

## Clipper: “Velho: mas não Obsoleto” - Parte II

Mário Leite

...

O Clipper, desde a sua primeira versão Nantucket Summer’84 como compilador para o dBase no sistema operacional DOS, e até a última comercializada como CA-Clipper 5.3B no ambiente do Windows, foi (é) uma linguagem compilada de altíssimo poder de processamento e até com recursos de Orientação a Objetos, mesmo que limitada a apenas quatro classes: TBrowse, TBColumn, Get e Error. A linguagem possuía a tecnologia *rushmore*, o que lhe conferia um desempenho bem acima das linguagens comerciais; e foi adquirida pela Microsoft para integrar o seu recente Access, já naquela época. E embora com recursos limitados de OOP, o Clipper utilizava as classes TBrowse e Get para o desenvolvimento de telas bem amigáveis nas entradas de dados, tal como os *formulários* de hoje em dia. As instâncias desses dois objetos ofereciam controles bem atuais como: *checkbox*, *listbox*, *botões de pressão*, *botões de rádio*, *barras de rolagem*, *barras de menus*, etc, visíveis na tela do MS-DOS, mediante caracteres semi-gáficos, utilizando a tabela ASCII estendida. Na sua sintaxe existiam recursos fantásticos que a maioria das linguagens de hoje não possuem. Por exemplo, considere linguagens que criam variáveis estáticas como Pascal, ou dinâmicas como o Python: em nenhuma delas se consegue criar um identificador de variável de maneira automática, pois tem que ser utilizada a instrução **Var num: integer** (estaticamente em Pascal), ou inicializando com **num = valor** (dinamicamente em Python). Então, a questão é a seguinte: como criar nove variáveis inteiras identificadas como: *num1*, *num2*, *num3*, *num4*, *num5*, *num6*, *num7*, *num8* e *num9*, e sem declarar estaticamente ou iniciar dinamicamente essas variáveis, de maneira explícita!? Com o “velho” Clipper é possível fazer isto utilizando uma de suas mais poderosas armas: o operador macro **&**, como mostra o código da **figura 1**: a criação das nove variáveis citadas, praticamente do nada. Este é um recurso que em algumas situações pode ser necessário, e que não é oferecido por outras linguagens de programação. E com a tendência mundial de um mercado de programadores autônomos, em vez de grandes equipes nas empresas, é importante conhecer linguagens que ofereçam recursos extras; e o Clipper, assim como outras “velhas” linguagens, pode fazer a diferença na codificação do programa, ou em rotinas específicas. Afinal de contas, o mais interessado no programa, aquele que realmente manda no sistema é o USUÁRIO, e ele não está interessado na linguagem que o programa foi codificado; o que ele lhe interessa é que o sistema funcione ao seu gosto!

Portanto, o Clipper é um clássico; não velho! E foi ressuscitado, e revigorado, no padrão Harbour (ou xHarbour) em novo ambiente, e mais poderoso ainda: tal como um novo “*terminator*” reprogramado! E para aqueles que o consideram velho, cuidado: ele pode cumprir a ameaça: “*I’ll be back*”.

---

```

/*CriaVariaveis.prg
  Mostra como criar variáveis automaticamente
  Em Clipper 5.01
  Autor: Mário Leite
*/
Function Main()
j := 1
Do While (j<=9)
  var := "num" + str(j,1,0)
  &var. := j //atribuição do conteúdo de var
  j := j + 1
  Print &var.
EndDo
Return NIL //retorna e finaliza o programa

```

**Figura 1 - Criando variáveis automaticamente**