

# Instituto Federal Sudeste de Minas Gerais

## Campus Barbacena

Curso de Tecnologia em Sistemas para Internet

Disciplina: Lógica de Programação

Prof. Wender Magno Cota

Assunto: Estruturas Repetição

# Motivação

Programa que tenha como entrada a nota de cada um dos três alunos de uma turma. Calcular/imprimir a média da turma.

```
#include <stdio.h>
int main(){
    //constante que armazena o número de alunos;
    const int NA=3;
    float nota1,nota2,nota3,media;
    printf("Forneça as 3 notas:");
    scanf("%f %f %f",&nota1,&nota2,&nota3);
    media=(nota1+nota2+nota3)/NA;
    printf("Média da Turma = %.2f",media);
    return 0;
}
```

Problema: Muitas variáveis. Não preciso de armazenar a nota de cada aluno, posso ler e armazenar em uma variável o somatório parcial das notas à medida que o usuário vai fornecendo

# Motivação

```
#include <stdio.h>
int main(){
    //constante que armazena o número de alunos;
    const int NA =3;
    float nota;//armazena a nota de um aluno
    soma=0;//armazena a soma das notas

    printf("Forneça a nota do primeiro aluno:");
    scanf("%f",&nota);
    soma=soma+nota;

    printf("Forneça a nota do segundo aluno:");
    scanf("%f",&nota);
    soma=soma+nota;

    printf("Forneça a nota do terceiro aluno:");
    scanf("%f",&nota);
    soma=soma+nota;

    printf("Média da Turma = %.2f",soma/NA);
    return 0;
}
```

Nota: O mesmo trecho de código sendo escrito repetidas vezes. O problema torna-se impraticável quando o tamanho da intância cresce, que nesse exemplo representa o número de alunos. Solução: Comandos de Repetição.

# Laços de Repetição

Laços(loop) são comandos usados sempre que uma ou mais instruções devam ser repetidas enquanto um certa condição estiver sendo satisfeita

Em C teremos os três comandos abaixo

while

do...while

for

Programação Estruturada: **sequência, desvios e repetição**

# Comando while

```
while(condição){  
    Comando1;  
    Comando2;  
    ...  
    Comandon;  
}
```

Onde:

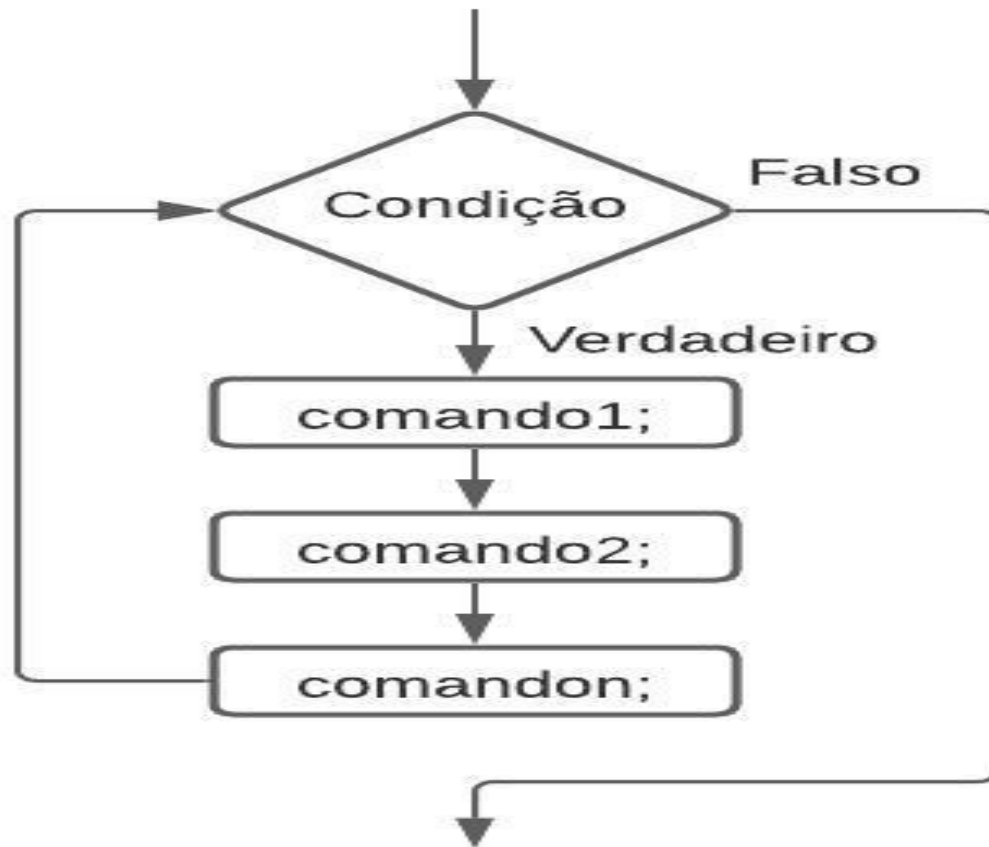
**condição** tem ser uma expressão **lógica**.

**comando1; comando2;...comandon;** pode ser qualquer comando válido da linguagem.

Os **comando1; comando1; comando2;...;comandon;** serão executados enquanto **condição** for verdadeira.

Denomina-se **iteração** a repetição de um conjunto de comandos: cada execução do corpo do laço, juntamente com a condição de terminação do laço, é uma iteração.

# Comando while-Fluxograma



# Comando while

```
#include <stdio.h>
int main(){
    //constante que armazena o número de alunos;
    const int NA =3;
    int contador;//usada para contar o número de iterações
    float nota;//armazena a nota de um aluno
    soma=0;//armazena a soma das notas

    contador=1;
    while(contador<=NA){
        printf("Forneça a nota do primeiro aluno:");
        scanf("%f",&nota);
        //realizando o somatório parcial das notas à medida que o usuário vai fornecendo
        soma=soma+nota;
        contador=contador+1;//indicando que mais um aluno foi processado
    }

    printf("Média da Turma = %.2f",soma/NA);
    return 0;
}
```

# Comando while

- Como o teste é feito no início do comando, poderemos ter a situação onde os comandos pertencentes ao while não serem executados nenhuma vez. Tal situação ocorre quando a condição resultar em um valor falso assim que o fluxo de execução do programa chegar ao while.
- Algo dentro do laço deve ser capaz de modificar o resultado da condição(expressão booleana), caso contrário o laço nunca terminará, e o programa entrará em “loop infinito”.
- No exemplo anterior bastou usar uma variável para contar o número de interação e usando essa variável na expressão que determina o término da execução do while. Isso foi possível, pois está explícito o número de iterações a serem executadas pelo programa.