

5. TÉCNICAS GERAIS DE PROGRAMAÇÃO

5.1. Recursividade

- Algoritmo recursivo é aquele que contém, em sua descrição, uma ou mais chamadas a si mesmo.
- Em C, funções podem chamar a si mesmas; Uma função é recursiva se um comando no corpo da função a chama (direta ou indiretamente - quando uma função X contém uma chamada a outra função Y, que por sua vez contém uma chamada a X).
- Para garantir que uma chamada recursiva não criará um laço que será executado infinitamente, é necessário que ela esteja condicionada a uma expressão lógica que, em algum instante, se tornará falsa e permitirá que a recursão termine.
- A recursividade é uma poderosa ferramenta de programação que é bastante empregada, mas nem sempre ela pode ser aplicada. É preciso analisar o problema e ver se necessita de uma solução recursiva.
- Geralmente uma solução não recursiva é mais eficiente, pois cada chamada recursiva implica em um custo (tempo e espaço - variáveis locais são recriadas a cada chamada, isto é, a cada chamada de procedimento informações são armazenadas na pilha). Entretanto, a recursão quando bem utilizada pode tornar um algoritmo muito elegante (claro, simples e conciso)
- Exemplo:

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

```
long Fat(long n);
```

```
main() {
    int num;
    long result;
    clrscr();
    printf("Digite o numero para calcular fatorial:");
    scanf("%d", &num);
    result = Fat(num);
    printf("\n Resultado: %ld ", result);
    getch();
}
```

```
long Fat (long N) {
    if (N==0)
        return (1);
    else
        return( N * Fat(N-1) );
}
```