de o corpo do loop ser executado.
- O corpo do laço **será executado pelo menos uma vez**, independentemente da condição inicial.

```
1 do
2 {
3     // instruções
4 } while (/* condicao */)
```

Código 7: Estrutura básica do while.

```
1 #include <stdio.h>
2 int main()
3 {
4     int numero = 10;
5     do
6     {
7         printf("Ainda nao terminou!\n");
8         numero--;
9     } while (numero >= 0);
10    printf("Terminou!\n");
11    return 0;
12 }
```

Código 8: Estrutura decremento.

```
1 #include <stdio.h>
2 int main()
3 {
4     int numero;
5     do
6     {
7         printf("Escolha um numero!\n");
8         scanf("%d", &numero);
9     } while (numero < 50);
10    return 0;
11 }
```

Código 9: Exemplo número.

```
1 #include <stdio.h>
2 int main()
3 {
4     char opcao;
5     do{
6         printf("Deseja continuar?\n");
7         printf("Digite s para continuar.\n");
8         scanf(" %c", &opcao);
9     }while (opcao == 's');
10    return 0;
11 }
```

Código 10: Exemplo sair.

- O `while`:
 - Pode ser que o bloco não seja executado nenhuma vez.
- O `do/while`:
 - Repete até uma condição deixar de ser verdadeira, ou seja, se a condição for verdadeira, as instruções dentro do bloco são executadas.
 - Executa pelo menos uma única vez, independentemente da condição inicial.

(Deitel; Deitel, 2011) - Capítulo/Seção 3.7, 4.2, 4.3, 4.8.



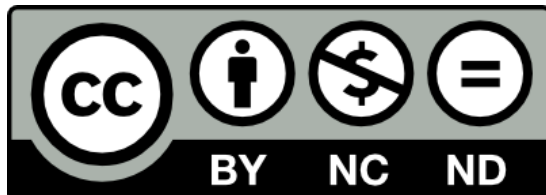
DE OLIVEIRA, J.F.; MANZANO, J.A.N.G. **Algoritmos: Lógica para Desenvolvimento de Programação de Computadores**. 16. ed. São Paulo: Editora Érica, 2004.

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.

Algoritmo e Linguagem de Programação

Linguagem de Programação C
Estrutura de Repetição Indefinida