

Desmistificando a Programação - Parte II

Mário Leite

...

Nas leituras e pesquisas sobre linguagens, pode ser que o programador encontre um *algoritmo* que resolva seu problema; então poderá codificar na linguagem que conhece, a partir da Lógica estabelecida no programa. Por outro lado, o bom profissional não deve ser necessariamente tão "moderninho" e sofisticado, a ponto de achar que *backup* é sinônimo de *xerox*; muitos adoram termos sofisticados para se expressarem e, às vezes, desnecessariamente! Uma coisa que o bom profissional de programação deve perseguir sempre é escrever seus programas de maneira clara para ele e para os colegas de trabalho. Criar rotinas com códigos complicados e cheias de subterfúgios inúteis para "esconder" dos outros é atitude mesquinha. Quando o profissional sabe, mesmo, ele não tem o que temer; mesmo porque, se não houver uma documentação adequada do programa, passado algum tempo nem ele mesmo vai entender o que escreveu antes. É claro que a obra do programador deve ser preservada contra possíveis *piratas*, mas enquanto desenvolvedor numa equipe, a interação com os colegas de projetos deve ser a melhor possível. A manutenção dos programas faz parte da profissão mas, é um dos "abacaxis" que o programador tem que "descascar" a todo momento, pois, o usuário sempre está pedindo algo mais; e é ele quem manda, é ele quem paga, ele é o verdadeiro dono do sistema. Além do mais, os custos de manutenção são os que mais oneram o preço final do *software*; por isto, os programas devem ser criados de modo a não precisarem tanto de manutenções corretivas. Entretanto, o programador não deve criar um sistema que apenas funcione; o sistema deve ser "à prova de balas" e deve poder ser estendido de maneira fácil e sem complicações. Por exemplo, se é um programador que codifica em **C**, use *ponteiros* só quando for realmente preciso; se for "das antigas" e usar o saudoso **Clipper (xHarbour)** só use *Code Blocks* quando for absolutamente necessário, pois senão a rotina vai ficar parecendo uma macarronada! Se for programador **VB** em versões até a 6, sempre declare variáveis e não coloque mais de uma instrução numa mesma linha; e, também, para o VB .Net, use a liberdade de indicar a variável de controle nos *loops For..Next* (por exemplo, **Next i**, **Next j**, **Next k**, em *loops* aninhados, para melhorar a legibilidade da estrutura). Se for programador **Delphi** procure escrever a dupla *begin..end* de modo claro e compreensível, com indentação. Se for programador **Python**, cuidado com as indentações, pois, elas podem atrapalhar a codificação (e mesmo contra as críticas de programadores tradicionais da linguagem, eu inventei meus próprios terminadores de estruturas, que me perdoe, o *Guido van Rossum*). De qualquer forma, o mais importante para qualquer linguagem é manter o código bem legível, usando sempre indentação nas estruturas, pois a coisa mais chata e mais antiprofissional é uma estrutura sem indentação; o código perde toda a elegância.

Continua na Parte III

Para adquirir o *pdf/e-book* dos meus livros sobre programação, entre em contato pelo *e-mail*:
marleite@gmail com
