Criando a Solução de um Problema

Mário Leite

...

A criação de um *programa* para obter a solução de um problema envolve várias etapas que devem ser executadas pelos analistas de sistemas e programadores. De qualquer forma, o que existe é o problema REAL que está na cabeça do usuário: a sua realidade. O que tem de ser feito é trazer essa realidade ao nível computacional. Veja o esquema da **figura 1** abaixo.

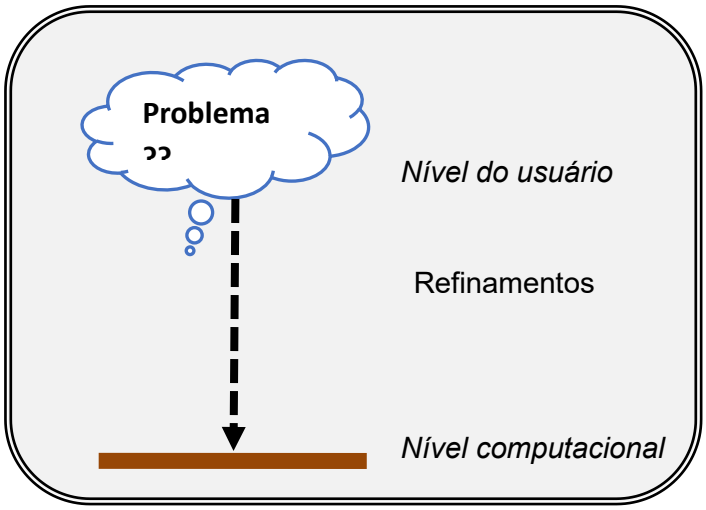

Figura 1 - Trazendo a realidade para o nível computacional

A **figura 1** indica que o problema REAL está na cabeça do usuário e só ele sabe o que REALMENTE quer; mas como não sabe resolver o problema, então ele o entrega para os profissionais de TI, que através de refinamentos de modelos possam resolvê-lo trazendo-o ao nível computacional. O que esses profissionais fazem é refinar a solução através de várias etapas, criando modelos que devem ser tratados até chegar à solução final em nível computacional eficiente e eficaz. Mas, como esses modelos são objetos de estudos mais avançados na Engenharia de Software, aqui, para o nível desta postagem, isto não interessa em detalhes. Porém, mesmo sabendo que os problemas são diferentes e, consequentemente as soluções, no momento de criar um programa em nível computacional algumas ações básicas devem ser consideradas:

1ª) Analisar o problema: entender bem a situação, conversando muito com o usuário.

2ª) Criar um modelo que melhor se adapta ao tipo de problema proposto que, dependendo do tipo de problema esse modelo poderá ser simples ou complexo.

3ª) Criar os algoritmos para a solução do problema.

4ª) Formatar e otimizar os algoritmos através de pseudocódigos.

5ª) Implementar (*codificar)* os pseudocódigos em uma linguagem de programação mais adequada.

E, embora a solução de um problema seja sempre diferente da solução de outros, as cinco etapas apresentadas acima sempre deverão ser seguidas para que a solução adotada seja a melhor possível. A solução, embora seja bem particular a cada programador, deve ser eficiente e tem que ser eficaz; isto é, além de ser correta ela deve agradar totalmente o mais interessado: o **usuário**.

Observe a **figura 2** onde é apresentado o problema inicial e a solução; nos dois extremos está uma figura oculta, mas onipresente: o **Sr. Usuário**. Para ele a *quadratura* do problema tem que se transformar numa solução *redondinha*, não importa o que o programador faça ou invente! O que se deseja é resolver o problema; os cinco passos mostrados (naquela ordem) são fundamentais

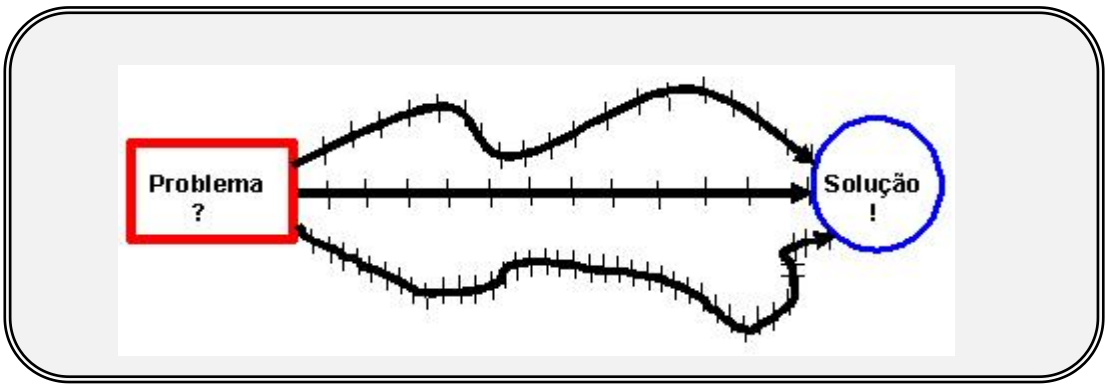

Figura 2 - Escolhendo o caminho da solução de um problema

A **figura 2.** apresenta três possíveis "caminhos" para se obter a solução do problema proposto pelo usuário. Qual deles seria o melhor?

a. O caminho superior: com 19 passos?
b. O caminho do centro: com 12 passos?
c. O caminho inferior: com mais de 30 passos?

Se esta pergunta fosse feita a um matemático ele certamente responderia que seria o caminho do centro, pois a geometria euclidiana ensina que "*a menor distância entre dois pontos é o segmento de reta que os une*". Além disto, do ponto de vista físico, a solução estaria pronta com apenas 12 passos. Mas a questão aqui não é puramente geométrica, e sim de lógica de programação; e nem sempre o caminho mais curto (leia-se menos instruções) produz a melhor solução. Isto acontece porque o produto final da execução destas instruções é algo abstrato e imponderável, mas real: um *software*. Do ponto de vista de eficiência o caminho reto é mesmo o melhor; mas e em termos de eficácia? A eficácia é dada por quem se interessa diretamente pelo produto final: o USUÁRIO. E nem sempre aquela solução sofisticada (cheia de rotinas recursivas, *loops* maravilhosos, telas bonitas ...) criada pelo programador, é aprovada pelo USUÁRIO!

A melhor solução para um programa é aquela que o usuário aprova, e não aquela que o programador "acha" que é a melhor. É claro que se a solução dada pelo programador agradar o usuário, tudo bem; teria eficiência e eficácia ótimas; mas é preciso entender que a solução deve ser aprovada pelo cliente, e isto é uma verdade incontestável, principalmente quando se trata de programas com interfaces gráficas.

Conclusão: "*Programar é definir a sequência lógica de instruções a serem executas para que se obtenha a solução do problema proposto*"; e essa *solução tem que agradar o usuário final*. Portanto, só DEPOIS que a solução do problema estiver criada é que a codificação, em alguma linguagem de programação, deve ser implementada; a codificação é a etapa final do processo, e não deve atropelar a criação da solução.

Nota1: Postagem baseada no livro: "*Curso Básico de Programação: Teoria e Prática*".
Publicado pelo autor na "Amazon", "Clube de Autores" e "Editora Ciência Moderna".

https://www.amazon.com.br/Curso-B%C3%A1sico-Programa%C3%A7%C3%A3o-Teoria-Pr%C3%A1tica/dp/8539908700

Nota2: Acesse o *link* abaixo para ver meus mais recentes livros de Python publicado pelo "Clube de Autores", no formato impresso, da coleção "***1001 Programas em Python Para Você Aprender Praticando***":
Volume1: Nível Básico (500 programas)
Volume2: Nível Intermediário (300 programas)
Volume3: Nível Avançado (201 programas)

https://clubedeautores.com.br/livros/autores/mario-leite