

## Paradigmas: Para que servem!? - Parte II

Mário Leite

...

Na **Parte I** destes artigos eu falei sobre “paradigma” de um modo geral e esclarecendo que, efetivamente, é um modo de fazer determinada “coisa” dentro de um padrão preestabelecido; seria como um “protocolo chic” de programação. Também, ao final daquele texto, deixei claro que as linguagens de programação podem ser classificadas quanto ao tipo de paradigma em que estão enquadradas. Foi mostrado que a literatura destaca sete principais tipos de paradigmas para classificar as linguagens de programação, embora este número possa variar de acordo com a fonte consultada.

### 1) Paradigma Funcional

Neste paradigma a escrita da linguagem é focalizada nas funções, como na matemática, com resultados oriundos de alguma  $y=f(x)$ . As linguagens que seguem este padrão têm como exemplos: *F#, Lisp, Heskell, Erlang, Elixir, Opa, Scala, R, etc*; e sendo considerada multiparadigma, o *PHP* também pode ser um exemplo de linguagem funcional.

### 2) Paradigma Imperativo

Neste padrão estão definidas todas as linguagens que seguem a chamada “*programação procedural*”. Deste modo, todas as linguagens clássicas estão enquadradas neste tipo de paradigma, além de algumas consideradas modernas: *Ada, Algol, Fortran, Cobol, Basic, Pascal, C, Lua, PHP, Python, etc*. Todas as linguagens enquadradas neste paradigma focalizam a mudança de estado de variáveis, baseando em rotinas que estruturam o programa. Uma observação interessante é que, mesmo sendo consideradas “modernas”, *PHP* e *Python*, seguem este paradigma que é baseado na “velha” arquitetura de Von Neumann.

### 3) Paradigma Declarativo

O paradigma *declarativo* é baseado nos padrões *lógico* e *funcional*, de modo que as linguagens assim classificadas descrevem **O QUE FAZEM**, encapsulando o COMO FAZEM. Entre os exemplos mais significativos estão as chamadas “Linguagens de marcação de texto”, como: *HTML, XHTML, XML, XSLT, XAML* etc. Entretanto, mesmo não sendo de marcação de texto, a *Prolog* - reconhecida como uma legítima representante do paradigma *lógico* - também é uma linguagem declarativa; assim como o *SQL* para pesquisas em bancos de dados.

### 4) Paradigma Estrutural

Linguagens que seguem este paradigma dão ênfase ao uso de sub-rotinas, laços de repetição, decisões condicionais e estruturas em bloco. Uma das literaturas consultadas diz o seguinte:

*“Os paradigmas de programação são muitas vezes diferenciados pelas técnicas de programação que proíbem ou permitem. Por exemplo, **a Programação Estruturada não permite o uso de goto**. Esse é um dos motivos pelo qual novos paradigmas são considerados mais rígidos que estilos tradicionais. Apesar disso, ...”*

Então, de acordo com as diretrizes deste paradigma, os programas codificados neste padrão fazem uso de sub-rotinas na sua modularização e só podem usar as estruturas de controle: *decisão, seleção e repetição*; o uso de saltos incondicionais com **goto** é proibido(!). Exemplos de linguagens que adotam este paradigma são: *Algol, Fortran, Cobol, Basic, Python, PHP, Pascal, C*; ou seja, todas as linguagens ditas tradicionais de alto nível estão enquadradas neste paradigma. Aqui um ponto de interrogação e reflexão: **Pascal** é **C** são linguagens altamente modulares; também são *imperativas e estruturais*, mas, como permitem o **goto**!?

---

**Continua na Parte III**