**TT-103 Linguagem e Técnicas de Programação I - boas práticas**

**Objetivo**

Indicar quais seriam as boas práticas de programação, ressaltando os vícios normalmente encontrados em C.

**1- Comente adequadamente**

No início de cada arquivo deve ter seu nome, uma descrição do que o arquivo contém, quem o criou e em qual data.

Antes de cada rotina, deve ser explicado o que ela faz, o que ela retorna e qual o comportamento das variáveis. Use como premissa que alguém consegue definir se irá ou não usar a rotina e poderá usá-la apenas lendo o comentário, sem precisar conhecer o código. O uso de linhas com * ou algum outro caracter ajuda a facilitar a identificação das rotinas no meio do código.

Se a rotina ficar grande, é importante explicar na criação das variáveis o que elas fazem.

Comente cada grande parte do código. Por exemplo, quando o programa entra em um if e quando entra em um else.

Mas não precisa comentar coisas óbvias como explicar que contador++ está incrementando o contador.

2- **Dê nomes claros às variáveis**

Uma variável não precisa ter nome e sobrenome, mas não deve ser chamada apenas por uma letra. Seu nome deve ter um significado lógico, de forma que alguém apenas lendo o nome consiga imaginar o que deve ser armazenado na variável.

São exemplos de nomes inadequados: i, j, aux.

**3- Declaração e inicialização de variáveis separadas da lógica**

Tenha como prática declarar todas as variáveis que um programa irá utilizar logo após o título da rotina. Desta forma fica muito mais simples para quem irá fazer a manutenção do código entendê-lo. Depois das declarações e antes da lógica as variáveis devem ser inicializadas. Não faça as coisas todas juntas, pois o código perde em clareza.

**4- Use comandos claros e usualmente conhecidos**

C tem um monte de comandos esdrúxulos e/ou exóticos. Legal que você os descubra como hobby. Mas usá-los não faz o menor sentido, salvo quando não haja um comando tradicional que faça o mesmo. Lembre-se que seu programa poderá ter que ser corrigido por alguém que nunca programou em C até o momento e nem tem a obrigação de conhecer todos os detalhes da linguagem.

### 5- Idente

É algo simples e que facilita incrivelmente tanto a construção de um programa como sua manutenção. Use, se possível, o próprio tab ao invés de espaços, para que seja fácil manter todas as linhas iniciando-se na mesma coluna. Deixe os { e } sempre em linhas independentes, para que seja fácil identificar-se tudo que está entre os dois.

### 6- Não economize {}

Mesmo que seja apenas um comando, é uma boa prática colocar {}. Muitas vezes, em um momento de nervosismo, quando por exemplo um cliente está com a fábrica parada por um erro no código, as pessoas incluem um comando a mais em um else, por exemplo, e se esquecem de checar se tem os { e } correspondentes. Se não houver, o comando ficará em lugar errado e o erro continuará ocorrendo.

### 7- Programe estruturadamente

C não é, *per se*, uma linguagem estruturada. Quem faz ela estruturada é quem programa. Assim, não use em hipótese alguma:
- goto e labels
- returns no meio de rotinas. Cada rotina deve ter apenas um único return, em sua última linha.
- breaks no meio de loops. Deve-se sair de um *loop* apenas com base nas regras lógicas do *loop*.

Não devem existir também segmentos de código iguais em várias partes do programa. Se isso acontecer, avalie como criar uma rotina e chamá-la nessas várias posições do código.
Também deve ser evitado ao máximo o uso de variáveis globais.

### 8- Divida as rotinas de forma adequada

Uma rotina que não cabe em uma tela não é uma rotina interessante. Lógico que em certos casos não ter-se-á como fazer diferente. Mas o ideal é sempre buscar-se construir rotinas pequenas.

### 9- Use tipagem forte logicamente

C é uma linguagem fracamente tipada. Ou seja, ele não checa se você está usando corretamente os tipos da linguagem. Você pode inserir um caracter em uma variável numérica, por exemplo. Pode, mas não deve! (salvo em raríssimas exceções onde isso precisará ser feito).
Tome o cuidado de usar variáveis do tipo correspondente ao seu uso.

### 10- Programe para outros e não para você

Lembre-se que na vida real não será você que irá corrigir o seu programa. Assim, você deve sempre programar de forma simples e clara, de forma que qualquer pessoa que precise alterá-lo o faça facilmente.
Se você pensar que um programa confuso pode garantir seu emprego, provavelmente você vai perder mais rapidamente seu emprego e muito dificilmente conseguirá recomendação para outro, dada a pouca qualidade do seu trabalho.
Lembre-se que você também precisará corrigir programas dos outros. Garanto que se os outros não seguirem as regras acima, você vai entender claramente porque elas são tão importantes!