

INF101202

Algoritmos e Programação

Modalidade Ead – Turma H

Material de apoio: capacitar a escrita de algoritmos iterativos

Estruturas lógicas de um programa

- Seqüencial

- Decisão

- Repetição



Motivação: calcular a média aritmética de 40 alunos, cada qual com 4 notas parciais.

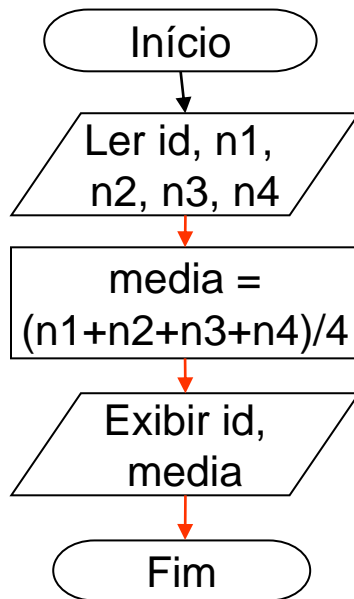
Neste caso:

a solução do problema exige que, para cada um dos 40 alunos, sejam fornecidas as 4 notas parciais e, a seguir, seja calculada a média aritmética dessas notas.

Novidade: a solução deste problema, por exigir a repetição de ações (40 vezes), torna interessante a utilização de uma estrutura lógica que, em programação, chamamos **laço** ou **loop** (em inglês).

Fluxograma: calcular a média aritmética de **1** aluno, cada qual com 4 notas parciais.

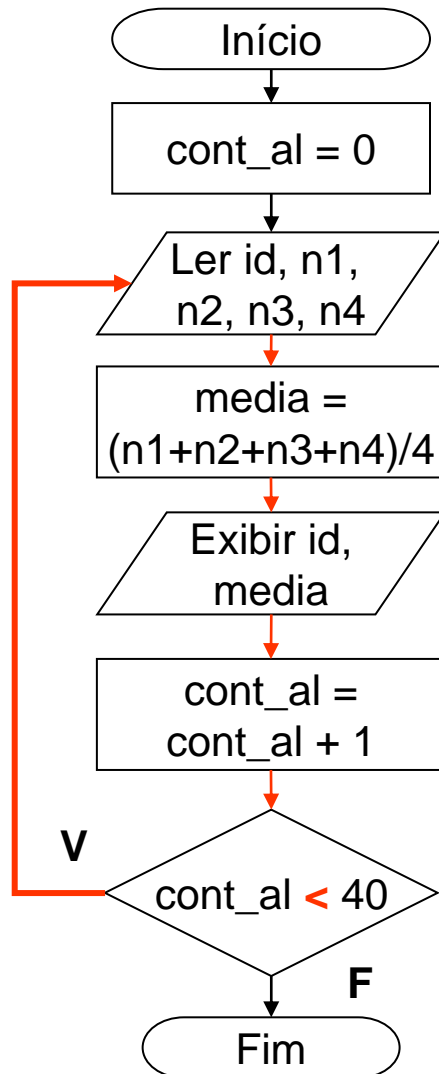
- **Primeiramente**, vamos lembrar a solução deste problema para 1 aluno:



- **Questionamentos ...**
 - Observe que o bloco de comandos entre o **início** e **fim** poderia ser estendido para os 40 alunos:
 - Após ler as notas de um aluno, calcular sua média e exibir o resultado,
 - um novo conjunto de notas poderia ser lido, aproveitando as mesmas variáveis (id, n1, n2, n3, n4) e a media calculada com base nesses novos valores...
 - Nada seria perdido, certo?
 - Mas como garantir que o bloco seja executado 40 vezes?

Fluxograma: solução para o cálculo da média aritmética de 40 alunos...

Teste de Mesa



Iteração	id	n1	n2	n3	n4	media	cont_al
# 1	1	7.5	8.0	6.4	6.0	6.9	1
# 2	2	10.0	8.5	9.0	9.5	9,2	2
...
# 40	40	9,5	8,1	6,0	7,8	7,8	40

Detalhes desta solução:

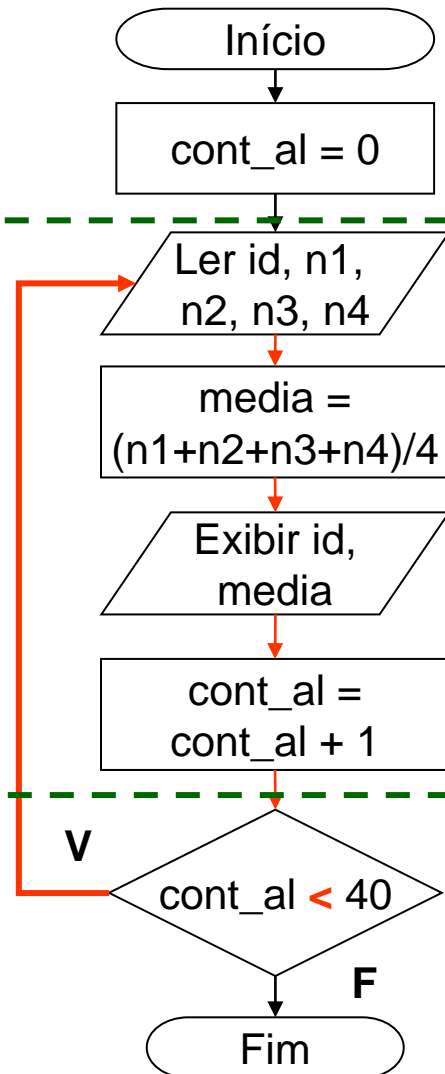
- 1) Pelo Teste de Mesa, a cada **iteração**, **novos valores** são atribuídos às variáveis em questão.
- 2) A variável **cont_al** é inicializada com zero e, a cada iteração, seu valor é acrescido de 1, ou seja, ela assume a função de **contar** o número de iterações;
- 3) A **condição de término** de execução é controlada pela condição (**cont_al < 40**). Quando o resultado for **F**, o programa encerra a execução.

Fluxograma: calcular a média aritmética de 40 alunos, cada qual com 4 notas parciais.

Uma **iteração** corresponde a uma execução do corpo do laço. Neste caso, são 40 iterações.

Laço

Estrutura lógica de programação que permite que **um** comando, ou um **conjunto** de comandos, sejam executados um número **definido** de vezes, ou até que alguma **condição** seja satisfeita!



O corpo do laço é formado por **um** ou **mais** comandos! O número de vezes que eles executam depende do **controle de término** do laço!

Corpo do Laço

O controle de término é definido pelo programador. Neste caso, a variável **cont_al** tem a função de contar até 40!

O comando iterativo while ...

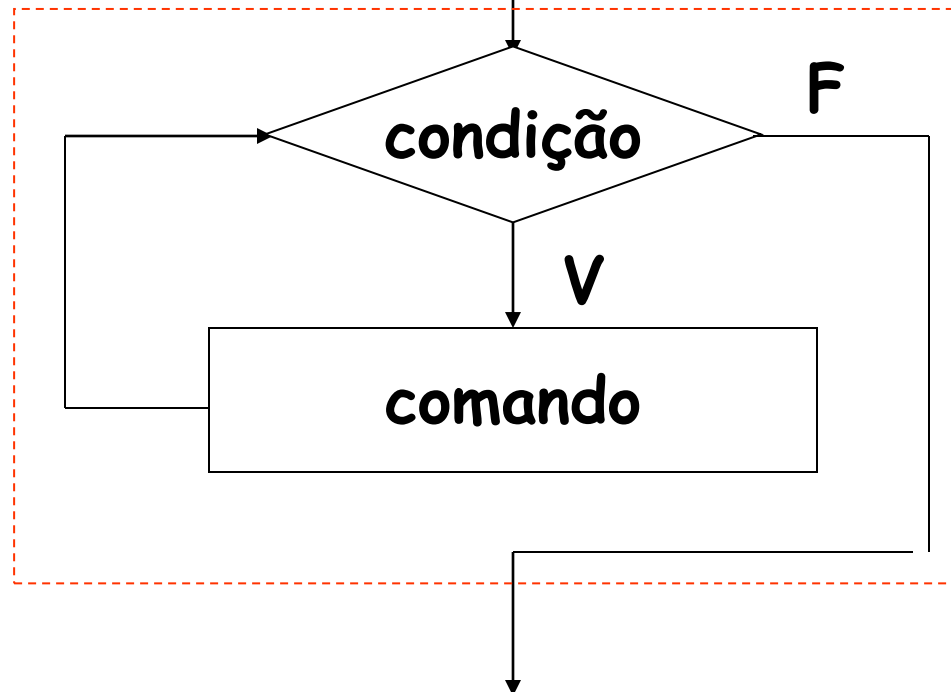
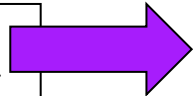
Pseudolinguagem

enquanto ***condição***
corpo;

Linguagem C

while (***condição***)
corpo;

lógica



1 comando ou mais comandos a serem executados repetidas vezes, enquanto a condição resultar verdadeira.

Lembre que, se o corpo contiver mais de um comando, colocá-los entre chaves.

O comando iterativo while ...

Exemplo 1:

```
...  
int k = 1;  
while (k < 4)  
{  
    printf ("Iteração: %d\n", k);  
    k++;  
}  
...
```



Observe que a variável de controle de término k, neste caso, **PRECISA** ser inicializada!

A screenshot of a Windows command prompt window. The title bar reads "C:\Documents and Settings\Cida Souto\Meus docume...". The window contains the following text:
Iteracao: 1
Iteracao: 2
Iteracao: 3
Pressione qualquer tecla para continuar. . . _
A purple arrow from the code block points to the first line of output, "Iteracao: 1".

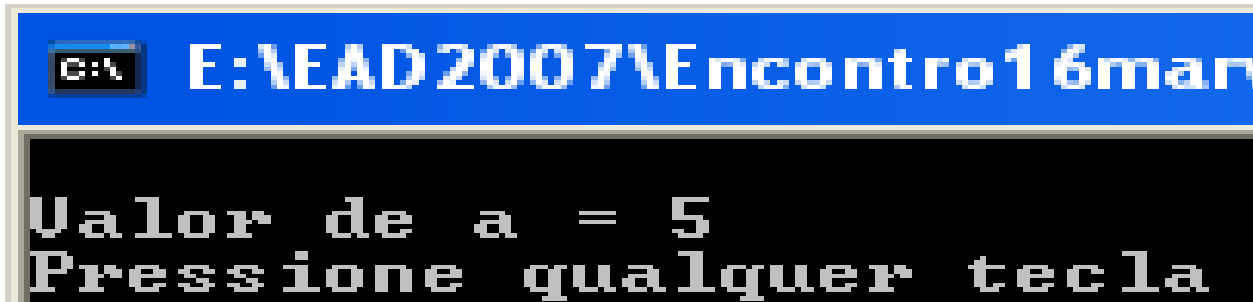
```
C:\Documents and Settings\Cida Souto\Meus docume...  
Iteracao: 1  
Iteracao: 2  
Iteracao: 3  
Pressione qualquer tecla para continuar. . . _
```


Exemplo 2:

```
...  
a = 1;  
while (a < 5)  
    a = a + 1;  
printf("Valor de a = %d\n\n", a);  
...
```

Obs. 1) a variável de controle de término **a**, neste caso, também PRECISOU ser inicializada!

Obs. 2) neste caso, o corpo do while tem apenas 1 comando, dispensando o uso das chaves!

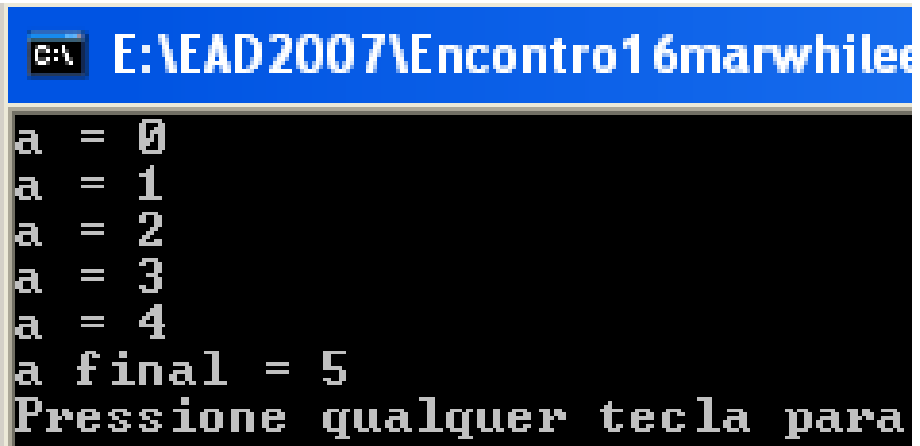


```
C:\> E:\EAD2007\Encontro16marv  
Valor de a = 5  
Pressione qualquer tecla
```

Execução

Mais exemplos:

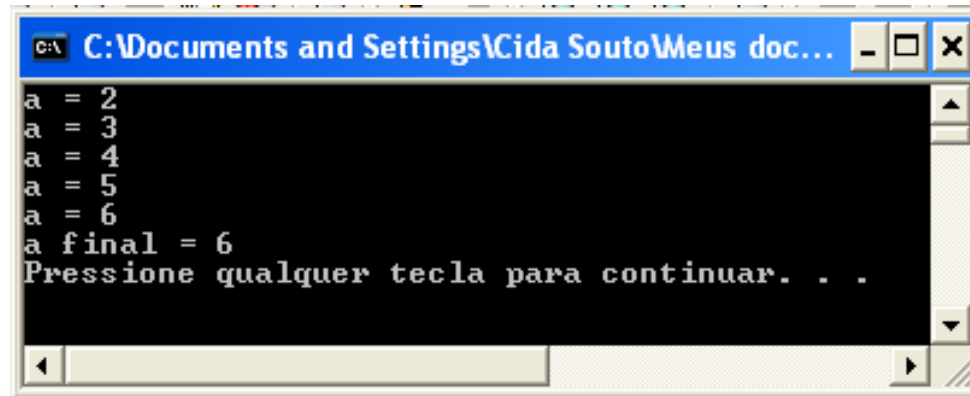
```
...  
a = 0;  
while (a < 5)  
{  
    printf("a = %d\n" , a);  
    a = a + 1;  
}  
printf("a final = %d\n", a);  
...
```



```
C:\ E:\EAD2007\Encontro16marwhilee  
a = 0  
a = 1  
a = 2  
a = 3  
a = 4  
a final = 5  
Pressione qualquer tecla para
```

Execução

```
...  
a = 1;  
while (a <= 5)  
{  
    a = a + 1;  
    printf("a = %d\n", a);  
}  
printf("a final = %d\n", a);  
...
```



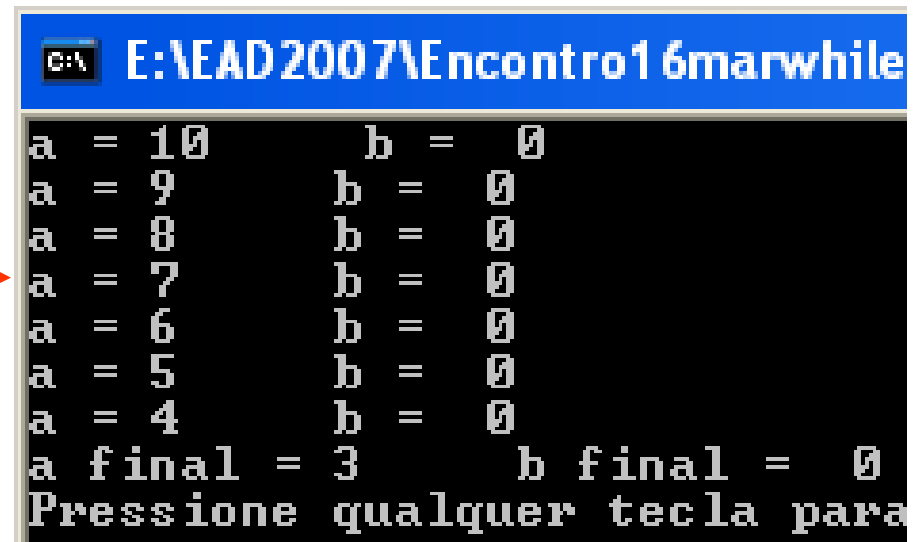
```
C:\Documents and Settings\Cida Souto\Meus doc...  
a = 2  
a = 3  
a = 4  
a = 5  
a = 6  
a final = 6  
Pressione qualquer tecla para continuar. . .
```

Execução

Outro exemplo:

```
...  
a = 10; b = 0;  
while ((a - 2) > (b + 1))  
{  
    printf("a = %d\t\tb = %d\n", a , b);  
    a = a - 1;  
}  
printf ("a final = %d\t\tb final = %d\n", a, b);  
...
```

O resultado é o que
você esperava?



```
C:\ E:\EAD2007\Encontro16marwhile  
a = 10      b = 0  
a = 9       b = 0  
a = 8       b = 0  
a = 7       b = 0  
a = 6       b = 0  
a = 5       b = 0  
a = 4       b = 0  
a final = 3      b final = 0  
Pressione qualquer tecla para
```

Execução

Utilização de variável inteira simulando *booleana* Exemplos

```
...
segue = 1;
while (segue)
{
    scanf("%d", &a);
    if (a != 0)
        printf("%d", a);
    else
        segue = 0;
}
...
```

```
...
parar = 0;
while (!parar)
{
    scanf("%d", &a);
    if (a != 0)
        printf("%d", a);
    else
        parar = 1;
}
...
```

O mesmo que testar se
variável diferente de zero

```
...
s = 0;
scanf("%d", &a);
while (a)
{
    printf("%d", a);
    s = s + 1;
    scanf("%d", &a);
}
printf("%d", s);
...
```

- 1ª vez que testar, **a** já deve ter um valor
- s informa quantos valores foram lidos

O comando iterativo while ...

- Conclusão
 - A variável ou variáveis que fazem parte da **condição de término** deverão ter seu valor **atribuído** através de um comando de **leitura** ou de **atribuição**,
 - Antes da estrutura de repetição while e
 - Dentro da estrutura

O programa: calcular a média aritmética de 40 alunos, cada qual com 4 notas parciais.

```
//Processa notas dos alunos de uma turma
int main ( )
{
    float  n1, n2, n3, n4;  // notas de um aluno
    float ma;
    int c;    // variável de contar alunos
    c = 0 ;  // inicializar contador c em zero
    while ( c <= 40)
    {
        // obtém as 4 notas de um aluno
        scanf("%f %f %f%f", &n1, &n2, &n3,&n4);
        ma = (n1 + n2 + n3 + n4)/4;
        c = c + 1;
        printf("Media do aluno %d : %6.2f", c, ma);
    }
    system(("PAUSE"));
    return 0;
}
```

Refazer o exemplo com um total de 10 alunos e incluir o cálculo da média da turma.

Ex: Processar as notas dos alunos de uma turma de total desconhecido.

Para cada aluno, ler suas 3 notas, calcular e informar a média.

Para encerrar entrar com valor não válido na primeira nota, assume-se -1. Esta adoção de um valor é dito condição de parada. No final, informar a média da turma.

Similar ao exemplo anterior, exceto por parar quando for lida a primeira nota negativa (não é nota válida, é só um sinal - sinal de parada).

```
int main ( )           //Calcula media de alunos
{
    float n1, n2, n3; // notas de um aluno
    float media, soma;
    int c; // variável de contar alunos
    soma = 0;  c = 0;
    scanf ("%f %f %f", &n1, &n2, &n3); // obtém as notas do primeiro aluno
    while (n1 !=-1)      ← // Condição de parada!!
    {
        media = (n1 + n2 + n3) / 3;
        soma = soma + media;
        c = c + 1;
        printf("Media do aluno %d: %7.2 ", c, media);
        scanf ("%f %f %f", &n1, &n2, &n3); // obtém as notas do próximo aluno
    } // fim do processamento de um aluno
    printf("\nMédia da turma: %7.2f\n", soma/c);
    system("PAUSE");
    return 0;
}
```

Cuidado a tomar:

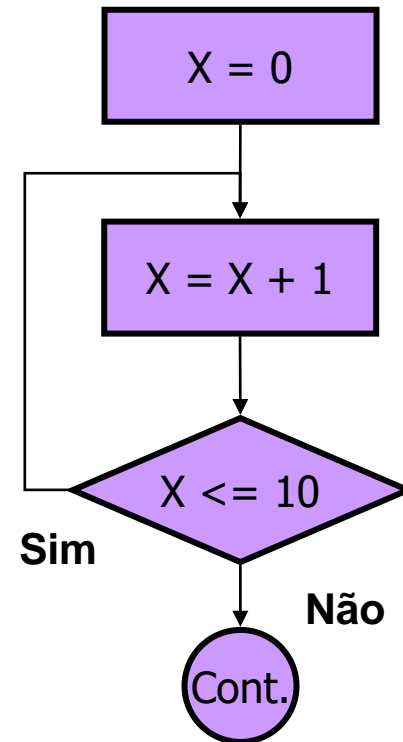
se o 'sinal de parada' faz ou não parte dos dados.

Controle de término de um laço

- O número de vezes que um laço deve ser executado pode ser **CONHECIDO** ou **NÃO**.
- Se **NÃO** for possível conhecer previamente o número de iterações de um laço, **ENTÃO** é necessário estabelecer uma **condição de parada**, sob pena do laço entrar em *loop* infinito.

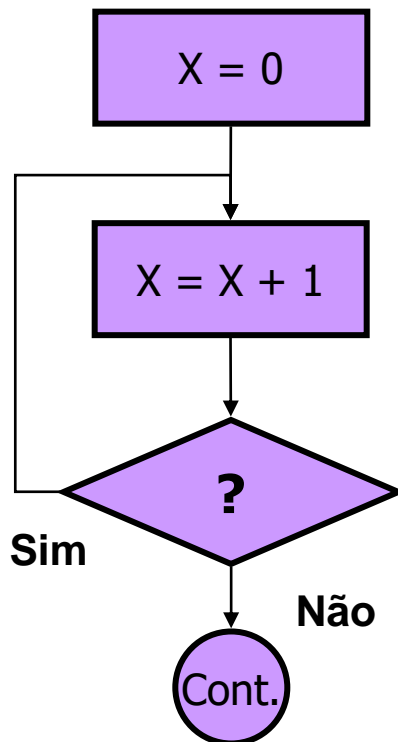
Controle de término de um laço

- Quando o número de iterações é **conhecido**
 - O controle de término é tipicamente feito por contagem, conforme fluxo ao lado.



Controle de término de um laço

- Quando o número de iterações **não é conhecido**
- Como fazer?



Quando a condição de término não é conhecida ...

- **Exemplo:** ler um conjunto não determinado de números de alunos ...
 - Definindo a condição de fim: número de aluno lido negativo
 - A cada novo valor lido, testar se o mesmo é negativo
 - Quando número do aluno lido for negativo, sair do laço

Quando a condição de término não é conhecida ...

- **Mas**, se os valores lidos puderem assumir qualquer valor (inteiro, real, positivo, negativo)?
- **Então:**
 - Método iterativo: comumente usado!
 - usar uma variável inteira adicional **v**, que assume dois valores (p. ex.: $v = 1$ - interessa; $v = 0$ - não interessa) para indicar se o valor lido interessa.

Variável do tipo *contador* ...

- Muito utilizada em programas para **CONTAR** o número de vezes em que um evento (=um comando ou um conjunto de comandos) acontece durante a sua execução
- **Exemplos:**
 - Total de alunos lidos
 - Total de clientes em débito
 - Total de alunos aprovados / reprovados
- **Características:**
 - Uma variável contadora **SEMPRE** inicia com valor zero
 - Uma variável contadora é **INCREMENTADA** de um valor constante igual a um (1) sempre que um novo evento acontece
- **Representação:**
 - **Cont = Cont + 1**

Variável do tipo *acumulador* ...

- Muito utilizada em programas para **SOMAR** (acumular) valores
- Exemplos:
 - Total de receitas no mês de abril
 - Total de despesas no mês de abril
 - Somatório das 4 notas de um aluno
- Características:
 - Uma variável acumuladora SEMPRE inicia com valor zero
 - Uma variável acumuladora é INCREMENTADA de um valor **variável**
- Representação:
 - **Soma = Soma + Valor**