



Estruturas de Controle de Repetição

Professora: Mirlem Rodrigues Ribeiro Pereira

Email: mirlem.pereira@ifam.edu.br

04 a 05/07/2024

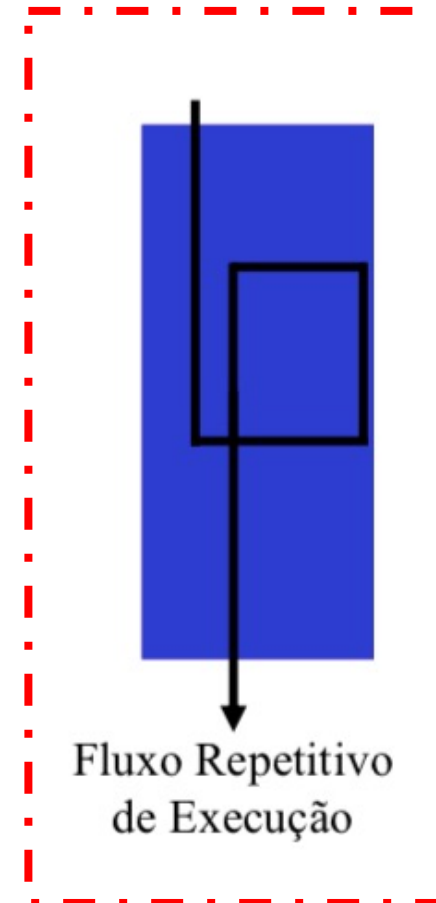
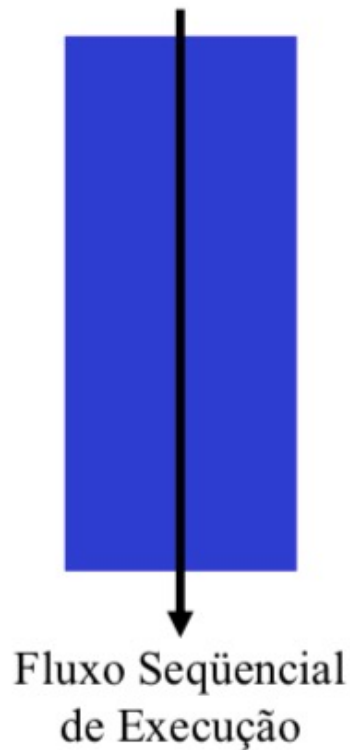
RETIFICANDO O PLANEJAMENTO – JULHO

DATA	CH	DESCRIÇÃO
02	1	Atividade de Revisão
04	2 1	Introdução às Estruturas de Controle de Repetição Revisão
05	2 1	Estruturas de Controle de Repetição Revisão
06	3	Aula Assíncrona – Lista de Repetição
08 a 12	-	Recesso Escolar
16	2 1	Estruturas Homogêneas – Vetores Revisão
18	2 1	Estruturas Homogêneas – Vetores Revisão
23	3	1ª AVALIAÇÃO PARCIAL
25	3	Estruturas Homogêneas – Matrizes
30	3	Estruturas Homogêneas – Matrizes

MONITORIA: A bolsista Viviane atenderá no recesso, terça e qui das 16 às 18h.

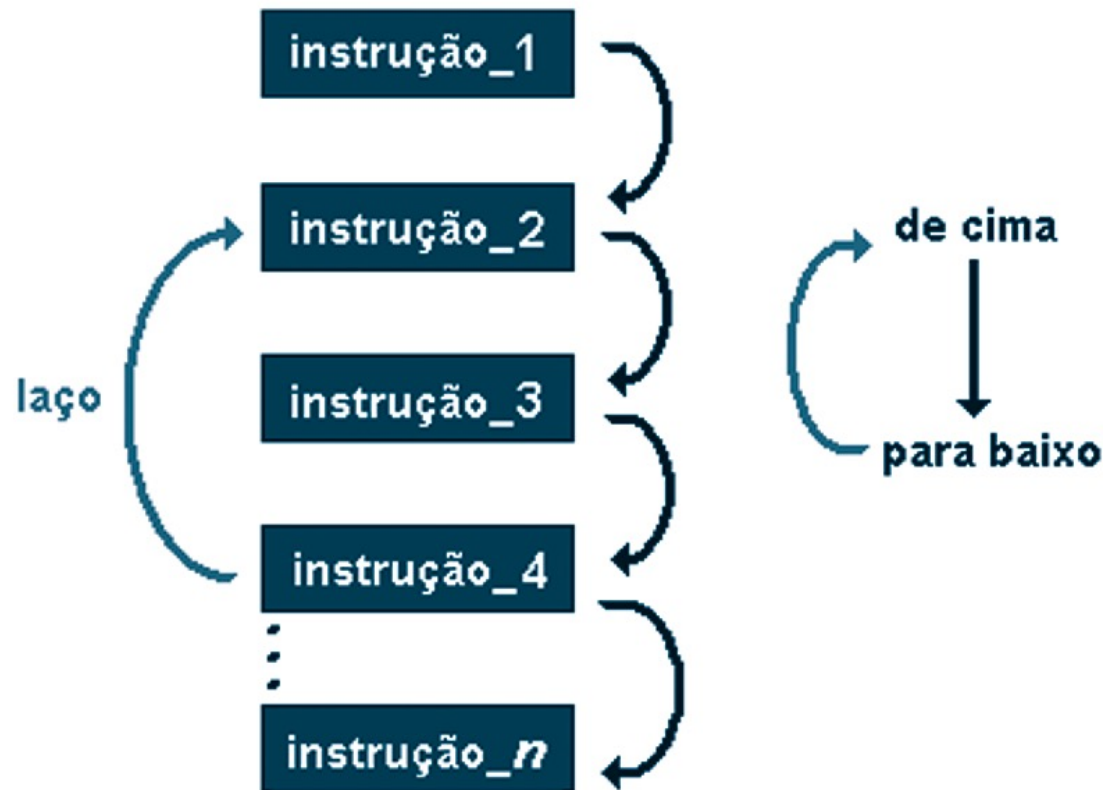
VISÃO GERAL

- ◆ As estruturas de controle são divididas em:
 - ◆ Estruturas Sequenciais
 - ◆ Estruturas de Decisão
 - ◆ Estruturas de Repetição



ESTRUTURAS DE CONTROLE DE REPETIÇÃO

Repetição ou laço





ESTRUTURAS DE CONTROLE DE REPETIÇÃO

- São aquelas que permitem executar mais de uma vez (repetir) um determinado trecho do programa.
- O trecho do programa em repetição é também chamado de laço (ou “loop”).
- As repetições devem ser sempre finitas.



ESTRUTURAS DE CONTROLE DE REPETIÇÃO

- Quanto a quantidade de repetições, os laços podem ser:
 - **Determinados:** Sabe-se antes a quantidade de execuções
 - **Indeterminados:** Não se conhece a quantidade de execuções



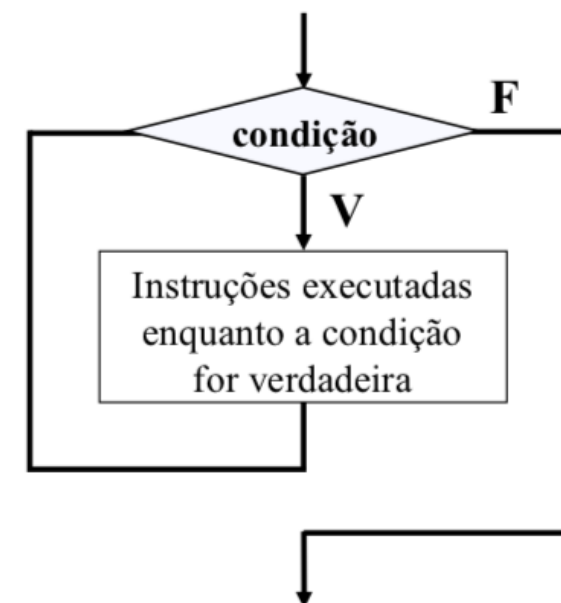
ESTRUTURAS DE CONTROLE DE REPETIÇÃO

- Quanto ao critério de parada, os laços podem utilizar:
 1. Teste no início
 2. Teste no final
 3. Variável de controle

1- REPETIÇÃO COM TESTE NO INÍCIO - **while**

- Verifica antes de cada execução, se é “permitido” executar o trecho do programa.
- É controlada por condição.
- Trata-se de um laço que se mantém repetindo enquanto uma dada condição permanecer verdadeira.

```
while (<condição>) {  
    instrução 1;  
    instrução 2;  
    ...  
    instrução n;  
}
```



Não é necessário usar { } para uma única instrução



1- REPETIÇÃO COM TESTE NO INÍCIO - **while**

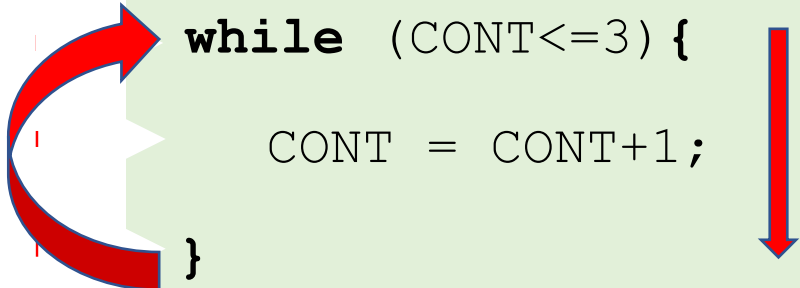
- A elaboração de algoritmos/programas com estas estruturas torna-se necessário o uso de dois tipos de variáveis:
 - Contadoras (**cont**)
 - Acumuladoras (**acm**)
- A **variável contadora** recebe um valor inicial antes do início de uma estrutura de repetição e é incrementada no interior da estrutura de um valor constante, geralmente 1 (um), conforme o exemplo a seguir:

VARIÁVEL CONTADORA

■ INCREMENTO

```
int CONT=1;

while (CONT<=3) {
    CONT = CONT+1;
}
```



CONT	CONT<=3
1	1<=3 (V)
2	2<=3 (V)
3	3<=3 (V)
4	4<=3 (F)

▶ DECREMENTO

```
int CONT=3;

while (CONT>=1) {
    CONT = CONT-1;
}
```

CONT	CONT<=3
3	3>=1 (V)
2	2>=1 (V)
1	1>=1 (V)
0	0>=1 (F)




Exemplo_1: Imprimir “Olá, Pessoal!” 5 vezes

```
#include <stdio.h>

int main() {
    int cont=1;

    while (cont<=5) {
        printf("Olá Pessoal!");
        cont++; //cont=cont+1;
    }
}
```



Exemplo_2: Imprimir os 10 primeiros números inteiros positivos

```
#include <stdio.h>

int main() {
    int numero=1;

    while (numero<=10) {
        printf("%d", numero);
        numero++;
    }
}
```



Exemplo_3: Imprimir os N primeiros números inteiros

```
#include <stdio.h>
```

```
int main() {
```

```
    int N, numero=1;
```

```
    printf("Digite a quantidade de números);
```

```
    scanf("%d", &N);
```

```
    while (numero<=N) {
```

```
        printf("%d", numero);
```

```
        numero++;
```

```
    }
```

```
}
```

Exemplo_4: Calcular a média aritmética para 20 alunos sendo que são fornecidas suas 4 notas. Espera-se, para cada aluno, o cálculo e resultado da sua média e a mensagem “Aluno aprovado, Parabéns!” Quando a média for ≥ 6 ou “Aluno reprovado, estude mais!” quando contrário. Utilize o enquanto faça.

```
#include <stdio.h>

int main() {
    float N1, N2, N3, N4, media;
    int cont=1;
    while (cont<=20) {
        printf("Digite as 4 notas do aluno: ");
        scanf ("%f %f %f %f",&N1, &N2, &N3, &N4);
        media=(N1+N2+N3+N4)/4;
        if (media<=6)
            printf ("Aluno aprovado, Parabéns!");
        else
            printf ("Aluno reprovado, estude mais!");
        cont++;
    }
}
```

Versão A

Exemplo_4: Calcular a média aritmética para N alunos sendo que são fornecidos suas 4 notas. Espera-se, para cada aluno, o cálculo e resultado da sua média e a mensagem “Aluno aprovado, Parabéns!” Quando a média for ≥ 6 ou “Aluno reprovado, estude mais!” quando contrário. Utilize o enquanto faça.

```
#include <stdio.h>
int main() {
    float N1, N2, N3, N4, media;
    int cont=1;

    printf("Quantidade de alunos");


    scanf("%d",&n);
    while(cont<=n) {
        printf("Digite as 4 notas do aluno: ");
        scanf ("%f %f %f %f",&N1, &N2, &N3, &N4);
        media=(N1+N2+N3+N4)/4;
        if (media<=6)
            printf ("Aluno aprovado, Parabéns!");
        else
            printf ("Aluno reprovado, estude mais!");
        cont++;
    }
}
```

Versão B

VARIÁVEL ACUMULADORA

Uma variável acumuladora é uma variável que recebe um **valor inicial**, geralmente **0 (zero)** antes do início de uma estrutura de repetição, e é **incrementada no interior da estrutura de um valor variável**, geralmente a variável usada na estrutura de controle, conforme o exemplo abaixo:

```
int cont=1, num, acm=0;
while(cont<=3){
    printf("Escolha um valor:");
    scanf("%d", &num);
    acm = acm + num; // acumulador
    cont++; // (contador)
}
```



```
printf("Contador: %d\n", cont);
printf("Acumulador: %d\n", acm);
```

CONT	CONT<=3	NUM	ACM
1	1<=3 (V)	15	0+15= 15
2	2<=3 (V)	7	15+7= 22
3	3<=3 (V)	20	22+20= 42
4	4<=3 (F)		



Exemplo_5: Calcular a média das idades de 20 alunos

```
#include <stdio.h>
int main() {
    float mediaIdade;
    int idade, cont=1, acm=0;
    while (cont<=20) {
        printf("Digite a idade: ");
        scanf ("%d", &idade);
        acm=acm+idade;
        cont++;
    }
    mediaIdade=acm/20;
    printf ("A média das idades é de %.0f", mediaIdade);
}
```



EXERCÍCIOS

1. Apresentar os quadrados dos números inteiros de 15 a 35.
2. Apresentar o total da soma dos 100 primeiros números inteiros ($1 + 2 + 3 + 4 + \dots + 99 + 100$).
3. Elaborar um programa que apresente no final o somatório dos valores pares existentes na faixa de 1 até 500.
4. Apresentar os resultados de uma tabuada de um número qualquer. Esta deverá ser impressa no seguinte formato:
Considerando como exemplo o fornecimento do número **2**
 $2 \times 1 = 2$
 $2 \times 2 = 4$
...
 $2 \times 10 = 20$
5. Apresentar as potências de 3 variando de 0 a 15. Deve ser considerado que qualquer número elevado a zero é 1, e elevado a 1 é ele próprio.
6. Elaborar um programa que apresente o valor de uma potência de uma base qualquer elevada a um expoente qualquer, ou seja, de N^M , sem usar a operação de exponenciação (pow).



EXERCÍCIOS

7. Escreva um programa que calcule e apresente o fatorial de N (N!), sendo que o valor inteiro de N é fornecido pelo usuário.

Sabendo que:

$$N! = 1 \times 2 \times 3 \times \dots \times (N-1) \times N$$

$$0! = 1, \text{ por definição}$$

$N < 0$ - não há fatorial de número negativo

Exemplo: $5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$

8. Escreva um programa que apresente a série de Fibonacci até o décimo quinto termo. A série de Fibonacci é formada pela sequência: 1, 1, 2, 3, 5, 8, 13, 21, 34, ... etc. Esta série se caracteriza pela soma de um termo posterior com o seu anterior subsequente.
9. Escreva um programa que verifique se um número fornecido pelo usuário é primo ou não.
10. Escreva um programa que leia um conjunto de N números e mostre qual foi o maior e o menor valor fornecido.