

Visual Basic for Applications (VBA)

Introdução

- VBA é "Visual Basic for Application". É uma linguagem de programação que permite os usuários a programarem macros para efetuar tarefas complexas dentro de uma aplicação. Com o VBA do Excel você pode desenvolver pequenos procedimentos (macros e/ou funções) que tornarão sua vida profissional mais fácil e lhe permitir fazer mais em menos tempo. Mas o VBA também é uma linguagem de programação muito poderosa com a qual você pode desenvolver dentro do Excel programas reais que efetuarão em uns poucos minutos tarefas muito complexas. Com o VBA para o Excel você pode desenvolver um programa que faça EXATAMENTE o que você precisa.

Origens e influências

- Foi inicialmente integrada com o Excel 5 em 1994 e a partir daí a sua expansão para outras aplicações foi gradual. Foi com a saída do Office 97 em 1997 que a Microsoft concretizou um dos seus grandes objetivos que era ter um ambiente de programação completamente integrado entre os seus quatro famosos produtos: Word, Excel, Access e PowerPoint. Atualmente, o VBA é já por si só um produto independente, que outras companhias podem adoptar e incorporar nas suas aplicações

Classificação

- Em suas primeiras versões, o Visual Basic não permitia acesso a bancos de dados, sendo portanto voltado apenas para iniciantes, mas devido ao sucesso entre as empresas — que faziam uso de componentes adicionais fabricados por terceiros para acesso a dados — a linguagem logo adotou tecnologias como DAO, RDO, e ADO, também da Microsoft, permitindo fácil acesso a bases de dados. Mais tarde foi adicionada também a possibilidade de criação de controles ActiveX, e, com a chegada do Visual Studio .NET, o Visual Basic — que era pseudo-orientada a objetos — tornou-se uma linguagem totalmente orientada a objetos (OO).

Avaliação em comparação com outras linguagens

- O VBA foi criado para resolver problemas e expandir as funcionalidades dos aplicativos do Microsoft Office. Então, para aprender VBA, é necessário ter um conhecimento básico do Microsoft Office (Word, Excel, PowerPoint ou Outlook).
- O VBA precisa de poucas linhas de código para resolver um problema de um aplicativo Office que, se fizesse por outra linguagem, levaria dezenas de linhas. E mais: o ambiente de desenvolvimento de VBA já vem integrado ao aplicativo que você quer automatizar, enquanto que outra solução necessitaria instalar um software extra no computador. Você pode ligar qualquer computador que possua Office e começar a desenvolver em VBA, mas em outras linguagens de programação, não.

- Uma grande vantagem do VBA é a velocidade em que se constroem programas: é muito rápido o processo de escrever, testar e executar. Quando escrevemos um programa e vamos executá-lo, normalmente temos que esperar um tempo em que a máquina virtual do ambiente de desenvolvimento converte seu código para linguagem de máquina (chamado de *tempo de compilação*). À medida que a linguagem de programação se torna mais complexa, esse tempo aumenta. No VBA, esse tempo é quase zero. Para fazer scripts e testes rápidos, ele é imbatível! Além disso, ele já está integrado na aplicação que você quer automatizar, e então não é preciso importar nem desenvolver códigos extras que apontem para o aplicativo desejado.

Exemplos de código representativos

- **Hello World**

```
Sub hello()  
  MsgBox "Hello World - Excel VBA here"  
End Sub
```

- **Somar dois números**

```
Dim a, b, c As Integer  
Call Soma(a, b, c)
```

```
Sub Soma(a As Integer, b As Integer, c As Integer)  
  c = a + b  
End Sub
```

- **Manipulando dados do excel**

```
Sub manipula()  
  Range("A1").Value = "Guru do Excel"  
End sub
```

Bibliografia

-
- http://www.dcc.fc.up.pt/~ricroc/aulas/0203/sap/pdf/vba_excel.pdf
- <http://ambienteoffice.com.br/vba/>
- <http://www.bertolo.pro.br/FinEst/SemanaContabeis2007/MacroExcel.pdf>
- <https://www.oficinadanet.com.br/artigo/excel/estudos-sobre-vba-introducao>
- <http://webserver.mohid.com/MTDP/downloads/Introdu%C3%A7%C3%A3o%C3%A0%20programa%C3%A7%C3%A3oemVBA.pdf>
- <http://www.officevb.com/2011/02/orientacao-objetos-no-vba-parte-i.html>