**Desmistificando a Programação**

**Mário Leite**

[**marleite@gmail.com**](mailto:marleite@gmail.com)

Programar é criar uma rotina para que, seguindo uma lógica passo a passo, pode-se concluir um trabalho qualquer**.** Esta definição, certamente levará o leitor a pensar que programar é criar uma receita de bolo; e é isto mesmo! Afinal, escrever um programa não é nada mais do que criar uma *receita* para resolver um problema qualquer;. então qualquer um pode aprender a programar?! Claro que pode, apesar de alguns acharem que todo programador *nasce* com o dom de programar; não é bem assim..!

O que ocorre na maioria das vezes é que a pessoa interessada em aprender programação, ao ler as revistas e periódicos e publicação de programadores “experts” (as aspas são intencionais) fica com uma imagem um pouco distorcida do que é realmente programação**.** Isto porque temos poucas publicações especializadas, sérias e isentas, que tratam do assunto de maneira clara, sem preconceitos e num nível que os simples “mortais” possam entender**.** O que se vê normalmente é uma avalanche de livros grossos, parecendo uma lista telefônica, cheio de termos estranhos, que desencoraja qualquer um que esteja interessado no assunto; e até nas mídias visuais, ocorre alguma tendência perigosa**.** É claro que não estou fazendo apologia de publicações e cursos baixo nível; o importante é fazer com que o futuro programador não se sinta rejeitado na primeira leitura sobre o assunto**.** E como fazer com que o candidato possa realmente se interessar por um curso de programação sem se chocar com o conteúdo da matéria? Podemos analisar essa questão sob dois ângulos**:** primeiro, se a pessoa não tem conhecimento prévio de nenhuma linguagem ela deve começar estudando *Lógica* *de* *Programação* antes de ver a linguagem propriamente dita, pois sem estudar Lógica não é possível programar; mas isto deve ser feito através de um curso não muito longo, porém objetivo, para não produzir efeitos colaterais no interessado**.** Um curso para ser bom, eficiente, agradável, agregando realmente valor ao aprendizado, não deve ser necessariamente complicado; caso contrário o iniciante em programação desistirá cedo de sua carreira que poderia ser bem promissora**.** Em algumas aulas virtuais, o apresentador, às vezes, sobrecarrega sua exposição com termos em inglês sem necessidade, sem trazer nenhum benefício prático ao curso, e incorrendo em erros gramaticais, realmente sérios, como o famigerado “prá mim fazer”...

Segundo, se o candidato a programador já tem algum conhecimento sobre uma linguagem qualquer, a coisa fica mais tranquila, pois será mais fácil o trabalho de aprendizagem e conscientização sobre a profissão**.** Sim, a conscientização do profissional é o passo inicial mais importante; ele deve saber de imediato que programar, apesar de não ser difícil, deve ser levado muito a sério e que o estudo e a pesquisa devem ser diários**.** É importante saber que a Informática é a ciência que evolui mais rapidamente no mundo, e que para estar atualizado na linguagem (ou nas linguagens) escolhida(s) deverá adquirir revistas, livros, periódicos, etc, enfim, investir em tudo que se relaciona com programação na linguagem e também com as outras, mas tendo a percepção de que a programação em si não depende da linguagem, mas, basicamente, de Lógica**.** Por exemplo, pode ser que numa publicação sobre uma outra linguagem, que não seja a sua, ele encontre um *algoritmo* que resolva seu problema; então poderá codificar na linguagem que conhece, a partir da Lógica estabelecida**.** Por outro lado, o bom profissional não deve ser necessariamente tão "moderninho" (as aspas são intencionais) e sofisticado,a ponto de achar que *back-up* é sinônimo de *xerox*; muitos adoram termos sofisticados para se expressarem; mas, às vezes isto não é necessário!. 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 cheios 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 projeto deve ser a melhor possível**.** A manutenção nos programas é coisa rotineira e 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 manutençõescorretivas**.**

Por outro lado, o programador não deve criar um sistema que apenas funcione; o sistema deve ser à prova de balas; deve poder ser estendido de maneira fácil e sem complicações**.** Por exemplo, se é um programador **C** use *ponteiros* só quando for realmente preciso; se for “das antigas” e usar o saudoso **Clipper xHarhour** só use *Code Blocks* quando for absolutamente necessário, pois senão sua 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, se for **Delphi** procure escrever a dupla *begin ...end* de modo claro e compreensível; e o mais importante: use sempre endentação nas estruturas, pois a coisa mais horrível e mais antiprofissional é uma estrutura sem endentação, o código perde toda a elegância**.**

Bem, eu comecei este *paper* dizendo que programar nada mais é do que criar uma receita que, se for seguida resolverá um problema; a receita de um bolo é o exemplo clássico de uma rotina...

Mas, e a Programação Orientada a Objetos e Orientada a Eventos, como é que ficam nessa estória ?!

Esse tipo de programação difere um pouco da programação tradicional do tipo receita de bolo, pois são os eventos sofridos pelos *objetos* que vão ditar o fluxo do programa**.** Mas, por outro lado, pode ser afirmado que as rotinas que comandam as ações disparadas pelo usuário são procedurais! É isto mesmo; apesar de termos um novo paradigma na programação com o advento desses novos tipos de desenvolvimento no ambiente Windows©, as rotinas são procedurais; isto é, as instruções são executadas do início para o fim, uma a uma**.** Isto porque, graças a Deus, o computador ainda não pode decidir por ele mesmo, a despeito das linguagens de Inteligência Artificial onde pesquisadores estão tentando tornar a CPU inteligente a ponto de tomar decisões; mas esse é um assunto muito polêmico e discutível, e foge a este escopo**.** De qualquer forma, sempre é possível encarar um programa como sendo uma receita, não interessa o tipo de linguagem: funcional, imperativa, orientada a eventos, orientada ao objeto, etc, etc, e não importando o seu nível de complexidade**.** De qualquer forma, o bom profissional em programação deve sempre desenvolver sistemas tais que minimizam a taxa de crescimento dos custos de *software*, pois enquanto os custos do *hardware* diminuem drasticamente, o mesmo não acontece com os custos dos programas**.** E para minorar esses custos, os sistemas devem ser escritos de forma que possam ser reaproveitados (um dos princípios da Orientação ao Objeto) e modulares, além de serem extensíveis.

Mas, afinal de contas, programação é arte ou ciência? Segundo os dicionários, programação significa: “elaboração de um programa para um computador”. Isto pode ser traduzido tecnicamente como: *programar é o ato de escrever um conjunto de ordens a serem executadas pelo computador para* *obter os resultados desejados***.** É aí então que alguns dizem que programar depende de criatividade o que seria um dom nato das pessoas; mas, os programas também necessitam ser eficientes e robustos, pelas razões expostas anteriormente; e com as novas técnicas de desenvolvimento o programador, mesmo não sendo um artista nato, pode perfeitamente se tornar um grande profissional; basta estudar muito e levar a sério o seu trabalho**.** Portanto, não existe esse “papo" de super raça em programação; você também pode ..!