

Linguagem C - Estruturas de Repetição

PARTE 2

Lógica de Programação

Professor: Vinícius T. Guimarães

viniciusguimaraes@ifsul.edu.br



Tecnólogo em Sistema para Internet
Campus Charqueadas

Introdução

Em nossa primeira aula sobre estruturas de repetição, trabalhamos com o laço **for(...)**, o qual é comumente utilizado para **repetições contadas**, ou seja, aquelas que sabemos **o número de vezes** que precisamos repetir o bloco de comandos.

Hoje vamos dar prosseguimento no estudo das estruturas de repetição, conhecendo mais dois comandos! 😊

Comando While

O comando **while()** nos permite realizar repetições com teste no início.

Vamos dar uma olhada, na forma geral do comando:

```
while(condição)
{
    <bloco de comandos>
}
```

→ Início do Bloco

→ Fim do Bloco

Comando While

A palavra **while**, traduzida para o português, significa **enquanto**. Assim, podemos interpretar o comando da seguinte forma:

“enquanto a condição for verdadeira, execute o bloco de comandos”



```
while(condição)
{
    <bloco de comandos>
}
```

→ Início do Bloco

→ Fim do Bloco

Comando While

A **condição** se caracteriza por ser uma **expressão relacional e/ou lógica** que **determina** quando o **laço deverá ser terminado**, ou seja, determina a **condição de parada** da estrutura de repetição.



```
while(condição)
```

```
{
```

 → Início do Bloco

```
    <bloco de comandos>
```

```
}
```

 → Fim do Bloco

Vamos ver um exemplo!



Exemplo de código while

<https://replit.com/@vicoguim/exemplo-repeticao-2-1>

```
1  #include <stdio.h>
2
3  int main(void) {
4      int i;
5      i = 1; // valor inicial para a variável
6      while(i <= 10){ // teste condicional
7          printf("%i\n",i);
8          i++; // incremento
9      }
10     return 0;
11 }
```

O que esse código faz?

Lógica de Programação



Exemplo de código while

<https://replit.com/@vicoguim/exemplo-repeticao-2-1>

```
1  #include <stdio.h>
2
3  int main(void) {
4      int i;
5      i = 1; // valor inicial para a variável
6      while(i <= 10){ // teste condicional
7          printf("%i\n",i);
8          i++; // incremento
9      }
10     return 0;
11 }
```

Basicamente, esse código imprime os número de **1 até 10**. Agora, vamos cuidar alguns detalhes importantes.

Exemplo de código while

<https://replit.com/@vicoguim/exemplo-repeticao-2-1>

```
1  #include <stdio.h>
2
3  int main(void) {
4      int i;
5      i = 1; // valor inicial para a variável
6      while(i <= 10){ // teste condicional
7          printf("%i\n",i);
8          i++; // incremento
9      }
10     return 0;
11 }
```

1 – Observem que a variável **i** foi iniciada com o valor **1** antes do comando **while** ser executado, pois nossa intenção era que o primeiro teste já resultasse em valor verdadeiro.

Exemplo de código while

<https://replit.com/@vicoguim/exemplo-repeticao-2-1>

```
1  #include <stdio.h>
2
3  int main(void) {
4      int i;
5      i = 1; // valor inicial para a variável
6      while(i <= 10){ // teste condicional
7          printf("%i\n",i);
8          i++; // incremento
9      }
10     return 0;
11 }
```

2 – O teste condicional foi estruturado com uma expressão relacional ***i* <= 10**, ou seja, enquanto o valor da variável ***i*** for menor ou igual a 10, **o bloco de comandos deverá ser repetido**.

Exemplo de código while

<https://replit.com/@vicoguim/exemplo-repeticao-2-1>

```
1  #include <stdio.h>
2
3  int main(void) {
4      int i;
5      i = 1; // valor inicial para a variável
6      while(i <= 10){ // teste condicional
7          printf("%i\n",i);
8          i++; // incremento
9      }
10     return 0;
11 }
```

3 – Observem que dentro do bloco de comandos temos o **i++**. Por que? Pois, se a variável **i** não for atualizada a cada repetição, o valor de **i** iria permanecer em **1** e teríamos um **loop infinito!**

Exemplo de código while

<https://replit.com/@vicoguim/exemplo-repeticao-2-2>

```
1  #include <stdio.h>
2
3  int main(void) {
4      int i;
5      i = 11; // valor inicial para a variável
6      while(i <= 10){ // teste condicional
7          printf("%i\n",i);
8          i++; // incremento
9      }
10     return 0;
11 }
```



Pode acontecer do bloco de comandos **nunca** ser repetido. Para isso, basta que a condição resulte em valor falso já no primeiro teste. **Observe o exemplo!**

Nesse exemplo, iniciamos o valor da variável **i** com **11**. Logo, quando fizer teste condicional do **while**, o resultado será falso e **o bloco da repetição não será executado!** ;)

Exemplo de código while

<https://replit.com/@vicoguim/exemplo-repeticao-2-2>

```
1  #include <stdio.h>
2
3  int main(void) {
4      int i;
5      i = 11; // valor inicial para a variável
6      while(i <= 10){ // teste condicional
7          printf("%i\n",i);
8          i++; // incremento
9      }
10     return 0;
11 }
```



Pode acontecer do bloco de comandos **nunca** ser repetido. Para isso, basta que a condição resulte em valor falso já no primeiro teste. **Observe o exemplo!**

Nesse exemplo, iniciamos o valor da variável **i** com **11**. Logo, quando fizer teste condicional do **while**, o resultado será falso e **o bloco da repetição não será executado!** ;)

Exemplo de código while

<https://replit.com/@vicoguim/exemplo-repeticao-2-3>

```
1  #include <stdio.h>
2
3  int main(void) {
4      int r = 1;
5      float n1,n2,media;
6
7      while(r){
8          printf("Informe a nota 1: ");
9          scanf("%f", &n1);
10         printf("Informe a nota 2: ");
11         scanf("%f", &n2);
12         media = (n1 + n2) / 2;
13         printf("Média = %.2f\n",media);
14         printf("Deseja continuar? 1-SIM | 0-NAO :");
15         scanf("%i", &r);
16     }
17     return 0;
18 }
```

O que esse código faz?



Exemplo de código while

<https://replit.com/@vicoguim/exemplo-repeticao-2-3>

```
1  #include <stdio.h>
2
3  int main(void) {
4      int r = 1;
5      float n1,n2,media;
6
7      while(r){
8          printf("Informe a nota 1: ");
9          scanf("%f", &n1);
10         printf("Informe a nota 2: ");
11         scanf("%f", &n2);
12         media = (n1 + n2) / 2;
13         printf("Média = %.2f\n",media);
14         printf("Deseja continuar? 1-SIM | 0-NAO :");
15         scanf("%i", &r);
16     }
17     return 0;
18 }
```

Solicita duas notas, calcula a média, imprime a média e pergunta se o usuário deseja continuar.



Exemplo de código while

<https://replit.com/@vicoguim/exemplo-repeticao-2-3>

```
1  #include <stdio.h>
2
3  int main(void) {
4      int r = 1;
5      float n1,n2,media;
6
7      while(r){
8          printf("Informe a nota 1: ");
9          scanf("%f", &n1);
10         printf("Informe a nota 2: ");
11         scanf("%f", &n2);
12         media = (n1 + n2) / 2;
13         printf("Média = %.2f\n",media);
14         printf("Deseja continuar? 1-SIM | 0-NAO :");
15         scanf("%i", &r);
16     }
17     return 0;
18 }
```

1 – Observem que esse é um caso em que o **while** é um comando mais adequado que o **for**, pois temos uma repetição que **não é por um número determinado de vezes**, mas sim condicionada ao desejo do usuário.

Exemplo de código while

<https://replit.com/@vicoguim/exemplo-repeticao-2-3>

```
1  #include <stdio.h>
2
3  int main(void) {
4      int r = 1;
5      float n1,n2,media;
6
7      while(r){
8          printf("Informe a nota 1: ");
9          scanf("%f", &n1);
10         printf("Informe a nota 2: ");
11         scanf("%f", &n2);
12         media = (n1 + n2) / 2;
13         printf("Média = %.2f\n",media);
14         printf("Deseja continuar? 1-SIM | 0-NAO :");
15         scanf("%i", &r);
16     }
17     return 0;
18 }
```

2 – A variável **r** foi iniciada com o valor **1** (verdadeiro) para garantir que a **repetição fosse executada pelo menos uma vez**, ou seja, solicitar as duas notas, fazer o cálculo e mostrar a média.

Exemplo de código while

<https://replit.com/@vicoguim/exemplo-repeticao-2-3>

```
1  #include <stdio.h>
2
3  int main(void) {
4      int r = 1;
5      float n1,n2,media;
6
7      while(r){
8          printf("Informe a nota 1: ");
9          scanf("%f", &n1);
10         printf("Informe a nota 2: ");
11         scanf("%f", &n2);
12         media = (n1 + n2) / 2;
13         printf("Média = %.2f\n",media);
14         printf("Deseja continuar? 1-SIM | 0-NAO :");
15         scanf("%i", &r);
16     }
17     return 0;
18 }
```

3 – Observe que na linha 14 indicamos para o usuário o valor **1** (verdadeiro) para continuar lendo notas e o valor **0** (falso) caso o usuário deseje parar. Lembrando que no C qualquer valor diferente de **0** é verdadeiro e o **0** é falso.

Comando Do ... While

O comando **do ... while()** é um comando complementar ao **while()**, nos permitindo fazer o teste no final, **garantindo assim a execução do bloco de comando por pelo menos uma vez.**

Vamos dar uma olhada, na forma geral do comando:

```
do  
{ _____> Início do Bloco  
  <bloco de comandos>  
} _____> Fim do Bloco  
while(condição);
```

Comando Do ... While

A palavra **do**, traduzida para o português, significa **faça**. Assim, podemos interpretar o comando da seguinte forma:

“**faça** (execute) o bloco de comando, **enquanto** a condição for **verdadeira**”



```
do  
{  
    <bloco de comandos>  
}  
while(condição);
```

→ Início do Bloco

→ Fim do Bloco

Comando Do ... While

```
do  
{ _____> Início do Bloco  
  <bloco de comandos>  
} _____> Fim do Bloco  
while(condição);
```



A **condição** se caracteriza por ser uma **expressão relacional e/ou lógica** que **determina** quando o **laço deverá ser terminado**, ou seja, determina a **condição de parada** da estrutura de repetição.

Vamos ver o próximo exemplo que é uma versão do exemplo anterior utilizando do ... while!

Lógica de Programação



Exemplo de código while

<https://replit.com/@vicoguim/exemplo-repeticao-2-4>

```
1  #include <stdio.h>
2
3  int main(void) {
4      int r;
5      float n1,n2,media;
6      do{
7          printf("Informe a nota 1: ");
8          scanf("%f", &n1);
9          printf("Informe a nota 2: ");
10         scanf("%f", &n2);
11         media = (n1 + n2) / 2;
12         printf("Média = %.2f\n",media);
13         printf("Deseja continuar? 1-SIM | 0-NAO :");
14         scanf("%i", &r);
15     }while(r);
16     return 0;
17 }
```

Observem que o **teste é feito no final** e, por isso, não foi necessário iniciar a variável **r**. Basicamente, a variável **r** vai ser testada e, se o resultado for **true**, volta na linha 7 e repete o bloco. Se o resultado for **false**, termina a execução da repetição.

Resumo

- Os comandos **while()** e **do ... while()** são muito parecidos, com a sutil diferença que o **while()** faz o teste no início e o **do ... while()** faz o teste no final.
- Embora possamos fazer repetições contadas tanto com **while()** quanto com **do ... while()**, para esses casos é mais adequado utilizar o comando **for()**, o qual foi pensado para este fim.
- Os comando **while()** e **do ... while()** são recomendados para quando temos situações em que não conhecemos previamente o número de repetições como, por exemplo, quando desejamos que a solicitação de dados seja encerrada a partir de uma ação do usuário ou até que o usuário informe um dado correto.