

Programação e Codificação - Parte I

Mário Leite

...

Criar e desenvolver um programa para resolver um problema a partir da sua execução num computador envolve três ações que devem ser observadas pelos profissionais de desenvolvimento de sistemas: “O QUE FAZER”, “COMO FAZER” e “FAZER”. Este *post* discute dois assuntos fundamentais que compõe estas três ações, necessárias para solucionar problemas com a computação eletrônica: Programação e Codificação. Infelizmente, na maioria das vezes, Programação e Codificação são consideradas a “mesma coisa” pela grande maioria dos profissionais de Informática; e isto, a meu ver, contamina o aprendizado já no início, criando “profissionais” que acham que a linguagem de programação vai resolver todos os seus problemas. Entretanto, na verdade, são duas coisas diferentes que se completam; aqui será demonstrada essa diferença, que deve ser bem entendida pelos programadores; principalmente iniciantes. Embora, dizer que “*eu programo em Java*” não altere o fato de o programador saber codificar em Java, é importante ter consciência de que a Codificação não resolve o problema; apenas o coloca em condições de ser executado pelo computador; mas o trabalho “pesado” e intelectual é feito na etapa de Programação.

Desde os primórdios da Programação, quando Euclides de Alexandria criou um algoritmo - que leva seu nome - para calcular o MDC de dois números, “O QUE FAZER” e “COMO FAZER” sempre precederam o “FAZER”. Assim, ele teve que pensar/planejar/formalizar a solução do problema. E, além disto, também descobriu várias coisas durante sua “programação”: uma delas é que se subtrairmos o menor número do maior o MDC não se altera; e mais ainda: como ele mesmo já tinha conhecimento dos números primos, concluiu também, que se um dos números for primo o MDC será 1. E naquela época não havia Codificação! Outro exemplo mais recente, e transformado em filme, foi a história de Alan Mathison Turing; famoso matemático e criptoanalista inglês que conseguiu decifrar as mensagens codificadas pela máquina Enigma dos nazistas durante a Segunda Guerra Mundial, através da programação de decodificação numa máquina tratada por ele de “Bomba Eletrônica”. Esta máquina foi previamente programada com as instruções que ele desenvolveu em algoritmo; mas a máquina não criou nada, apenas automatizou a solução dada por Alan e sua equipe. Ele programava e depois introduzia as instruções na máquina para que a mensagem fosse mostrada de maneira legível. Este é mais um exemplo de que a Codificação apenas reproduz o que a lógica da Programação exige! Por isto, é muito importante não confundir Programação com Codificação; pois programar é ENTENDER O PROBLEMA, PENSAR E CRIAR A SOLUÇÃO; isto é o que define a parte de Programação propriamente dita. Então, somente DEPOIS é que se deve pensar no FAZER (codificar a solução dentro da sintaxe exigida pela linguagem de programação). Isto significa que a etapa de Codificação (o FAZER) é a última na criação/desenvolvimento de um programa; a programação propriamente dita não depende de nenhuma linguagem e de nenhuma máquina. Portanto, ao codificar o programa, a solução do problema já deve estar previamente pronta e correta. Na parte de Programação é onde a solução do problema é criada; e esta solução deve envolver a Lógica de Programação planejada e “desenhada” através de algum algoritmo, com “lápiz e papel” e/ou com o auxílio de algum editor de texto (sem formatação). Deste modo, com a solução do problema formalizada num algoritmo, ela deverá ser testada com alguma ferramenta. Então, só depois que a solução algoritmizada estiver testada e com os resultados corretos, é que se deve partir para a automatização do algoritmo, que nada mais é do que a Codificação numa linguagem selecionada pelo programador. Portanto, ao codificar um programa este já deve estar correto e funcional, pois nesta etapa não é o momento de verificar se ele está correto; é possível até trabalhar na eficiência em função dos recursos oferecidos pela linguagem, mas não na sua eficácia!

Continua na Parte II