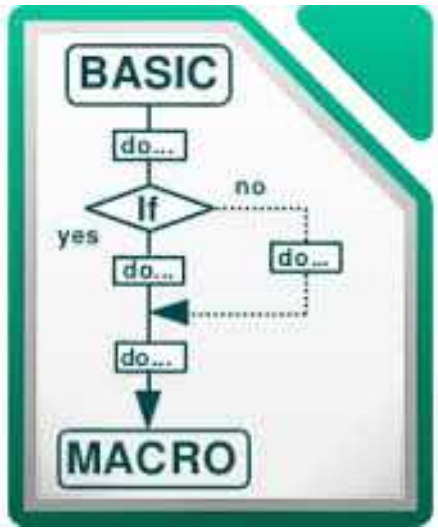
Ambiente Livre
Tecnologia

LibreOffice
The Document Foundation

Programação de Macros com LibreOffice Basic



Palestrante:
Marcio Junior Vieira

Marcio Junior Vieira

- 15 anos de experiência em informática, vivência em desenvolvimento e análise de sistemas de Gestão empresarial.
- Trabalhando com Software Livre desde 2000 com serviços de consultoria e treinamento.
- Graduado em Tecnologia em Informática(2004) e pós-graduado em Software Livre(2005) ambos pela UFPR.
- Palestrante em diversos Congressos relacionados a Software Livre tais como: CONISLI, SOLISC, FISL, LATINOWARE, SFD, JDBR, Pentaho Day.
- Fundador e atual CEO da Ambiente Livre.
- **Programador de Macros desde 2001**

Ambiente Livre
Tecnologia

Uma empresa de Software Livre e Software Aberto

LibreOffice
The Document Foundation

Sobre a Ambiente Livre

- Fundada em 2004 com foco de atuar em consultoria com software livre.
- 2009 ampliou sua soluções para atender ao mercado de gestão empresarial com software livre.
- Tem 14 soluções distintas para geração de negócios com software livre.

Soluções

- Consultoria.
- Desenvolvimento.
- Suporte.
- Treinamento.
- Transferência de Tecnologia.

Ambiente Livre
Tecnologia

Ecosistema

pentaho
Alfresco
SUGARCRM
LIFERAY
Enterprise. Open Source. For Life.
moodle
Joomla!™
...because open source matters
dotProject
Project Management Software
Drupal
OcoMon
Monitoramento e Gerência de Ocorrências
LPI
WordPress
LimeSurvey
Google SEO
OpenCerne
hadoop
LibreOffice
The Document Foundation
Iguana
Business Intelligence
for Open Source and Commercial Systems
MySQL
MariaDB
php
PostgreSQL
the world's most advanced open source database

LibreOffice
The Document Foundation

LibreOffice

- **Writer** - É o editor de textos.
- **Calc** - Planilha eletrônica com todos os recursos para calcular, analisar, resumir e apresentar seus dados em relatórios numéricos ou em gráficos.
- **Impress** – Criação de apresentações multimídia eficientes.
- **Draw** – Desenho vetoriais, produz desde simples diagramas até ilustrações com aparência 3D.
- **Math** - Editor de fórmulas. Extremamente útil para trabalhos científicos ou escolares.
- **Base** - Permite manipular bancos de dados.

Macros

- Uma macro é um programa escrito numa linguagem suportada pelo LibreOffice com a finalidade de automatizar tarefas. Atualmente, as linguagens suportadas são:

  - LibreOffice Basic;
  - JavaScript;
  - JavaBeans;
  - Java;
  - Python;

LibreOffice Basic

- Mantém as principais características do BASIC: sintaxe, tipos de dados, operadores, comandos, funções internas e organização geral do programa.
- Permite o acesso a uma grande quantidade de objetos, com seus métodos e propriedades, específicos do LibreOffice.
- IDE (Integrated Development Environment - Ambiente de Desenvolvimento Integrado) completo: edição de código fonte, verificação de erros, criação de diálogos e gerenciamento de bibliotecas.

Onde posso rodas as Macros

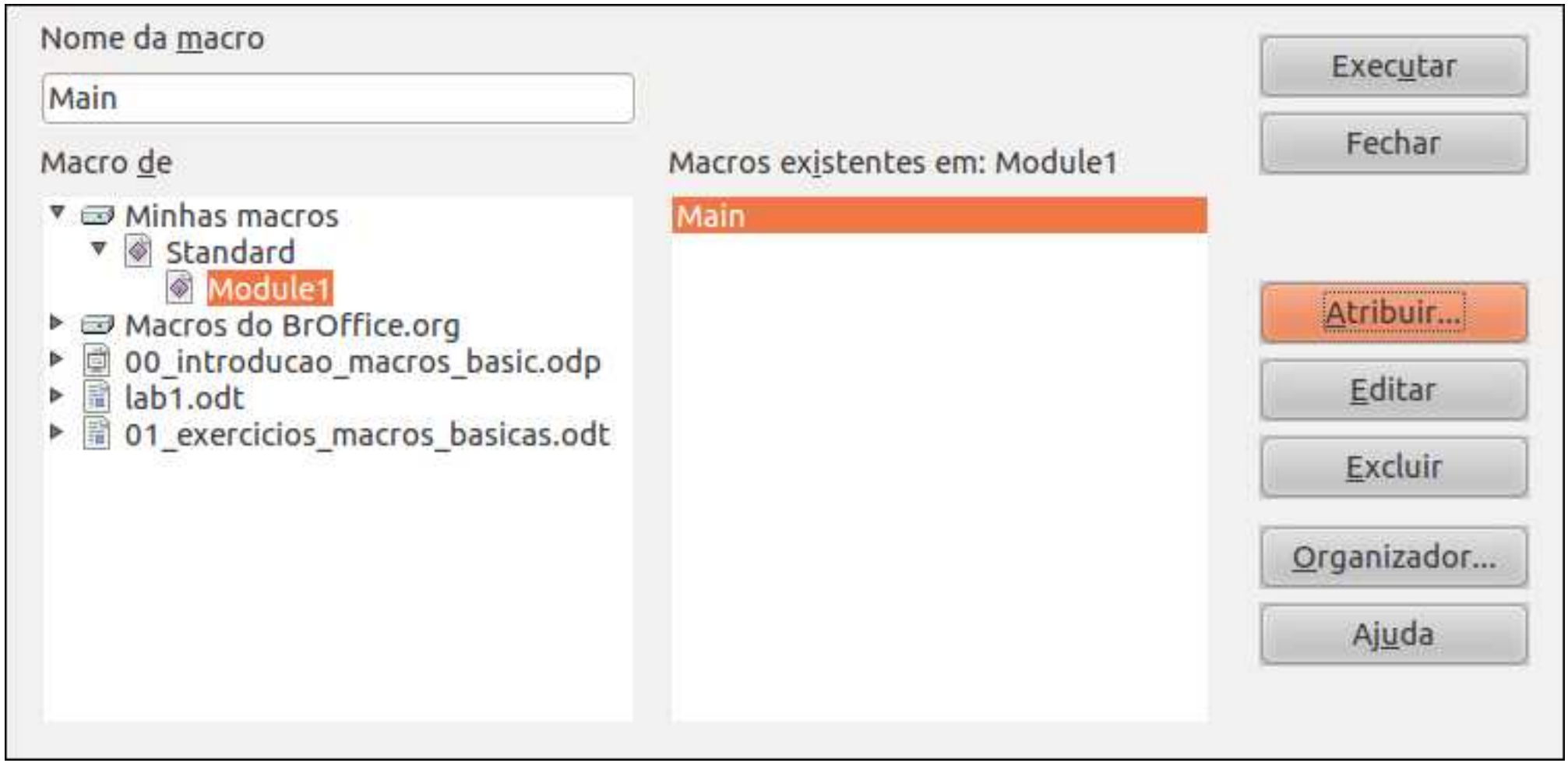
LibreOffice

Apache OpenOffice™

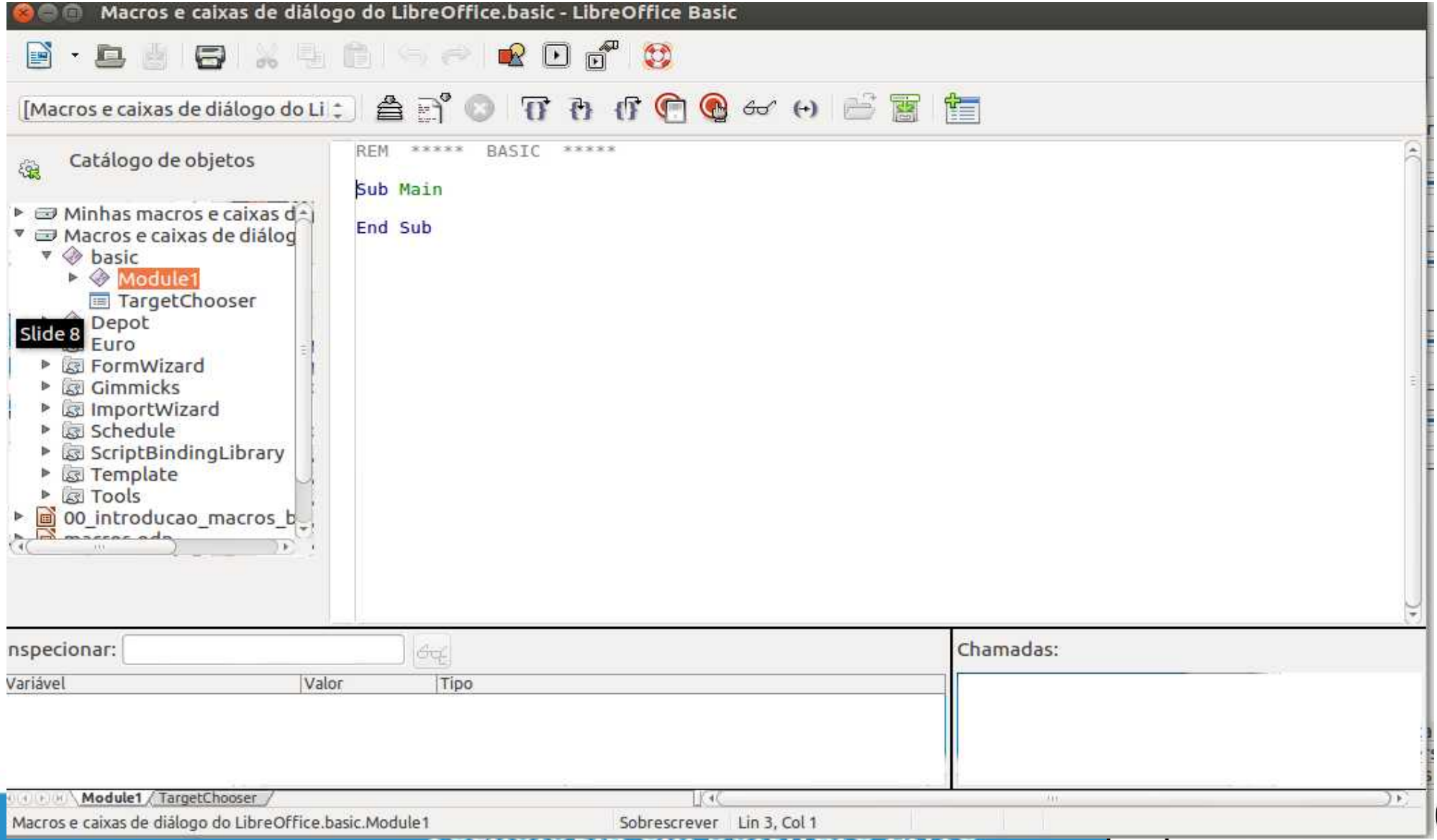
OpenOffice.org™

BROffice.org

Organização de Macros

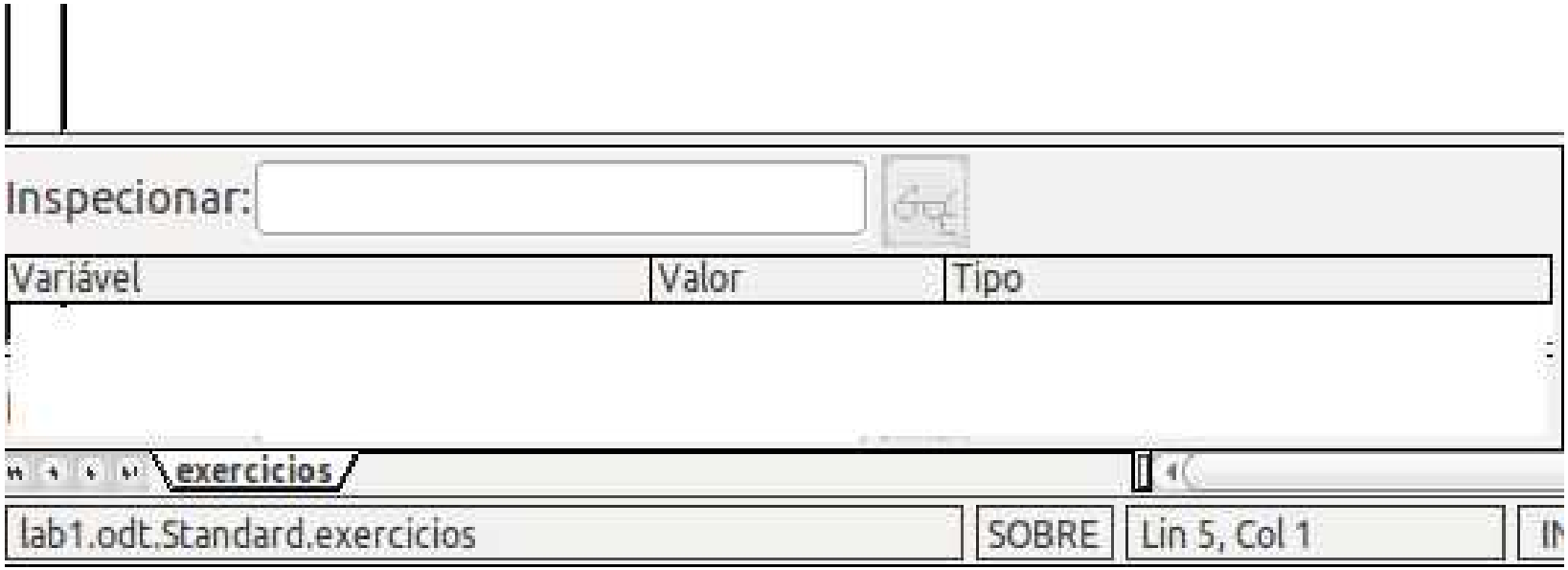


IDE Basic


Macros e caixas de diálogo do LibreOffice.basic - LibreOffice Basic
[Macros e caixas de diálogo do Li
Catálogo de objetos
Minhas macros e caixas d
Macros e caixas de diálog
basic
Module1
TargetChooser
Depot
Euro
FormWizard
Gimmicks
ImportWizard
Schedule
ScriptBindingLibrary
Template
Tools
00_introducao_macros_b
REM ***** BASIC *****
Sub Main
End Sub
nspecionar:
Variável
Valor
Tipo
Chamadas:
Module1
TargetChooser
Macros e caixas de diálogo do LibreOffice.basic.Module1
Sobrescrever
Lin 3, Col 1

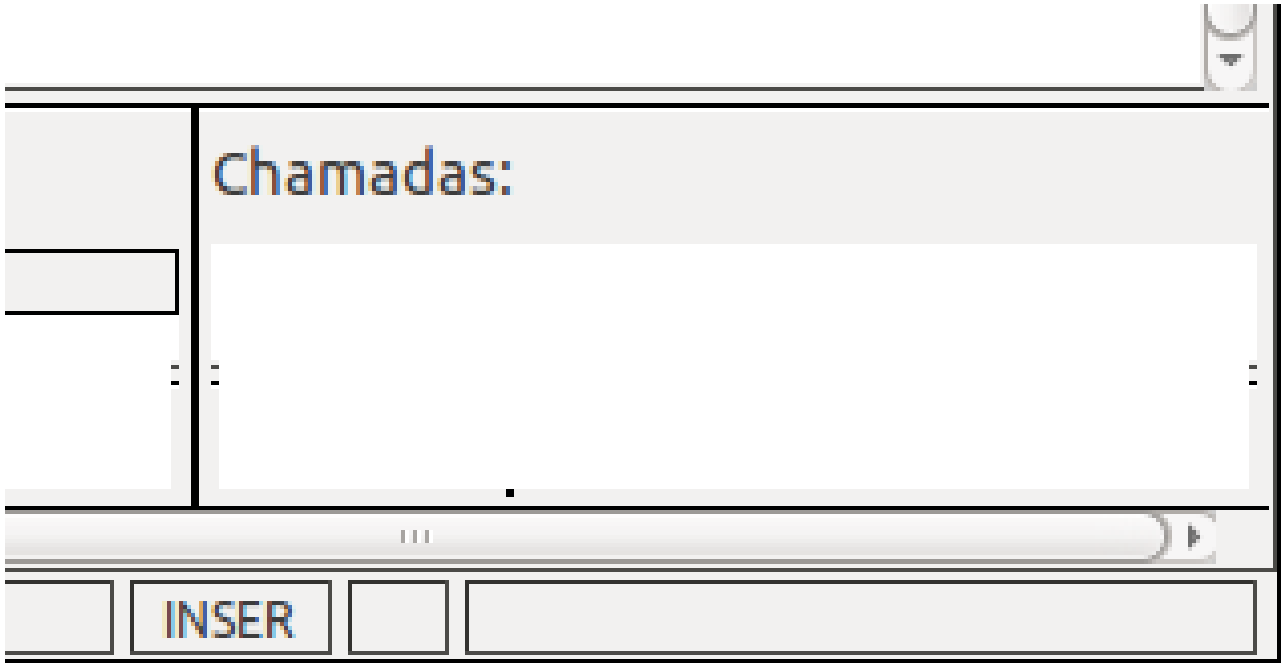
IDE Basic - Inspecionar

- Permite visualizar valores de variáveis das macros em tempo de execução.



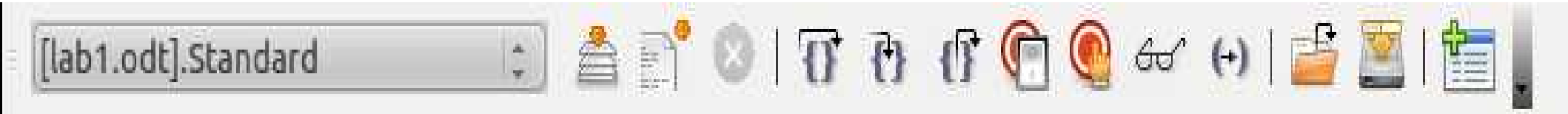
IDE Basic - Chamadas

- Mostra a chamada a procedimentos da macro em tempo de execução.

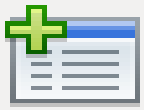


IDE Basic

-  **Compilar** – Compila o código-fonte Basic.
-  **Executar o Basic** – Executa uma macro até um pondo te interrupção.
-  **Parar Macro** - termina a execução de uma macro. Somente será ativado quando a execução da macro iniciar.
-  **Passar ao Seguinte** – Executa a macro linha a linha – passo a passo



IDE Basic

-  **Ativar/desativar pontos de Interrupção** - define um ponto de interrupção.
-  **Gerenciar Pontos de Interrupção**
-  **Inserir código-fonte BASIC** – Abre e insere o conteúdo de um arquivo Basic (*.bas)
-  **Salvar BASIC** – Salva o modulo com o código-fonte em arquivo Basic (*.bas)
-  **Importar caixa de dialogo**

Características

- Linguagem Interpretada
- Case-insensitive (*1)

Ambiente Livre
Tecnologia

SubRotinas

- Uma macro executa uma **subrotina**;
- **Subrotinas** são blocos de instrução;
- Uma **subrotina** começa com Sub e termina com End Sub.
- Uma subrotina pode receber parâmetros.

```
REM  *****  BASIC  *****

Sub Main

End Sub
```

Linhas de Programação

```
• LongExpression = (Expression1 * Expression2) + _
• (Expression3 * Expression4) + _
• (Expression5 * Expression6) + _
• (Expression7 * Expression8)
```

Linhas de Programação

```
a = 1
a = a + 1
a = a + 1
```

- **Ou**

```
a = 1 : a = a + 1 : a = a + 1
```

Comentários

```
' este é um comentário  da linha
REM este é um comentário usando a chave REM
<expressão>   ' este é um comentário que por
              ' ser extenso usa três linhas para
              ' ser escrito.
```

Marcadores

- Podem ser usados caracteres LATIN , números e underscores (_)
- Não pode ser usados caracteres especiais
- O tamanho máximo é 255 Caracteres
- Não é case-sensitive
- Não é permitida acentuação

Marcadores

- Surname ' OK
- Surname5 ' OK
- First Name ' Incorreto espaço não é permitido
- DéjàVu ' acentuação não permitida
- 5Surnames ' incorreto primeiro caracter não pode ser um numero
- First,Name ' incorreta

Marcadores

```
Dim [First Name] As String
Dim [DéjàVu] As Integer
[First Name] = "Andrew"
[DéjàVu] = 2
```

Tipos de Variáveis

- numéricas,
- lógicas,
- strings,
- Datas e objetos.

Variáveis Implícitas

- a = b + c
- **Option Explicit** – Adicionado ao inicio das macros obriga que as variáveis sejam definidas.

Implícitas

- MyVar = "Hello World" ' string
- MyVar = 1 ' numero
- MyVar = 1.0 ' float / decimal
- MyVar = True 'Booleana

Varíaveis Dimencionais

- É possível definir novos tipos, que combinam tipos existentes
- **Dim** é a forma de declarar uma variável; Serve para declarar vetores, matrizes ou arrays.

```
Dim x As tipo-da-variável

Dim m(3,5) As tipo-da-variável
```

- O que é declarado é o valor **final** da dimensão, que começa com zero. Então, uma declaração do tipo :
  `Dim m(3,5) As Integer`
  Declara uma matriz de (3 + 1) x (5 + 1) inteiros.

Variáveis Númericas

- **Integer**;
- **Long Integer**;
- **Decimal**;
- **Single** (ponto flutuante, precisão simples);
- **Double** (ponto flutuante, precisão dupla);
- **Currency** (para guardar valores monetários com alta precisão)

SIGIL

- Variáveis também podem ser declaradas usando um sigil após o Dim.

```
Dim x$   ' x é uma String

Dim y%   ' y é um Inteiro

Dim z#   ' z é um Double
```

Variáveis

- String

```
Dim sVar as String
```

- **`Dim a,b,c,d as String`**
- Boolean

```
Dim bVar as Boolean
```

- Date

```
Dim dVar as Date
```

Simples Array

- Dim MyArray(3)

Global – Privada - Constantes

- **Global**
  Global A As Integer
- **Private**

  Private C As Integer
- **Constante**

```
Const A = 10

Const B As Double = 10
```

Operadores Matemáticos

- **+** - Adição – Números , datas e strings
- **&** - liga strings
- **-** - Subtração – números , datas
- ***** - Multiplicação de números.
- **/** - Divisão de números
- **** - Divisão de números com um resultado inteiro (arredondado)
- **MOD** - operação módulo (cálculo do resto de uma divisão) Ex.: MyVar = 3 MOD 2

Operadores de Comparação

- = Igualdade de números, datas e strings
- <> Desigualdade de números, datas e strings
- > Maior que
- >= Maior igual
- < Menor que
- <= Menor igual

- LibreBasic não suporta comparador Like - VBA

If...Then...Else

```
If A > 3 Then
  B = 2
Else
  B = 0
End If
```

Select...Case

```
Select Case DayOfWeek
  Case 1:
    NameOfWeekday = "Domingo"
  Case 2:
    NameOfWeekday = "Segunda"
  Case 3:
    NameOfWeekday = "Terça"
End Select
```

Repetição - For...Next

```
Dim I
For I = 1 To 10
   ' ... Código para repetição
Next I
```

- Incrementação automática!

For Each

```
Const d1 = 2
Const d2 = 3
Const d3 = 2

Dim i
Dim a(d1, d2, d3)
For Each i In a()
     '... passará 36 vezes neste loop
Next i
```

Do Until / Do loop

```
Do Until A > 10
    ' ... loop body
Loop

Do
    ' ... loop body
Loop While A > 10
```

Object

- Usado para o tratamento de estruturas complexas.
- Ao se rodar o BASIC em um documento do Writer ou Calc, o documento em si é um objeto, associado à variável **ThisComponent**.
- Em um documento do Calc, a totalidade das planilhas é o objeto **ThisComponent.Sheets**.
- Para acessar, por exemplo, uma planilha de nome "Alunos", pode-se declarar:

```
Dim alunos As Object
alunos = ThisComponent.Sheets.GetByName("Alunos")
```

Estruturas

- Forma complexa de definir variáveis
  Type ... End Type

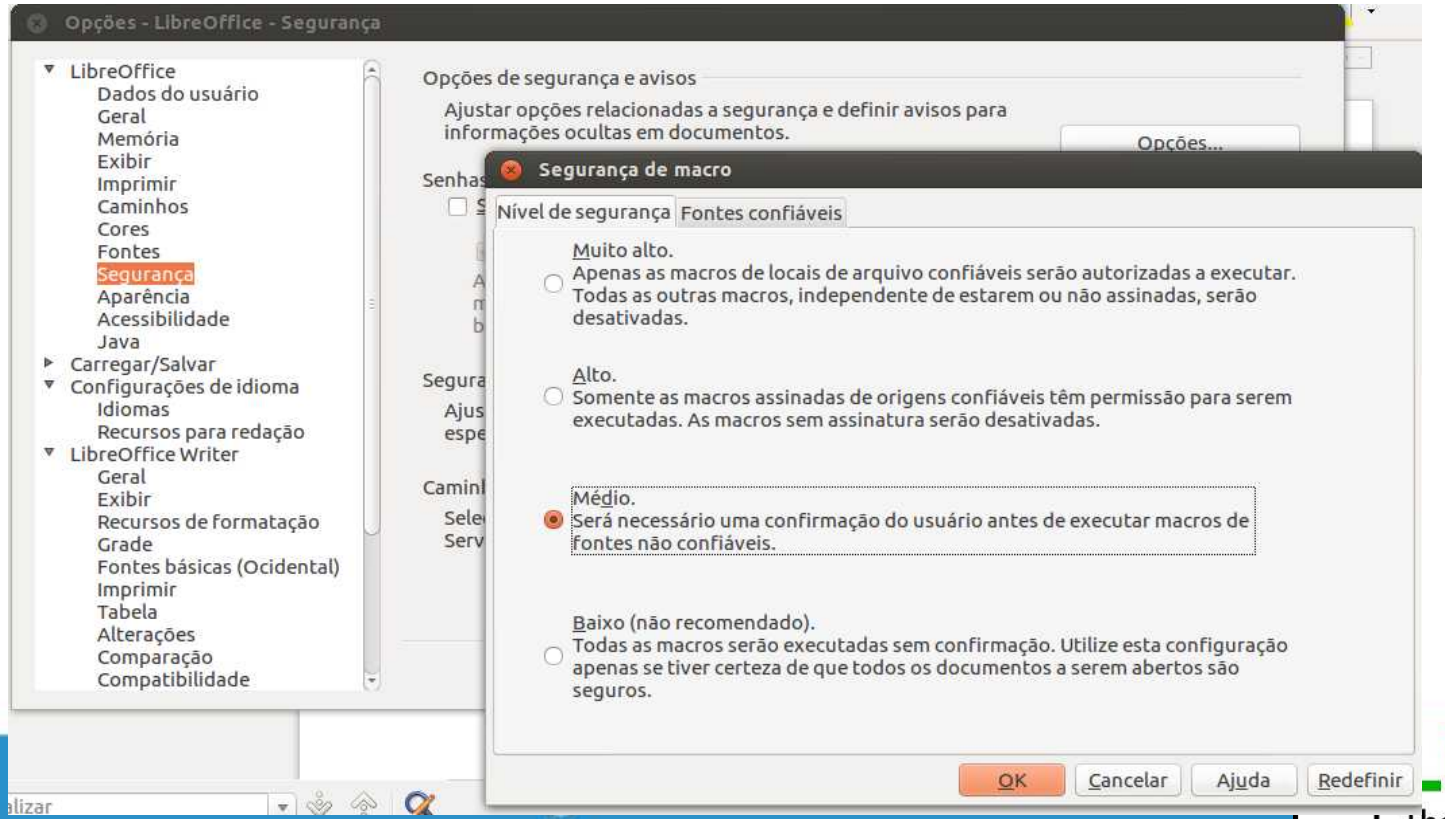
```
Type Timedefutebol
     Nome as String
     V as Integer
     E as Integer
     D as Integer
     GP as Integer
     GC as Integer
  End Type
```

Estruturas

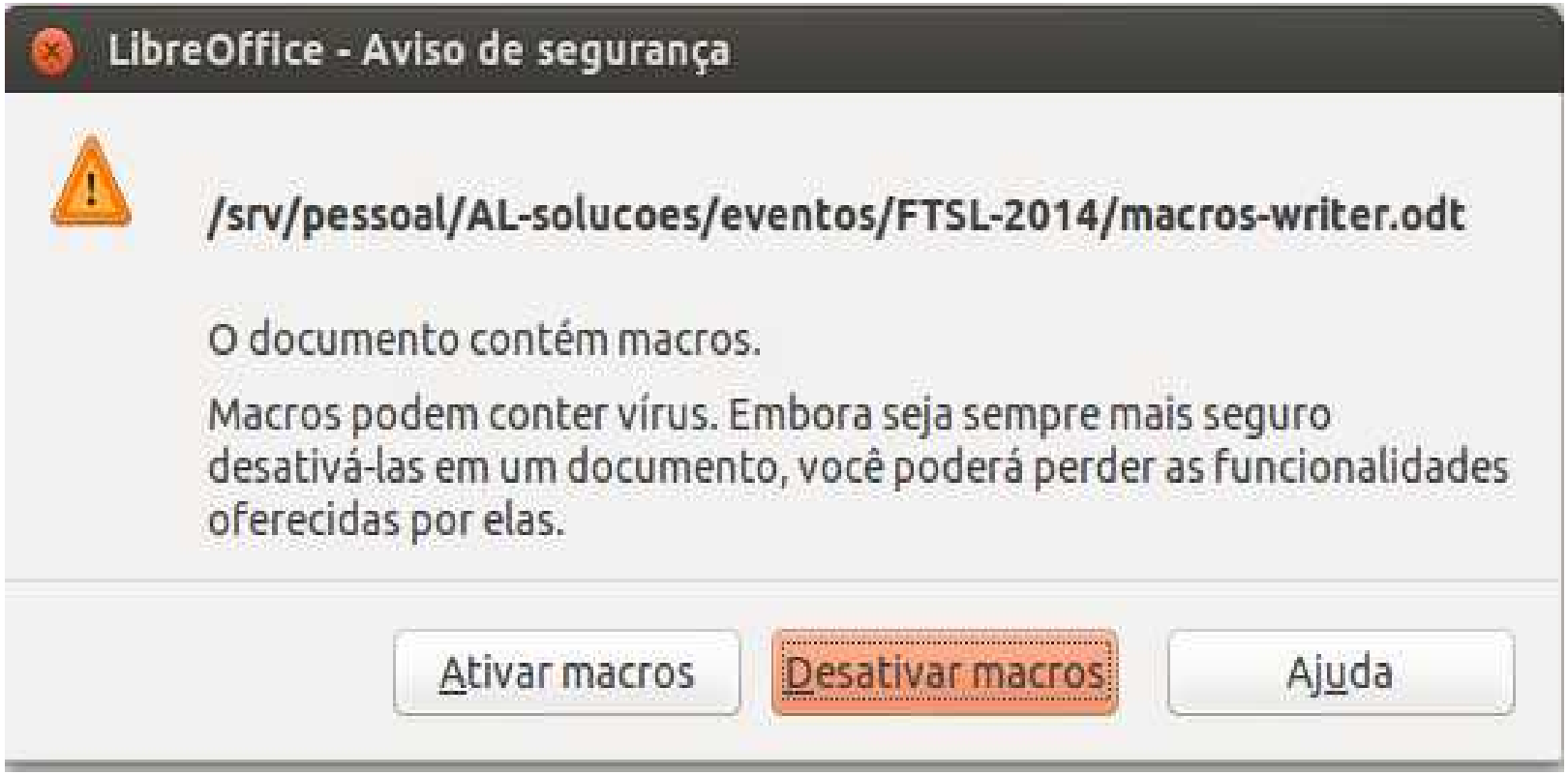
```
Dim MeuCampeonato(10) as New Timedefutebol
  MeuCampeonato(1).Nome = "BROFFICE F.C."
  MeuCampeonato(1).V = 54
  MeuCampeonato(1).E = 1
  MeuCampeonato(1).D = 0
  MeuCampeonato(1).GP = 283
  MeuCampeonato(1).GC = 2
```

Segurança Execução de Macros

- Por padrão a execução de macros encontra-se “protegido” na instalação do LibreOffice
- Para ativar acesse o Menu **Ferramentas - Opções –LibreOffice – Segurança – (Segurança de Macros...)**



Mensagem de alerta - Segurança



LibreOffice Basic

- Procedimentos
- Funções
- Recursos de recursividade
- Conversões de Tipo
- Manipulação e Formatação de String
- Acesso ao sistema de arquivos/diretório
- Comandos de desvio
-

API LibreOffice

- **Serviços.**
- **Objetos.**
- **Interfaces.**
- **Métodos**
- **Propriedades**

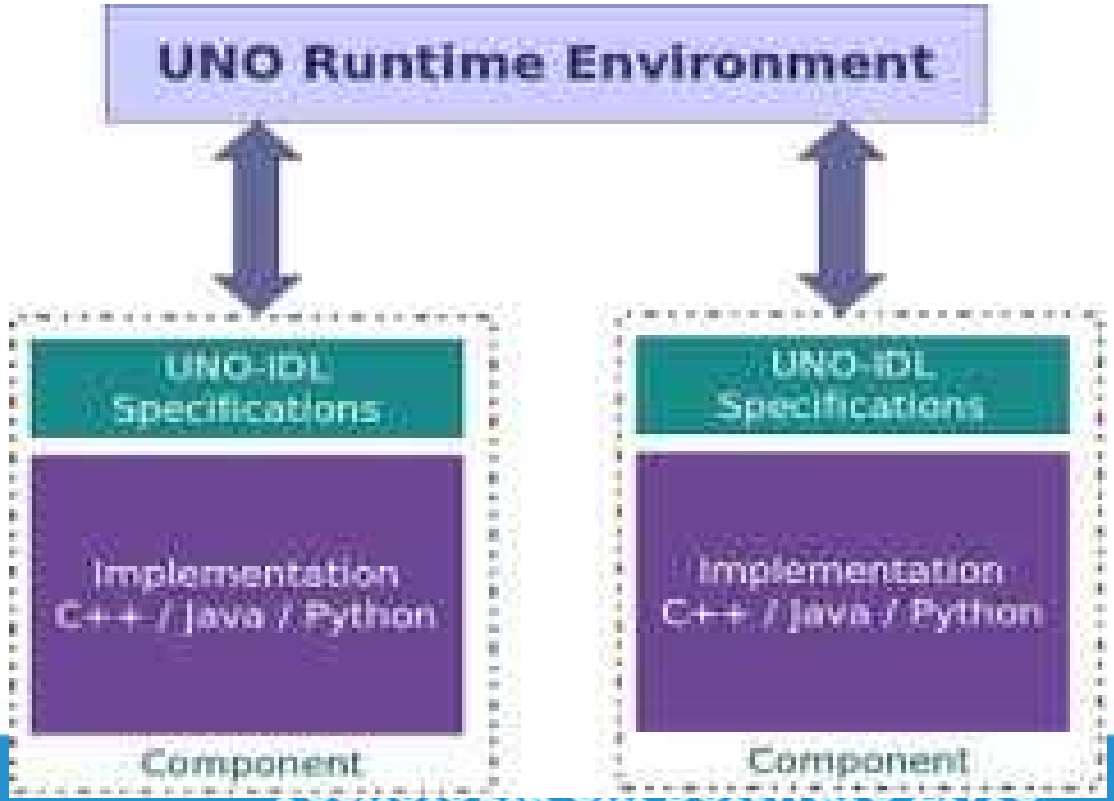
API LibreOffice

- A chave para a criação de programas que usam a API do LibreOffice são os **serviços**.
- Um **serviço** é uma especificação de um objeto, que engloba um conjunto de **interfaces** e propriedades.
- Uma **interface** é uma coleção de métodos.
- Uma **propriedade** é um valor que determina uma característica de um serviço e é formada por um **nome** e um **valor.**

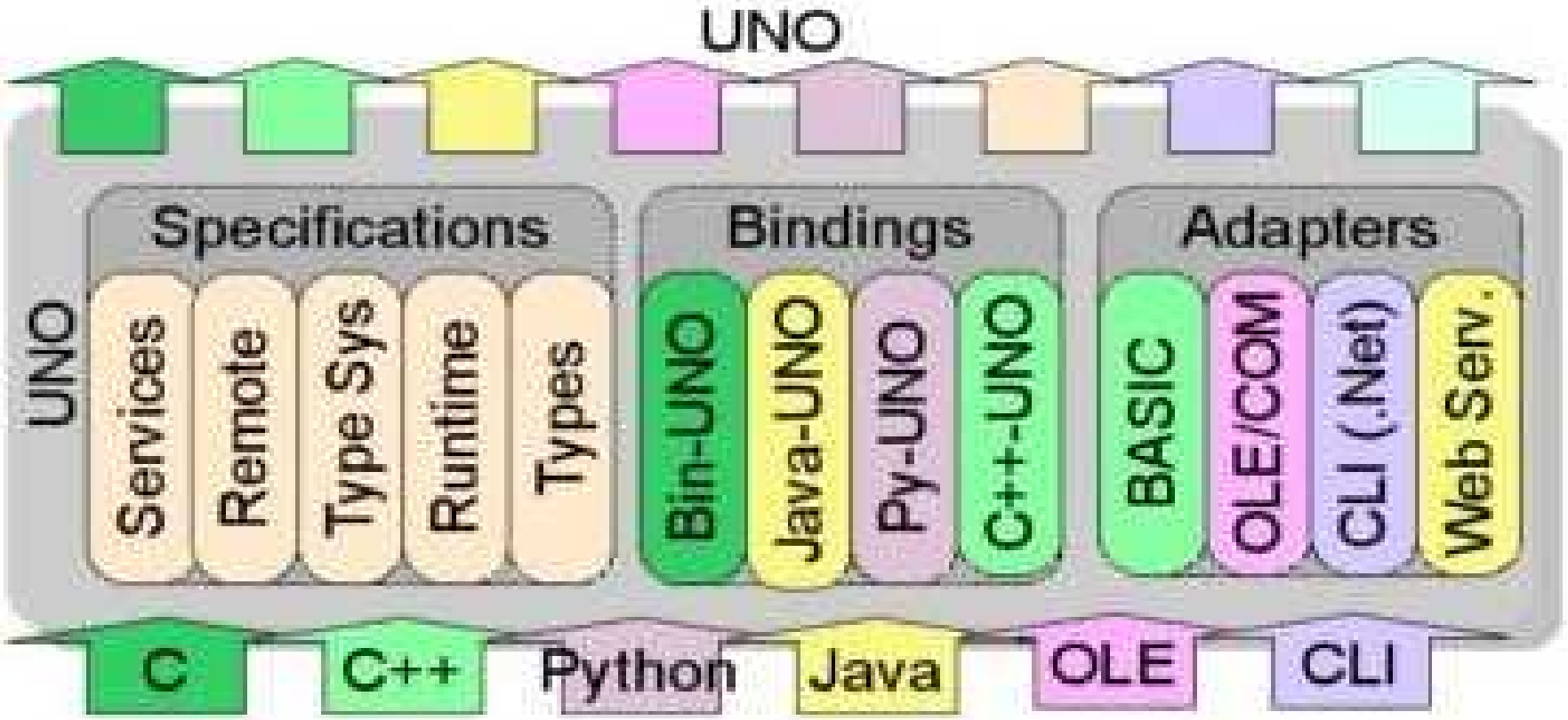
Universal Network Objects (UNO)

- A variável de objeto deve ser criada e inicializado para que ele possa ser usado. CreateUnoService

```
Dim Obj As Object
Obj = createUnoService("com.sun.star.frame.Desktop")
```



UNO


UNO
UNO
Specifications
Services
Remote
Type Sys
Runtime
Types
Bindings
Bin-UNO
Java-UNO
Py-UNO
C++-UNO
Adapters
BASIC
OLE/COM
CLI (.Net)
Web Serv.
C
C++
Python
Java
OLE
CLI

VBA X LOBasic

- A estrutura de um objeto no VBA é definida pela classe à qual ele pertence
- Em LOBasic a estrutura é definida através dos serviços que ele suporta.
- Um objeto VBA é sempre atribuído a exatamente uma única classe.
- O objeto LOBasic objeto pode ter apoio de vários serviços.

Termologia VBA X LOBasic

- VBA OOoBasic
- **O objeto documento**
- Workbook - SpreadsheetDocument
- **Páginas Individuais**
- Worksheets - Sheet

VBA to LOBasic

- http://www.business-spreadsheets.com/vba2oo.asp
-

Propriedades

- As propriedades são definidas por meio de uma atribuição simples:

```
Document.Title = "{{OOo}} Treinamento OOoBasic"
Document.Filename = "treina_ooobasic.odt"
```

- A propriedade, assim como uma variável normal, tem um **tipo** que define que valores ela pode registrar.
- As propriedades **Filename** e **Title** são do tipo string.

Métodos

- Os métodos podem ser entendidos como funções que se relacionam diretamente entre objeto que fazem chamadas uns aos outros.
- O Objeto documento poderia, por exemplo, fornecem um método Save, que pode ser chamado como segue:

```
Document.Save()
```

Módulos

- com.sun.star.awt - interface do usuário
- com.sun.star.beans - acesso a propriedades
- com.sun.star.container - coleções e recipientes
- com.sun.star.document - documentos do office
- com.sun.star.drawing - desenho
- com.sun.star.text - documentos texto
- com.sun.star.sdb - banco de dados
- com.sun.star.sheet - planilhas
- com.sun.star.util - utilidade diversa

Trabalhando com planilhas

- **ThisComponent** - documento em que a macro está sendo executada
- **<Objeto>.Sheets.(n)** - ' recebe as pastas do documento
- **<Objeto>**.GetCellByPosition(Coluna, Linha)
- **<objetocelula>**.Value 'valor na célula

Criando Planilhas

```
Sheet = Doc.createInstance("com.sun.star.sheet.Spreadsheet")

Doc.Sheets.insertByName("PlanilhaN", Sheet)
```

Linhas e Colunas nas Planilhas

- Sheet = Doc.Sheets(0)
- FirstCol = Sheet.Columns(0)
- FirstRow = Sheet.Rows(0)

- **Nota** - As listas de linhas e colunas podem ser acessados através de um índice em LOBasic. Diferente do VBA, a primeira coluna tem índice 0 e não o Índice 1.

Células

```
Dim Doc As Object
Dim Sheet As Object
Dim Cell As Object
Doc = Thiscomponent
Sheet = Doc.Sheets(0)
Cell = Sheet.getCellByPosition(0, 0)
Cell.String = "Test"
Cell.Value = 100
Cell.Formula = "=A1+A2"
```

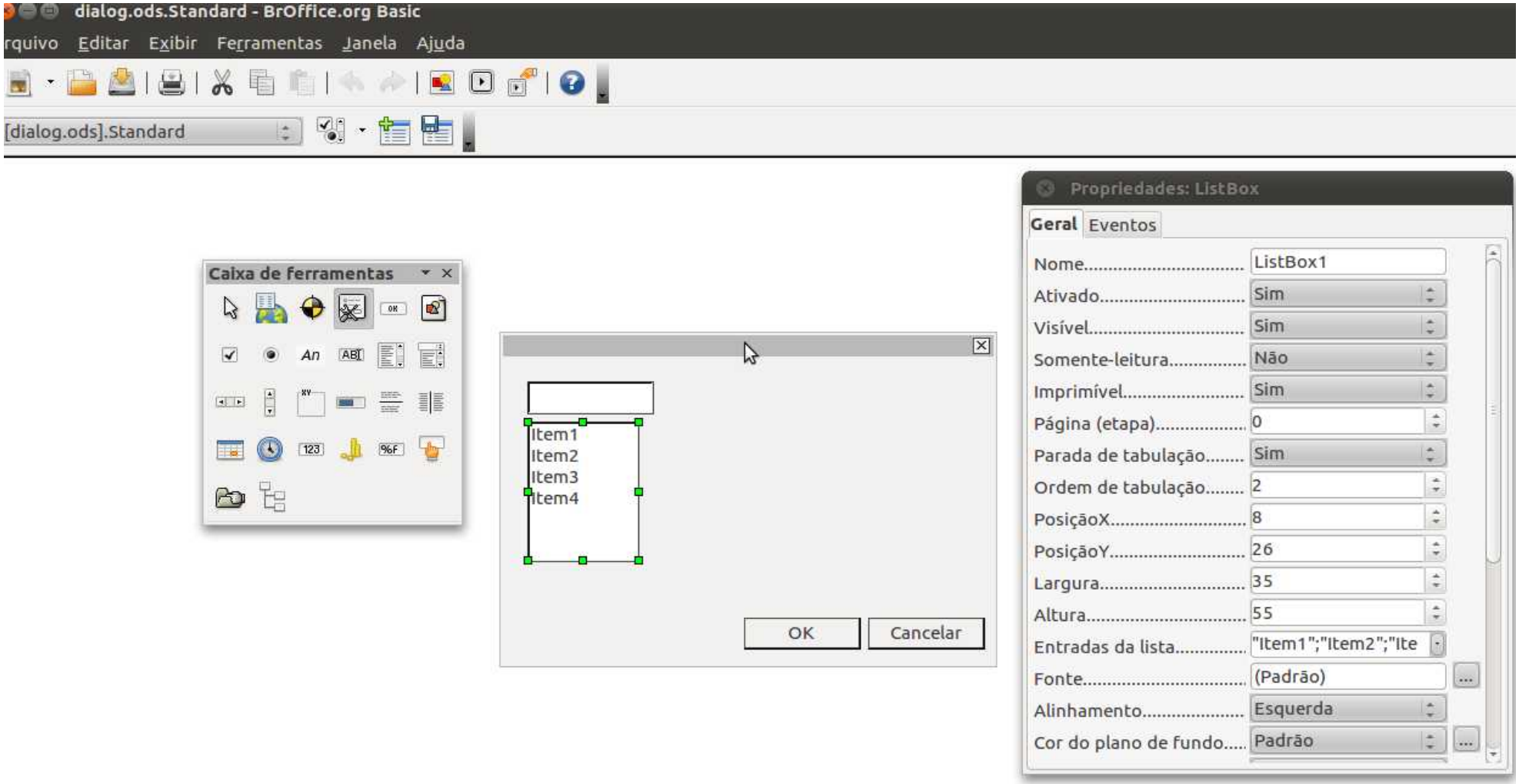
Gráficos

```
Dim Rect As New com.sun.star.awt.Rectangle
   Dim RangeAddress(0) As New com.sun.star.table.CellRangeAddress
   Doc = ThisComponent
   Charts = Doc.Sheets(0).Charts
   Rect.X = 8000
   Rect.Y = 1000
   Rect.Width = 10000
   Rect.Height = 7000
   RangeAddress(0).Sheet = 0
   RangeAddress(0).StartColumn = 0
   RangeAddress(0).StartRow = 0
   RangeAddress(0).EndColumn = 2
   RangeAddress(0).EndRow = 12
   Charts.addNewByName("MeuGrafico", Rect, RangeAddress(), True,
True)
```

Shell

- Shell(Pathname, Windowstyle, Param)
- Windowstyle
  0 – Programa recebe o foco e inicia janela escondida
  1 – programa recebe o foco e inicia janela normal
- Param: Parametro enviado ao programa externo.
- Paramâtros opcionais
- Exemplo
  Shell (“calc”)

IDE - Dialog


dialog.ods.Standard - BrOffice.org Basic
rquivo Editar Exibir Ferramentas Janela Ajuda
[dialog.ods].Standard
Caixa de ferramentas
Item1
Item2
Item3
Item4
OK
Cancelar
Propriedades: ListBox
Geral
Eventos
Nome
ListBox1
Ativado
Sim
Visível
Sim
Somente-leitura
Não
Imprimível
Sim
Página (etapa)
0
Parada de tabulação
Sim
Ordem de tabulação
2
PosiçãoX
8
PosiçãoY
26
Largura
35
Altura
55
Entradas da lista
"Item1";"Item2";"Ite
Fonte
(Padrão)
Alinhamento
Esquerda
Cor do plano de fundo
Padrão

Banco da Dados - Base

- Connection = DataSource.GetConnection("usuariodobanco", "senhadobanco")
- ...
- Statement = Connection.createStatement()
- ResultSet = Statement.executeQuery(" COMANDO SQL DESEJADO") '

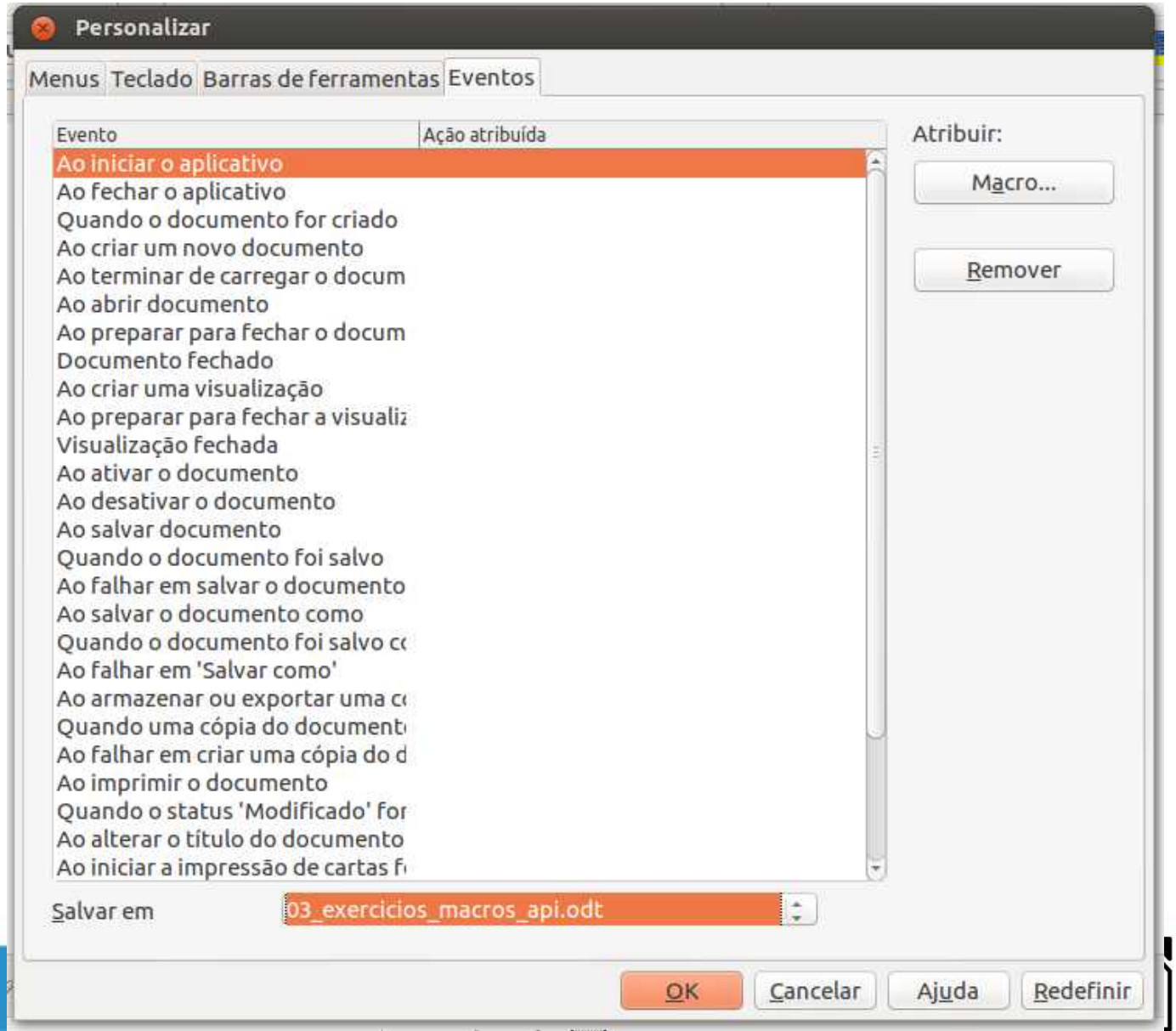
ResultSet - Navegando nos registro

```
If Not IsNull(ResultSet) Then
While ResultSet.next
   MsgBox ResultSet.getString(1)
Wend
```

- next() – próximo registro de dados.
- previous() – registro de dados anterior.
- first() – primeiro registro de dados.
- last() – último registro de dados.
- beforeFirst() – registro de dados anterior ao primeiro.
- afterLast() – próximo registro de dados após o último.

Ambiente Livre
Tecnologia

Eventos



LibreOffice
The Document Foundation

Gravação de Macros

- Permite gravar ações no LibreOffice em Macros
- Dever ser ativados recursos experimentais
- Tem algumas limitações.

Referências

- LibreOffice 3 Basic Guide

Contatos

- e-mail:
- marcio @ ambientelivre.com.br
- http://twitter.com/ambientelivre
- @ambientelivre
- @marciojvieira
- Blog

  blogs.ambientelivre.com.br/marcio