

## Sobre o aprendizado de programação

Gustavo Noronha Silva<sup>1</sup>

**NOTA:** Esse artigo não está pronto, mas fica licenciado sob a GNU Free Documentation License, que pode ser encontrada no site da Free Software Foundation: <http://www.fsf.org/>.

Uma grande dificuldade que percebo nas pessoas que se empenham em aprender a programar é a falta de costume que têm de entender conceitos e implicações.

O que as escolas e universidades parecem esperar de seus estudantes é a simples apreensão de leis e fatos como, por exemplo, o que vem a ser um sistema operacional e os tipos de sistemas operacionais, ou que, em *sistemas operacionais* que fazem um gerenciamento de memória em *modo protegido*, diferentes *processos* não podem acessar livremente qualquer área de memória a não ser aquela reservada para si, tendo de passar para o *núcleo* qualquer solicitação desse tipo que for necessária.

Muito bem. A concepção atual de educação que ronda as escolas que ensinam programação parece ser a de que a simples memorização desses fatos e regras bastam para dizer que houve “aprendizado”. No meu caso, chamo esse tipo de conhecimento de *conhecimento descritivo* e digo que, embora seja de extrema importância, não passa de um conhecimento superficial e panorâmico do objeto de estudo.

Vamos investigar mais a fundo esse problema: o conhecimento descritivo é útil enquanto possibilitador do que chamarei de *conhecimento imperativo*. O conhecimento imperativo é o conhecimento das causas e implicações de fatos e regras, ou leis. Vamos tomar novamente o exemplo da regra que diz que um processo, em um sistema operacional que tenha gerência de memória feita em modo protegido: um estudante “nota 10” seria capaz de me repetir essa regra em quantas provas eu quisesse dar, mas seria ele capaz de usar esse conhecimento descritivo para, entendendo suas implicações resolver questões mais complexas como *Por que é que um softmodem precisa de um controlador de dispositivo que vá dentro do núcleo, não bastando para fazê-lo funcionar uma aplicação “servidor de modems”?*

O que se tem de compreender, quando se trata de desenvolvedores e cientistas da computação<sup>2</sup> é que, enquanto tais, têm a tarefa não de conhecer os processos e saber executá-los, mas de entender as implicações dos problemas que quer resolver e *criar*,

---

<sup>1</sup>Acadêmico do 3º período de Ciências Sociais e Desenvolvedor do Projeto Debian.

<sup>2</sup>Na verdade, essa é uma habilidade necessária a qualquer tipo de atividade humana, cito esses exemplos por serem meu foco de estudo atual.

a partir dos conhecimentos descritivos que detém, um processo qualquer para resolver eficiente e eficazmente o problema: a execução de tal processo fica à cargo da máquina. Para ser mais claro: o programador tem de saber ensinar a máquina o que fazer, e não o contrário.