

designed  
for excellence



# Tutoriais de Extensibilidade

ERP PRIMAVERA V8.10

Versão 1.0  
março 2013

# Índice

<b>Índice.....</b>	<b>2</b>
<b>Introdução .....</b>	<b>3</b>
<b>Exemplos de VBA.....</b>	<b>4</b>
<b>Área de Logística e Tesouraria .....</b>	<b>4</b>
Atribuir ao PVP o valor do PCM atual .....	4
Apresentar informação e validações específicas sobre Artigos .....	6
Duplicação de Documentos de Compras .....	14
Agendar o Recálculo de Stocks para efetuar periodicamente.....	31
Criação de um documento de liquidação .....	36
Criação de um documento de transferência de conta/estado.....	40
Criação de um documento de transferência bancária .....	44
Criação de um documento de pagamento de cartão de crédito.....	45
Criação de um documento de aplicação financeira.....	46
Criação de um documento de capitalização de aplicação financeira .....	47
Criação de um novo pendente.....	48
Inserção de um pendente de juros de mora .....	50
Restrição às liquidações de uma determinada entidade a um determinado utilizador .....	54
Preenchimento de um campo de utilizador nas liquidações .....	56
Restrição à edição de novos pendentes para uma determinada entidade a um determinado utilizador .....	57
Adicionar uma linha fixa aos novos pendentes.....	58
Informar o crédito disponível na inserção de documentos.....	61
Limitar o valor da transferência bancária .....	63
Adicionar documento de despesas bancárias sempre que se efetua uma transferência.....	63
<b>Área de Contabilidade.....</b>	<b>66</b>
Criação de um lançamento na contabilidade com reflexão para centros de custo .....	66

## Introdução

A Extensibilidade dos módulos é um conceito central no ERP PRIMAVERA na medida em que permite complementar a solução *standard* com funcionalidades adicionais e intervir nas funcionalidades nucleares de cada módulo para responder às necessidades específicas de cada negócio ou cliente em particular.

Este manual tem como propósito documentar – na perspetiva dos Parceiros PRIMAVERA – as ferramentas de Extensibilidade mais importantes do ERP PRIMAVERA, apresentando exemplos práticos da sua aplicação e as vantagens competitivas que poderão proporcionar.

De modo a consultar a versão mais atualizada deste manual, sugere-se a sua consulta online na PRIMAVERA KnowledgeBase.

## Exemplos de VBA

A implementação dos exemplos VBA que se seguem pretende:

- Mostrar a facilidade de utilização do VBA;
- Sublinhar as vantagens advindas do seu uso.

Todos os exemplos apresentados foram desenvolvidos numa Base de Dados de Demonstração.

### Área de Logística e Tesouraria

#### Atribuir ao PVP o valor do PCM atual

Este exemplo pretende, antes de gravar um Documento de Venda, efetuar a alteração do Preço Unitário, supostamente preenchido com o PVP para o respetivo Preço de Custo Médio (PCM), para cada Artigo presente no Documento. Para tal necessitamos incluir código VBA no evento "AntesDeGravar()" do Editor de Vendas

Para executar o exemplo abaixo devem-se verificar as seguintes condições:

1. Existir a seguinte referência no Projeto VBA;
  - PRIMAVERA EXECUTIVE 8.00
  - PRIMAVERA StdBE 8.00.
2. Existir um Campo de utilizador "CDU\_PVPPorPCM", do tipo "BIT", associado aos Documentos de Venda (tabela [DocumentosVenda]).

Apresenta-se de seguida o código que permite implementar a funcionalidade descrita.

```
Private Sub EditorVendas_AntesDeGravar(Cancel As Boolean)
' O código abaixo permite alterar o Preço Unitário (PVP) dos Artigos presentes num
' determinado Documento de Venda pelo valor respetivo do Preço de Custo Médio
' (PCM). Se o Documento em causa estiver configurado para substituir o PVP pelo
' PCM então, para cada linha que referencie um artigo substitui-se o Preço
' unitário do Artigo (PVP) pelo valor mais recente do PCM.

Dim blnPVPorPCM As Boolean
Dim dblPCM As Double
Dim lngI As Long
Dim strSQL As String
Dim vntPCMedio As Variant
Dim objLista As StdBELista
```

```

On Error GoTo ERRO

blnPVPorPCM = False
blnUsaPCM = False

'Verificar se este Documento está configurado para utilizar o PCM em vez do PVP
blnPVPorPCM =
Aplicacao.BSO.Comercial.TabVendas.DaValorAtributo(Me.DocumentoVenda.TipoDoc,
"CDU_PVPPorPCM")

If blnPVPorPCM Then
    'Se é um documento configurado para substituir o PVP pelo PCM...

    For lngI = 1 To Me.DocumentoVenda.Linhas.NumItens
        'Para todas as linhas do Documento a ser gravado...

        If (Me.DocumentoVenda.Linhas(lngI).Artigo <> vbNullString) Then
            'Se a linha refere um Artigo

            'Vai procurar o Preço de Custo Médio do Artigo em causa
            strSQL = vbNullString
            strSQL = strSQL & "SELECT PCM FROM LinhasStk "
            strSQL = strSQL & "WHERE (Artigo = '" &
Me.DocumentoVenda.Linhas(lngI).Artigo & "'" ) "
            strSQL = strSQL & "AND (EntradaSaida = 'E' OR EntradaSaida = 'I') "
            strSQL = strSQL & "AND (Data <= '" &
Format(Me.DocumentoVenda.Linhas(lngI).DataStock, "yyyy-mm-dd") & "'" ) "
            strSQL = strSQL & "ORDER BY Data DESC, EntradaSaida DESC"
            Set objLista = Aplicacao.BSO.Consulta(strSQL)

            If Not (objLista Is Nothing) Then

                If Not (objLista.NoInicio And objLista.NoFim) Then 'existem
registros
                    dblPCM = PlataformaPRIMAVERA.Utills.FDbI#(objLista("PCM"))
                Else
                    'não tem registos por isso lê diretamente da ficha de artigo
                    vntPCMedio = Aplicacao.BSO.Comercial.Artigos.DaValorAtributo(
-
Me.DocumentoVenda.Linhas(lngI).Artigo, "PCMedio")
                    dblPCM = PlataformaPRIMAVERA.Utills.FDbI#(vntPCMedio)
                End If
                Set objLista = Nothing

            'Altera o preço unitário para o PCM e o Iva para zero
    
```

```
Me.DocumentoVenda.Linhas(lngI).PrecUnit = dblPCM
Me.DocumentoVenda.Linhas(lngI).CodIva = 0
Me.DocumentoVenda.Linhas(lngI).TaxaIva = 0

    End If

End If

Next lngI

End If

Exit Sub

ERRO:
'Antes de apresentar o erro limpa o(s) objeto(s) da memória
If Not (objLista Is Nothing) Then Set objLista = Nothing
MsgBox "Ocorreu o erro abaixo: " & vbCrLf & Err.Number & vbCrLf &
Err.Description, vbCritical, Aplicacao.Nome
End Sub
```

### Apresentar informação e validações específicas sobre Artigos

Com o exemplo seguinte pretende-se realizar 3 operações distintas:

- 1) Garantir que determinados Vendedores não vendem determinados Artigos a determinados Clientes;
- 2) Configurar um Documento de Venda para utilizar um armazém por defeito para os seus Artigos;
- 3) Apresentar informação sobre as Unidade Base dos Artigos e sua relação com a respetiva Unidade de Venda.

Como estas três ações são efetuadas para cada Artigo iremos fazer uso do evento "ArtigoIdentificado()" do Editor de Vendas.

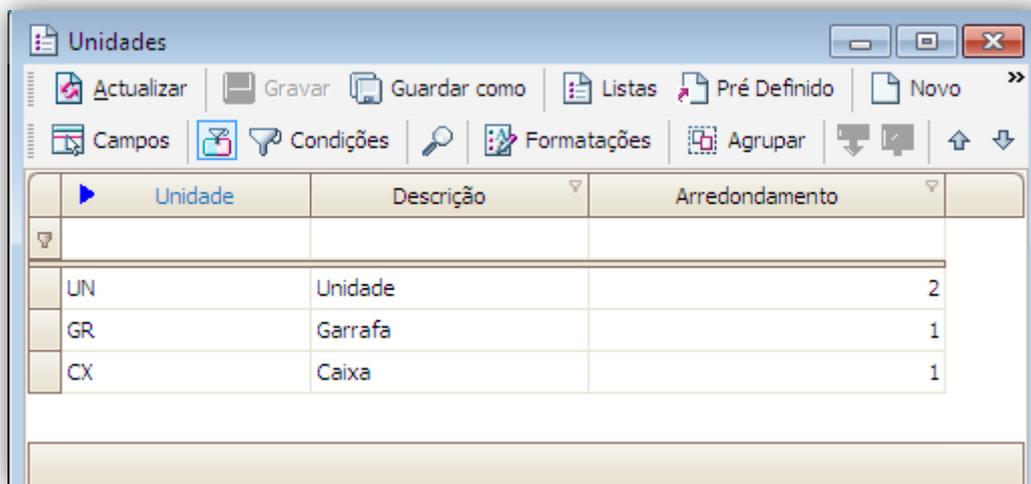
Para executar o exemplo abaixo devem-se verificar as seguintes condições:

1. Existir as seguintes referências no Projeto VBA;
  - PRIMAVERA EXECUTIVE 8.00;
  - PRIMAVERA StdBE 8.00.

2. Existir um Campo de utilizador “CDU\_Armazem” associado aos Documentos de Vendas (tabela [DocumentosVenda]);
3. Existirem dois Campos de utilizador definidos nas linhas do Editor de Vendas (tabela [LinhasDoc]):
  - “CDU\_NumDeUnidades” com descrição “N.º Unidades”, que indicará o n.º de unidades base que correspondem à quantidade vendida;
  - “CDU\_UnidadeBase” com descrição “Unid. Base” que indicará a Unidade base do Artigo referenciado.

Pretende-se com estes campos visualizar duas novas colunas no Editor de Vendas cujas células devem conter, para cada linha que referencie um artigo, a unidade base e o n.º de unidades base correspondentes à unidade de venda.

4. Ter inserido no sistema as seguintes unidades:



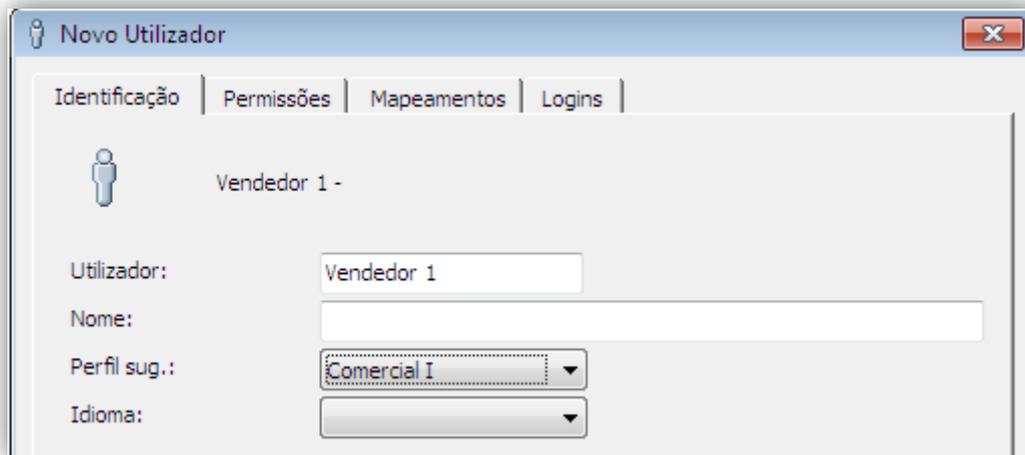
Unidade	Descrição	Arredondamento
UN	Unidade	2
GR	Garrafa	1
CX	Caixa	1

5. Ter alguns artigos cuja unidade Base difere da unidade de Venda. A imagem abaixo exemplifica para o caso do Artigo “A0001” cuja Unidade de Venda (CX - Caixa) corresponde a 12 unidades de Compra (GR - Garrafa):

Un.Origem	Descrição	Un.Destino	Descrição	Factor	Fórmula
CX	Caixa	GR	Garrafa	12,000000	

- Ter configurado o Editor de Vendas, para o utilizador em causa, para apresentar as colunas “N.º Unidades” e “Unid. Base” sempre que se seleccione um documento do tipo “Factura”:

- Ter configurado no Administrador PRIMAVERA , meramente a título de exemplo, dois Utilizadores:



Novo Utilizador

Identificação | Permissões | Mapeamentos | Logins

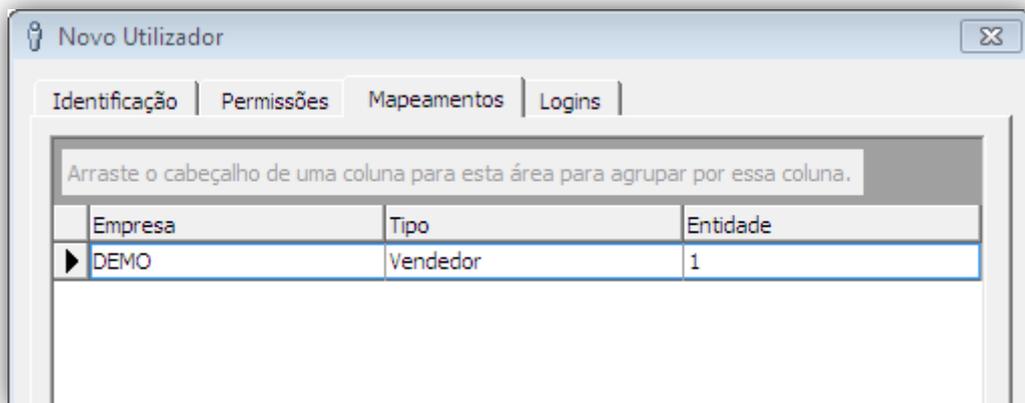
Vendedor 1 -

Utilizador:

Nome:

Perfil sug.:

Idioma:



Novo Utilizador

Identificação | Permissões | Mapeamentos | Logins

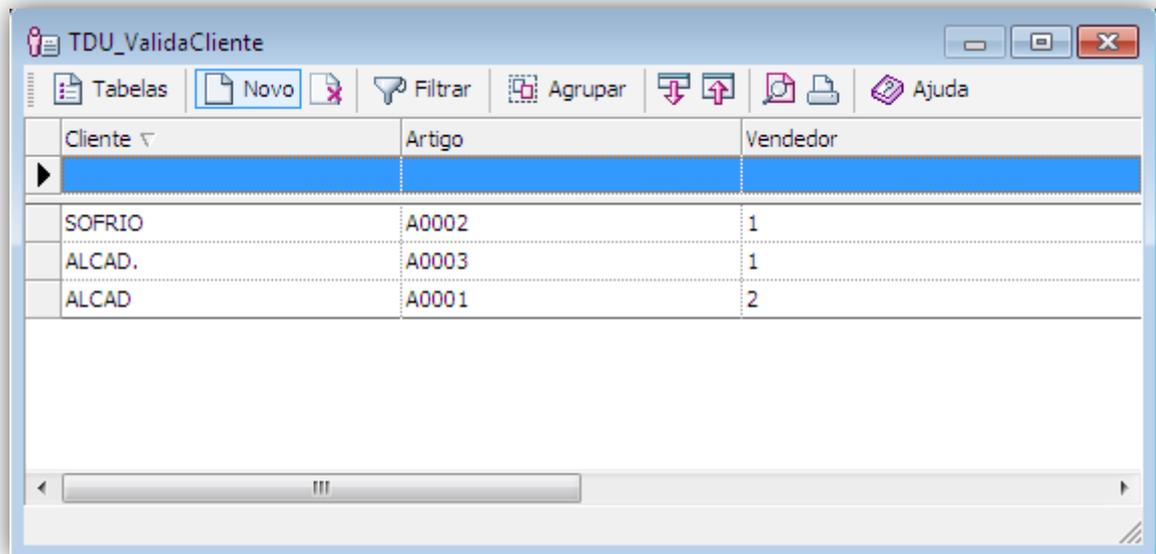
Arraste o cabeçalho de uma coluna para esta área para agrupar por essa coluna.

Empresa	Tipo	Entidade
▶ DEMO	Vendedor	1

O utilizador “Vendedor1” corresponde ao vendedor “1” para a Base de Dados em causa (DEMO);

Criar um segundo utilizador “Vendedor2” que corresponda ao vendedor “2” para a Base de Dados em causa (DEMO);

- Existir uma Tabela de Utilizador com os valores apresentados abaixo:



Cliente	Artigo	Vendedor
SOFRIO	A0002	1
ALCAD.	A0003	1
ALCAD	A0001	2

O objetivo que vamos dar a esta tabela é indicar quais os Artigos que não podem ser vendidos a determinados Clientes por determinado Vendedor.

Assim, no exemplo acima, a primeira linha indica que o Utilizador associado ao Vendedor "1" não pode efetuar vendas que incluam o Artigo "A0003" se o destinatário da venda for o cliente "ALCAD".

Apresenta-se de seguida o código que permite implementar a funcionalidade descrita.

```
Private Sub EditorVendas_ArtigoIdentificado(Artigo As String, NumLinha As Long,
Cancel As Boolean)
' O código abaixo permite efetuar três ações distintas, todas relacionadas com um
Artigo:
' 1) Garantir que o Vendedor "X" não vende o Artigo "Y" ao Cliente "Z";
' 2) Atribuir por defeito um Armazém a uma linha de um Artigo, desde que o
Documento de Venda
'   contenha o armazém pretendido num Campo de Utilizador criado para o efeito;
' 3) Apresentar informação sobre as Unidade Base dos Artigos e a sua relação com a
respetiva
'   Unidade de Venda configurada no próprio Artigo.

Dim dblFactorConversao As Double
Dim dblNumUnidades As Double
Dim lngI As Long
Dim strArmazem As String
Dim strUnidadeBase As String
Dim strUnidadeVenda As String
Dim strUtilizadorActual As String
Dim strVendedorActual As String
```

```
Dim strClienteActual      As String
Dim vntAuxiliar          As Variant 'utilizada apenas para conter valores
temporários
Dim objChave             As StdBECamposChave

Const strTabelaUtilizador As String = "TDU_ValidaCliente"

    On Error GoTo ERRO

    'Inicialização das variáveis
    strArmazem = vbNullString
    strArtigo = vbNullString
    strUnidadeVenda = vbNullString
    strUnidadeBase = vbNullString
    strUtilizadorActual = vbNullString
    strVendedorActual = vbNullString
    strClienteActual = vbNullString
    Set objChave = Nothing

    'verificar se o artigo vem corretamente preenchido
    If (Len(Artigo) <= 0) Then Exit Sub

    With Aplicacao

        '-----
        -----
        '1) Garantir que o Vendedor "X" não vende o Artigo "Y" ao Cliente "Z"
        strUtilizadorActual = .BSO.Contexto.UtilizadorActual
        If (Len(strUtilizadorActual) > 0) Then
            'se existe sistema de segurança ativo - caso contrário não faz sentido
            esta validação

            'obter o vendedor associado a este utilizador, se existir
            strVendedorActual =
PlataformaPRIMAVERA.Administrador.NovoMotorAdm.Utilizadores.DaMapeamentoUtilizador(
-
                                                strUtilizadorActual,
.Empresa.CodEmp, temVendedor)

            If (Len(strVendedorActual) > 0) Then
                'se existe vendedor associado

                'obter o cliente atual
                strClienteActual = Me.DocumentoVenda.Entidade

                Set objChave = New StdBECamposChave
```

```

objChave.AddCampoChave "CDU_Artigo", UCase$(Artigo)
objChave.AddCampoChave "CDU_Cliente", strClienteActual
objChave.AddCampoChave "CDU_Vendedor", strVendedorActual

    If (Aplicacao.BSO.TabelasUtilizador.Existe(strTabelaUtilizador,
objChave)) Then
        'se existe um registo que impede o artigo ser vendido a este
cliente por este vendedor...
        'então -> remover a linha referente ao artigo
        MsgBox "Não é possível atribuir o artigo '" & UCase$(Artigo) &
''' ao cliente '" & _
            UCase$(strClienteActual) & "'.'"
        Me.DocumentoVenda.Linhas.Remove (NumLinha)
        Set oChave = Nothing
        Exit Sub 'tem que sair do procedimento para evitar realizar
todo o código abaixo pois
removida
            'não faz sentido porque a linha do artigo será
removida
        End If

        Set oChave = Nothing

    End If

End If

'-----
'-----
'2) Atribuir por defeito um Armazém a uma linha de um Artigo
'Obter o armazém que está configurado para este Documento de Venda
vntAuxiliar =
.BSO.Comercial.TabVendas.DaValorAtributo(Me.DocumentoVenda.TipoDoc, "CDU_Armazem")
strArmazem = PlataformaPRIMAVERA.Utills.FStr$(vntAuxiliar)

'Se existe um armazém configurado, atribuir esse armazém à respetiva linha
If (Len(strArmazem) > 0) Then Me.DocumentoVenda.Linhas(NumLinha).Armazem =
strArmazem

'-----
'-----
'3) Apresentar informação sobre a Unidade Base do Artigo e sua relação com
a Unidade de Venda

'Para a linha atual do Documento a ser gravado...
If (Len(strArtigo) > 0) Then

```

```
'Se a linha refere um Artigo

'Obter o identificador da Unidade de Venda
vntAuxiliar = .BSO.Comercial.Artigos.DaValorAtributo(Artigo,
"UnidadeVenda")
strUnidadeVenda = PlataformaPRIMAVERA.Utills.FStr$(vntAuxiliar)

If (Len(strUnidadeVenda) > 0) Then
    'Se este Artigo tem uma Unidade de Venda definida

    'Obter o identificador da Unidade Base
    vntAuxiliar = .BSO.Comercial.Artigos.DaValorAtributo(Artigo,
"UnidadeBase")
    strUnidadeBase = PlataformaPRIMAVERA.Utills.FStr$(vntAuxiliar)

    'Atualizar de imediato a coluna da Unidade Base
Me.DocumentoVenda.Linhas(NumLinha).CamposUtil.Item("CDU_UnidadeBase").Valor =
strUnidadeBase

    vntAuxiliar =
.BSO.Comercial.ArtigosUnidades.DaValorAtributo(Artigo, strUnidadeVenda, _
strUnidadeBase,
"FactorConversao")
    dblFactorConversao = PlataformaPRIMAVERA.Utills.FDbl#(vntAuxiliar)

    dblNumUnidades = Me.DocumentoVenda.Linhas(NumLinha).Quantidade *
dblFactorConversao

Me.DocumentoVenda.Linhas(NumLinha).CamposUtil.Item("CDU_NumDeUnidades").Valor =
dblNumUnidades

    End If

End If

End With

Exit Sub

ERRO:
'Antes de apresentar o erro limpa o(s) objeto(s) da memória
If Not (objChave Is Nothing) Then Set objChave = Nothing
MsgBox "Ocorreu o erro abaixo: " & vbCrLf & Err.Number & vbCrLf &
Err.Description, vbCritical, Aplicacao.Nome
End Sub
```

## Duplicação de Documentos de Compras

Neste exemplo vamos personalizar uma interface para o utilizador de forma a este tomar algumas decisões antes de efetuar as operações pretendidas.

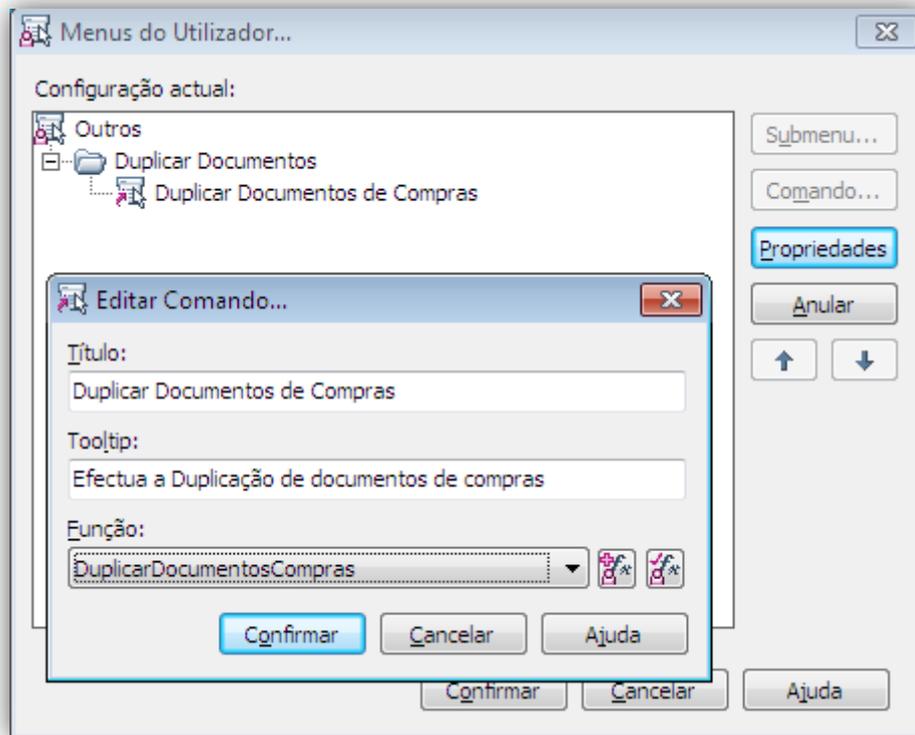
Temos como objetivo apresentar ao utilizador uma janela que lhe permita efetuar a duplicação de um Documento de Compra para um Documento de Venda.

Para tal o utilizador deve, num 1.º passo, escolher o Fornecedor, o Tipo de Documento de Compra pretendido, a sua respetiva Série e o Numerador do Documento. Desta forma identifica univocamente o Documento de Compra em causa.

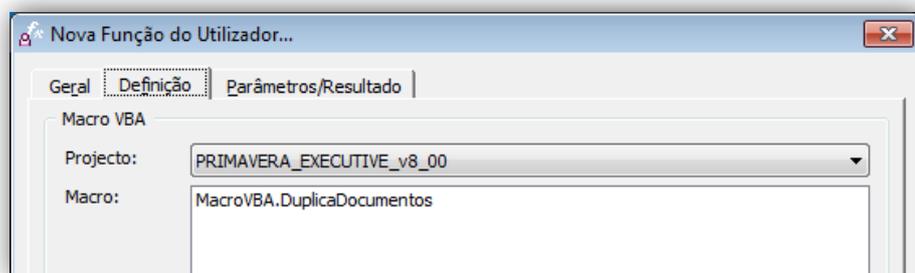
Seguidamente deve identificar o Tipo de Documento de Venda a criar, a sua respetiva Série, o Cliente ao qual vai ficar afeto e, caso deseje, uma Margem que lhe permitirá alterar os valores de Venda dos respetivos Artigos a incluir.

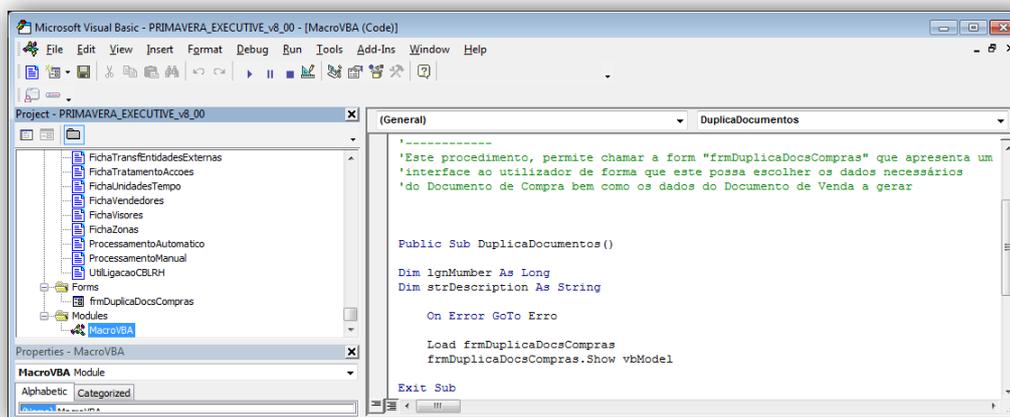
Para executar o exemplo abaixo devem-se verificar as seguintes condições:

1. Existir as seguintes referências no Projeto VBA;
  - PRIMAVERA EXECUTIVE 8.00;
  - PRIMAVERA GcpBE 8.00;
  - PRIMAVERA StdBE 8.00.
2. Existir um Menu de Utilizador “Duplicar Documentos de Compras” (consultar o capítulo “Menus do Utilizador” para efetuar a criação deste menu e a definição do comando respetivo):



A função “DuplicarDocumentosCompra” é uma Macro VBA que chama a Form “frmDuplicaDocsCompras” criada previamente no VBA:





Para disponibilizar esta funcionalidade apresenta-se de seguida o código que necessita ser implementado:

#### Option Explicit

'NOTAS: 1) A função "PlataformaPRIMAVERA.Utills.FStr\$( )" permite tratar valores inválidos lidos da Base de Dados  
' Sempre que o valor retornado da Base de Dados for NULL a função retorna "vbNullString"

#### 'Compras

```

Private ArrDocsCompras() As String
Private ArrSeriesCompras() As String
Private ArrFornecedores() As String
Private ArrNumsCompras() As String
    
```

#### 'Vendas

```

Private ArrDocsVendas() As String
Private ArrClientes() As String
Private ArrSeriesVendas() As String
Private ArrNumsVendas() As String
    
```

#### Private Sub CarregaDocumentosDeCompras()

```

Dim blnModificar As Boolean
Dim blnNovo As Boolean
Dim intContador As Integer
Dim strAuxiliar As String
Dim strSQL As String
Dim objLista As StdBELista
    
```

```
On Error GoTo ERRO

strSQL = "SELECT Documento, Descricao FROM DocumentosCompra"
Set objLista = Aplicacao.BSO.Consulta(strSQL)

'Se existirem documentos de compras definidos
If Not (objLista Is Nothing) Then

    'Inicialização dos controlos e variáveis
    ReDim ArrDocsCompras(objLista.NumLinhas)
    cboTipoDocC.Clear
    intContador = 0
    strAuxiliar = vbNullString

    'Percorrer a lista com todos os documentos existentes e preencher a respetiva
lista
    While Not (objLista.NoInicio Or objLista.NoFim)

        'Verificar se o utilizador atual tem permissões para aceder a este
documento
        strAuxiliar = PlataformaPRIMAVERA.Utills.FStr$(objLista("Documento"))
        blnModificar = Aplicacao.Utilizador.AcedeAtributo("Documentos de Compra",
strAuxiliar, "Modificar")
        blnNovo = Aplicacao.Utilizador.AcedeAtributo("Documentos de Compra",
strAuxiliar, "Criar")

        If blnModificar Or blnNovo Then
            ArrDocsCompras(intContador) = strAuxiliar
            strAuxiliar = strAuxiliar & " - " &
PlataformaPRIMAVERA.Utills.FStr$(objLista("Descricao"))
            cboTipoDocC.AddItem strAuxiliar
            intContador = intContador + 1
        End If

        'Item seguinte da lista
        objLista.Seguente

    Wend

    'Posiciona-se no primeiro item da lista
    If (cboTipoDocC.ListCount > 0) Then cboTipoDocC.ListIndex = 0

    'Limpar o objeto de memória
    Set objLista = Nothing

End If
```

```
Exit Sub

ERRO:
'Antes de apresentar o erro limpa o(s) objeto(s) da memória
If Not (objLista Is Nothing) Then Set objLista = Nothing
MsgBox "Ocorreu o erro abaixo indicado: " & vbCrLf & Err.Number & vbCrLf &
Err.Description, vbCritical, Aplicacao.Nome
End Sub

Private Sub CarregaDocumentosDeVendas()

Dim blnModificar As Boolean
Dim blnNovo As Boolean
Dim intContador As Integer
Dim strSQL As String
Dim strAuxiliar As String
Dim objLista As StdBELista

On Error GoTo ERRO

strSQL = "SELECT Documento, Descricao FROM DocumentosVenda"
Set objLista = Aplicacao.BSO.Consulta(strSQL)

'Se existirem documentos de vendas definidos
If Not (objLista Is Nothing) Then

'Inicialização dos controlos e variáveis
ReDim ArrDocsVendas(objLista.NumLinhas)
txtMargem.Value = 0
cboTipoDocV.Clear
intContador = 0
strAuxiliar = vbNullString

'Percorrer a lista com todos os documentos existentes e preencher a respetiva
lista
While Not (objLista.NoInicio Or objLista.NoFim)

'Verificar se o utilizador atual tem permissões para aceder a este
documento
strAuxiliar = PlataformaPRIMAVERA.Utills.FStr$(objLista("Documento"))
blnModificar = Aplicacao.Utilizador.AcedeAtributo("Documentos de Venda",
strAuxiliar, "Modificar")
blnNovo = Aplicacao.Utilizador.AcedeAtributo("Documentos de Venda",
strAuxiliar, "Criar")
```

```

        If blnModificar Or blnNovo Then
            ArrDocsVendas(intContador) = strAuxiliar
            strAuxiliar = strAuxiliar & " - " &
PlataformaPRIMAVERA.Utills.FStr$(objLista("Descricao"))
            cboTipoDocV.AddItem strAuxiliar
            intContador = intContador + 1
        End If

        'Item seguinte da lista
        objLista.Seguinte

    Wend

    'Posiciona-se no primeiro item da lista
    If (cboTipoDocV.ListCount > 0) Then cboTipoDocV.ListIndex = 0

End If

'Limpar o objeto de memória
Set objLista = Nothing

Exit Sub

ERRO:
'Antes de apresentar o erro limpa o(s) objeto(s) da memória
If Not (objLista Is Nothing) Then Set objLista = Nothing
MsgBox "Ocorreu o erro abaixo indicado: " & vbCrLf & Err.Number & vbCrLf &
Err.Description, vbCritical, Aplicacao.Nome
End Sub

Private Sub cboFornecedores_Click()

Dim intContador    As Integer
Dim strSQL         As String
Dim objLista       As StdBLista

    On Error GoTo ERRO

    strSQL = "SELECT NumDoc, ID FROM CabecCompras "
    strSQL = strSQL & "WHERE ((TipoDoc = '" & ArrDocsCompras(cboTipoDocC.ListIndex) &
    "')"
    strSQL = strSQL & "AND (Serie = '" & ArrSeriesCompras(cboSerieC.ListIndex) & "')"
    strSQL = strSQL & "AND (Entidade = '" &
ArrFornecedores(cboFornecedores.ListIndex) & "'))"
    strSQL = strSQL & "ORDER BY NumDoc ASC"
    Set objLista = Aplicacao.BSO.Consulta(strSQL)

```

```
'Se existirem Documentos definidos
If Not (objLista Is Nothing) Then

    'Inicialização dos controlos e variáveis
    ReDim ArrNumsCompras(objLista.NumLinhas)
    cboDocumentos.Clear
    intContador = 0

    'Percorrer a lista com todos os Documentos existentes e preencher a respetiva
lista
    While Not (objLista.NoInicio Or objLista.NoFim)

        cboDocumentos.AddItem PlataformaPRIMAVERA.Utills.FStr$(objLista("NumDoc"))
        ArrNumsCompras(intContador) =
PlataformaPRIMAVERA.Utills.FStr$(objLista("ID"))
        intContador = intContador + 1

        'Item seguinte da lista
        objLista.Seguinte

    Wend

    'Limpar o objeto de memória
    Set objLista = Nothing

End If

Exit Sub

ERRO:
'Antes de apresentar o erro limpa o(s) objeto(s) da memória
If Not (objLista Is Nothing) Then Set objLista = Nothing
MsgBox "Ocorreu o erro abaixo indicado: " & vbCrLf & Err.Number & vbCrLf &
Err.Description, vbCritical, Aplicacao.Nome
End Sub

Private Sub cboSerieC_Click()

Dim intContador    As Integer
Dim strSQL         As String
Dim strAuxiliar    As String
Dim objLista       As StdBELista

    On Error GoTo ERRO
```

```
strSQL = "SELECT DISTINCT C.Entidade, F.Nome "  
strSQL = strSQL & "FROM CabecCompras AS C INNER JOIN Fornecedores AS F ON  
(C.Entidade = F.Fornecedor) "  
strSQL = strSQL & "WHERE ((C.TipoDoc = '" & ArrDocsCompras(cboTipoDocC.ListIndex)  
& "') "  
strSQL = strSQL & "AND (C.Serie = '" & ArrSeriesCompras(cboSerieC.ListIndex) &  
"')) "  
Set objLista = Aplicacao.BSO.Consulta(strSQL)  
  
'Se existirem Documentos definidos  
If Not (objLista Is Nothing) Then  
  
    'Inicialização dos controlos e variáveis  
    ReDim ArrFornecedores(objLista.NumLinhas)  
    cboFornecedores.Clear  
    intContador = 0  
    strAuxiliar = vbNullString  
  
    'Percorrer a lista com todas as Entidades existentes e preencher a respetiva  
lista  
    While Not (objLista.NoInicio Or objLista.NoFim)  
  
        strAuxiliar = PlataformaPRIMAVERA.Utills.FStr$(objLista("Entidade"))  
        ArrFornecedores(intContador) = strAuxiliar  
        strAuxiliar = strAuxiliar & " - " &  
PlataformaPRIMAVERA.Utills.FStr$(objLista("Nome"))  
        cboFornecedores.AddItem strAuxiliar  
        intContador = intContador + 1  
  
        'Item seguinte da lista  
        objLista.Seguinte  
  
    Wend  
  
    'Limpar o objeto de memória  
    Set objLista = Nothing  
  
End If  
  
Exit Sub  
  
ERRO:  
    'Antes de apresentar o erro limpa o(s) objeto(s) da memória  
    If Not (objLista Is Nothing) Then Set objLista = Nothing  
    MsgBox "Ocorreu o erro abaixo indicado: " & vbCrLf & Err.Number & vbCrLf &  
Err.Description, vbCritical, Aplicacao.Nome
```

```
End Sub

Private Sub cboSerieV_Click()

Dim intContador      As Integer
Dim strSQL           As String
Dim strAuxiliar      As String
Dim objLista         As StdBLista

    On Error GoTo ERRO

    strSQL = "SELECT DISTINCT Cliente AS Entidade, Nome FROM Clientes"
    Set objLista = Aplicacao.BSO.Consulta(strSQL)

    'Se existirem Documentos definidos
    If Not (objLista Is Nothing) Then

        'Inicialização dos controlos e variáveis
        ReDim ArrClientes(objLista.NumLinhas)
        cboCliente.Clear
        intContador = 0

        'Percorrer a lista com todos os Clientes existentes e preencher a respetiva
lista
        While Not (objLista.NoInicio Or objLista.NoFim)

            strAuxiliar = PlataformaPRIMAVERA.Utills.FStr$(objLista("Entidade"))
            ArrClientes(intContador) = strAuxiliar
            strAuxiliar = strAuxiliar & " - " &
PlataformaPRIMAVERA.Utills.FStr$(objLista("Nome"))
            cboCliente.AddItem strAuxiliar
            intContador = intContador + 1

            'Item seguinte da lista
            objLista.Seguente

        Wend

        'Limpar o objeto de memória
        Set objLista = Nothing

    End If

Exit Sub
```

```
ERRO:
    'Antes de apresentar o erro limpa o(s) objeto(s) da memória
    If Not (objLista Is Nothing) Then Set objLista = Nothing
    MsgBox "Ocorreu o erro abaixo indicado: " & vbCrLf & Err.Number & vbCrLf &
Err.Description, vbCritical, Aplicacao.Nome
End Sub

Private Sub cboTipoDocV_Change()

Dim intContador      As Integer
Dim strSQL           As String
Dim strAuxiliar      As String
Dim objLista         As StdBLista

    On Error GoTo ERRO

    strSQL = "SELECT * FROM SeriesVendas WHERE (TipoDoc = '' &
ArrDocsVendas(Me.cboTipoDocV.ListIndex) & '')"
    Set objLista = Aplicacao.BSO.Consulta(strSQL)

    'Se existirem Documentos definidos
    If Not (objLista Is Nothing) Then

        'Inicialização dos controlos e variáveis
        ReDim ArrSeriesVendas(objLista.NumLinhas)
        cboSerieV.Clear
        intContador = 0
        strAuxiliar = vbNullString

        'Percorrer a lista com todas as Séries existentes e preencher a respetiva
lista
        While Not (objLista.NoInicio Or objLista.NoFim)

            strAuxiliar = PlataformaPRIMAVERA.Utills.FStr$(objLista("Serie"))
            ArrSeriesVendas(intContador) = strAuxiliar
            strAuxiliar = strAuxiliar & " - " &
PlataformaPRIMAVERA.Utills.FStr$(objLista("Descricao"))
            cboSerieV.AddItem strAuxiliar
            intContador = intContador + 1

            'Item seguinte da lista
            objLista.Seguinte

        Wend

        'Limpar o objeto de memória
```

```

        Set objLista = Nothing

    End If

    Exit Sub

ERRO:
    'Antes de apresentar o erro limpa o(s) objeto(s) da memória
    If Not (objLista Is Nothing) Then Set objLista = Nothing
    MsgBox "Ocorreu o erro abaixo indicado: " & vbCrLf & Err.Number & vbCrLf &
    Err.Description, vbCritical, Aplicacao.Nome
End Sub

Private Sub cboTipoDocC_Click()

    Dim intContador      As Integer
    Dim strSQL           As String
    Dim strAuxiliar      As String
    Dim objLista         As StdBLista

    On Error GoTo ERRO

    strSQL = "SELECT Serie, Descricao FROM SeriesCompras WHERE (TipoDoc = '" &
    ArrDocsCompras(cboTipoDocC.ListIndex) & "')"
    Set objLista = Aplicacao.BSO.Consulta(strSQL)

    'Se existirem séries definidas para o Documento de compras selecionado
    If Not (objLista Is Nothing) Then

        'Inicialização dos controlos e variáveis
        ReDim ArrSeriesCompras(objLista.NumLinhas)
        cboSerieC.Clear
        cboFornecedores.Clear
        cboDocumentos.Clear
        intContador = 0
        strAuxiliar = vbNullString

        'Percorrer a lista com todas as séries existentes e preencher a respetiva
lista
        While Not (objLista.NoInicio Or objLista.NoFim)

            strAuxiliar = PlataformaPRIMAVERA.Utills.FStr$(objLista("Serie")) & " - "
            strAuxiliar = strAuxiliar &
            PlataformaPRIMAVERA.Utills.FStr$(objLista("Descricao"))
            cboSerieC.AddItem strAuxiliar
        
```

```
        ArrSeriesCompras(intContador) =  
PlataformaPRIMAVERA.Utills.FStr$(objLista("Serie"))  
        intContador = intContador + 1  
  
        'Item seguinte da lista  
        objLista.Seguinte  
  
    Wend  
  
    'Limpar o objeto de memória  
    Set objLista = Nothing  
  
End If  
  
Exit Sub  
  
ERRO:  
    'Antes de apresentar o erro limpa o(s) objeto(s) da memória  
    If Not (objLista Is Nothing) Then Set objLista = Nothing  
    MsgBox "Ocorreu o erro abaixo indicado: " & vbCrLf & Err.Number & vbCrLf &  
Err.Description, vbCritical, Aplicacao.Nome  
End Sub  
  
Private Sub UserForm_Activate()  
  
    CarregaDocumentosDeCompras  
    CarregaDocumentosDeVendas  
  
End Sub  
  
Private Sub cmdConfirmar_Click()  
  
Dim strMensagemErro    As String  
Dim blnErroValidacao   As Boolean  
  
    On Error GoTo ERRO  
  
    strMensagemErro = vbNullString  
    blnErroValidacao = False  
  
    'Verificar se a série do documento de compra está corretamente preenchida  
    If (cboSerieC.ListIndex = -1) Then  
        strMensagemErro = strMensagemErro & "- A série do Documento de Compras não é  
válida." & vbCrLf  
        blnErroValidacao = True  
    End If
```

```
'Verificar se o identificador do documento de compra está corretamente preenchido
If (cboDocumentos.ListIndex = -1) Then
    strMensagemErro = strMensagemErro & "- O Número do Documento a duplicar é
incorreto." & vbCrLf
    blnErroValidacao = True
End If

'Verificar se a série do documento de venda está corretamente preenchida
If (cboSerieV.ListIndex = -1) Then
    strMensagemErro = strMensagemErro & "- A série do Documento de Vendas não é
válida." & vbCrLf
    blnErroValidacao = True
End If

'Verificar se a margem do documento de venda está corretamente preenchida
If (Not IsNumeric(txtMargem.Value)) Then
    strMensagemErro = strMensagemErro & "- A Margem definida não é correta." &
vbCrLf
    blnErroValidacao = True
End If

'Verifica se o resultado das validações impede a duplicação dos documentos
If blnErroValidacao Then
    strMensagemErro = "Existem dados incorretamente preenchidos:" & vbCrLf &
strMensagemErro
    MsgBox strMensagemErro, vbExclamation, Aplicacao.Nome
    Exit Sub
End If

'Se as validações estão todas OK e o utilizador permitir, duplica o documento
If (MsgBox("Confirma a duplicação do Documento de Compra para um Documento de
Venda?", vbYesNo) = vbYes) Then
    DuplicaDocumentoDeCompra
End If

Exit Sub

ERRO:
    MsgBox "Ocorreu o erro abaixo indicado: " & vbCrLf & Err.Number & vbCrLf &
Err.Description, vbCritical, Aplicacao.Nome
End Sub

Private Sub cmdCancelar_Click()

Unload Me
```

```
End Sub

Private Sub DuplicaDocumentoDeCompra()

Dim strSQL          As String
Dim strMensagem     As String
Dim objCompra       As GcpBEDocumentoCompra
Dim objLinhaCompra As GcpBELinhaDocumentoCompra
Dim objVenda        As GcpBEDocumentoVenda      'Para carregar o objeto e
percorrer as linhas para atribuição.
Dim objLinhaVenda   As GcpBELinhaDocumentoVenda
Dim objLista        As StdBELista

Const TIPO_ENTIDADE_CLIENTE As String = "C"

    On Error GoTo ERRO

    Set objCompra =
Aplicacao.BSO.Comercial.Compras.EditaID(ArrNumsCompras(cboDocumentos.ListIndex))

    'Se existir o Documento selecionado
    If Not (objCompra Is Nothing) Then

        'Criar o objeto de Venda para gravar as informações
        Set objVenda = New GcpBEDocumentoVenda

        objVenda.TipoEntidade = TIPO_ENTIDADE_CLIENTE

        objVenda.Entidade = ArrClientes(cboCliente.ListIndex)
        objVenda.TipoDoc = ArrDocsVendas(cboTipoDocV.ListIndex)
        objVenda.Serie = ArrSeriesVendas(cboSerieV.ListIndex)

        Aplicacao.BSO.Comercial.Vendas.PreencheDadosRelacionados objVenda,
vdDadosTodos

        For Each objLinhaCompra In objCompra.Linhas

            'Criar uma linha do Documento de venda
            Set objLinhaVenda = New GcpBELinhaDocumentoVenda

            With objLinhaVenda

                .Armazem = objLinhaCompra.Armazem
                .ArredFConv = objLinhaCompra.ArredFConv
                .Artigo = objLinhaCompra.Artigo
```

```
.CodigoBarras = objLinhaCompra.CodigoBarras
.CodIva = objLinhaCompra.CodIva
.DataEntrega = objLinhaCompra.DataEntrega
.Desconto1 = objLinhaCompra.Desconto1
.Desconto2 = objLinhaCompra.Desconto2
.Desconto3 = objLinhaCompra.Desconto3
.DescontoComercial = objLinhaCompra.DescontoComercial
.Descricao = objLinhaCompra.Descricao
.Estado = objLinhaCompra.Estado
.FactorConv = objLinhaCompra.FactorConv
.Formula = objLinhaCompra.Formula
.IDLinha = vbNullString
.IDLinhaOriginal = vbNullString
.IntrastatCodigoPautal = objLinhaCompra.IntrastatCodigoPautal
.IntrastatMassaLiq = objLinhaCompra.IntrastatMassaLiq
.IntrastatRegiao = objLinhaCompra.IntrastatRegiao
.IntrastatValorLiq = objLinhaCompra.IntrastatValorLiq

.Lote = objLinhaCompra.Lote
.MovStock = objLinhaCompra.MovStock

.NumLinDocOriginal = 0
.NumLinhaStkGerada = 0
.PrecoLiquido = objLinhaCompra.PrecoLiquido
.QuantFormula = objLinhaCompra.QuantFormula
.Quantidade = objLinhaCompra.Quantidade
.QuantReservada = objLinhaCompra.QuantReservada
.QuantSatisfeita = objLinhaCompra.QuantSatisfeita
.RegimeIva = objLinhaCompra.RegimeIva
.SujeitoRetencao = objLinhaCompra.SujeitoRetencao
.TaxaIva = objLinhaCompra.TaxaIva
.TipoLinha = objLinhaCompra.TipoLinha
.Unidade = objLinhaCompra.Unidade
.VariavelA = objLinhaCompra.VariavelA
.VariavelB = objLinhaCompra.VariavelB
.VariavelC = objLinhaCompra.VariavelC

'Set .LinhasDimensoes = objLinhaCompra.LinhasDimensoes
Set .NumerosSerie = objLinhaCompra.NumerosSerie

.CamposUtil("CDU_Referencia").Valor =
objLinhaCompra.CamposUtil("CDU_Referencia").Valor

End With
```

```
        If (txtMargem.Value <> 0) Then
            objLinhaVenda.PrecUnit = objLinhaCompra.PrecUnit +
            (objLinhaCompra.PrecUnit / txtMargem.Value)
        Else
            objLinhaVenda.PrecUnit = objLinhaCompra.PrecUnit
        End If

        objVenda.Linhas.Insere objLinhaVenda
        Set objLinhaVenda = Nothing

    Next

    Aplicacao.BSO.Comercial.Vendas.Actualiza objVenda

    'Limpar os objetos de memória
    Set objVenda = Nothing
    Set objCompra = Nothing

    strMensagem = "Foi criado o documento indicado abaixo:" & vbCrLf
    strMensagem = strMensagem & "Documento: [" & objVenda.TipoDoc & "]; "
    strMensagem = strMensagem & "Série: [" & objVenda.Serie & "]; "
    strMensagem = strMensagem & "Numerador: [" & objVenda.NumDoc & "]"
    MsgBox strMensagem, vbInformation, Aplicacao.Nome

Else

    strMensagem = "O Documento de Compra selecionado não existe."
    MsgBox strMensagem, vbExclamation, "Duplicação de Documentos de Compra"

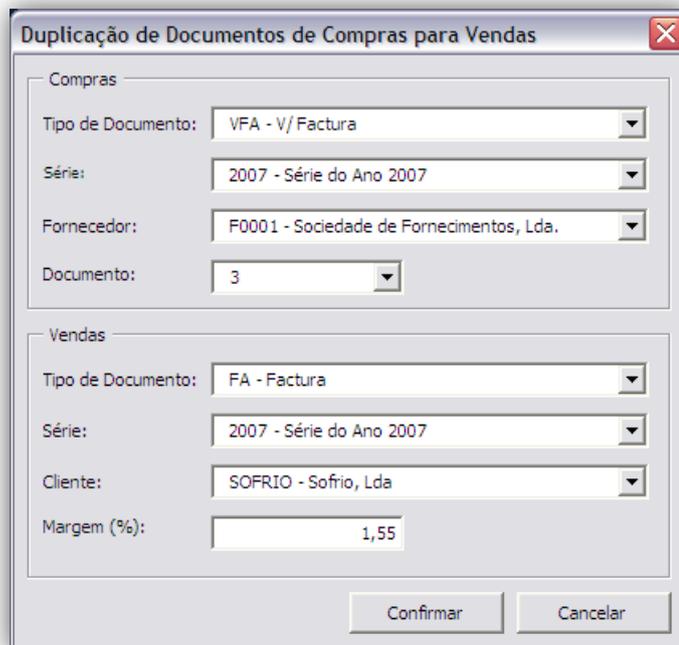
End If

Exit Sub

ERRO:
    strMensagem = Err.Description
    'Antes de apresentar o erro limpa o(s) objeto(s) da memória
    If Not (objCompra Is Nothing) Then Set objCompra = Nothing
    If Not (objLinhaCompra Is Nothing) Then Set objLinhaCompra = Nothing
    If Not (objVenda Is Nothing) Then Set objVenda = Nothing
    If Not (objLinhaVenda Is Nothing) Then Set objLinhaVenda = Nothing
    MsgBox "Ocorreu o erro abaixo indicado: " & vbCrLf & Err.Number & vbCrLf &
    strMensagem, vbCritical, Aplicacao.Nome
End Sub
```



Apresenta-se de seguida a janela que o utilizador visualizará ao executar este comando, e na qual já se efetuaram algumas escolhas.



Para gerar o respetivo Documento de Venda indicado na escolha do utilizador basta clicar no botão “Confirmar”.

Estas alterações estão dependentes das validações do Motor de Vendas e poderá existir informação em falta no Documento de Compra que impeça a criação do respetivo Documento de Venda.

### Agendar o Recálculo de Stocks para efetuar periodicamente

Pretende-se efetuar as operações de Recalculo de Stocks de forma periódica, através de um Processo de Utilizador que execute uma Macro VBA contendo o código necessário à execução do Recalculo com algumas parametrizações definidas por defeito.

Na inicialização de variáveis realizada na Macro VBA criada, tomam-se várias decisões:

- 1) Recalcular só os artigos marcados para recalculo;
- 2) Não recalcular reservas de encomendas;
- 3) Atualizar o Preço de Custo Médio nos Documentos de Venda;
- 4) Efetuar o Recalculo para todos os Artigos;
- 5) Não efetuar o Recalculo por Armazém (supomos que configurámos no Administrador a Valorização do Stock por P.C.M. global).

Para disponibilizar esta funcionalidade apresenta-se de seguida o código que necessita ser implementado na Macro VBA:

```

Option Explicit

'-----
'-----
'Variáveis globais para o Recálculo
Private blnRecalculaMarcados As Boolean
Private blnRecalculaReservas As Boolean
Private blnActualizaDocsVenda As Boolean
Private strArtigoInicial As String
Private strArtigoFinal As String

' Este procedimento permite chamar os respetivos métodos do Motor de forma a executar
' corretamente as operações de recálculo de stocks sobre o exercício atual.

Public Sub RecalculoStocksVBA()

    On Error GoTo Erro

    'Inicialização dos parâmetros necessários ao recálculo
    DefineParametrosRecalculo

    'Operação de Recálculo de Stocks
    ExecutaRecalculo

    Exit Sub

Erro:
    lngNumber = Err.Number
    strDescription = Err.Description
    MsgBox "Ocorreu o erro abaixo indicado: " & vbCrLf & lngNumber & vbCrLf &
    strDescription, vbCritical, Aplicacao.Nome
End Sub

'-----
'-----
'Na inicialização de variáveis tomámos várias decisões:
' 1) Recalcular só os artigos marcados para recálculo
' 2) Não recalcular reservas de encomendas
' 3) Atualizar o Preço de Custo Médio nos Documentos de Venda
' 4) Efetuar o Recálculo para TODOS os Artigos
' 5) Não efetuar o Recálculo por Armazém
' (para tal estamos a supôr que se encontra configurado no Administrador
    
```

```
' a Valorização do Stock por Preço de Custo Médio global)

Private Sub DefineParametrosRecalculo()

Dim lngNumber      As Long
Dim strDescription As String
Dim strKey         As String
Dim strSQL        As String
Dim strSource     As String
Dim objLista      As StdBLista

    On Error GoTo Erro

    'Recalcular só os artigos marcados para recálculo
    blnRecalculaMarcados = True

    'Não recalcula reservas de encomendas
    blnRecalculaReservas = False

    'Atualiza P.C.M. nos Documentos de Venda
    blnActualizaDocsVenda = True

    'Inicializa as variáveis com os valores por defeito
    strArtigoInicial = vbNullString
    strArtigoFinal   = vbNullString

    strSQL = "SELECT MIN(Artigo), MAX(Artigo) FROM Artigo"
    Set objLista = Aplicacao.BSO.Consulta(strSQL)

    'Se existem artigos definidos
    If Not (objLista Is Nothing) Then

        If Not (objLista.NoInicio And objLista.NoFim) Then 'existem registos
            strArtigoInicial = PlataformaPRIMAVERA.Utills.FStr$(objLista(0))
            strArtigoFinal   = PlataformaPRIMAVERA.Utills.FStr$(objLista(1))
        End If

        'Limpar o objeto de memória
        Set objLista = Nothing

    End If

    Exit Sub

Erro:
```

```
'Como temos várias operações antes de apresentar o erro convém guardar os valores
'corretos do erro para os apresentar mais tarde...
lngNumber = Err.Number
strSource = Err.Source
strDescription = Err.Description

'Coloca nas variáveis os valores por defeito
strArtigoInicial = vbNullString: strArtigoFinal = vbNullString

'Antes de apresentar o erro limpa o(s) objeto(s) da memória
If Not (objLista Is Nothing) Then Set objLista = Nothing

'Despoletar o erro com a informação correta
Err.Raise lngNumber, strSource, strDescription
End Sub

'-----
'-----
' Opereção de Recálculo de Stocks baseada nas parametrizações anteriormente efetuadas

Private Sub ExecutaRecalculo()

Dim intRet           As Integer
Dim strMensagem     As String
Dim strErro         As String
Dim strArmazemInicial As String
Dim strArmazemFinal As String
Dim msgRes          As VbMsgBoxResult

    On Error GoTo Erro

    strMensagem = "A operação de recálculo atualiza o stock dos artigos a partir dos
    respetivos movimentos. "
    strMensagem = strMensagem & "Simultâneamente procede ao recálculo dos Preços de
    Custo Médio." & vbCrLf
    strMensagem = strMensagem & "Estas operações podem ser demoradas. Deseja
    continuar?"
    msgRes = MsgBox(strMensagem, vbYesNo, Aplicacao.Nome)
    If (msgRes = vbNo) Then Exit Sub

    'Como decidimos não efetuar o Recálculo por Armazém estas variáveis devem levar o
    valor por defeito
    strArmazemInicial = vbNullString
    strArmazemFinal = vbNullString

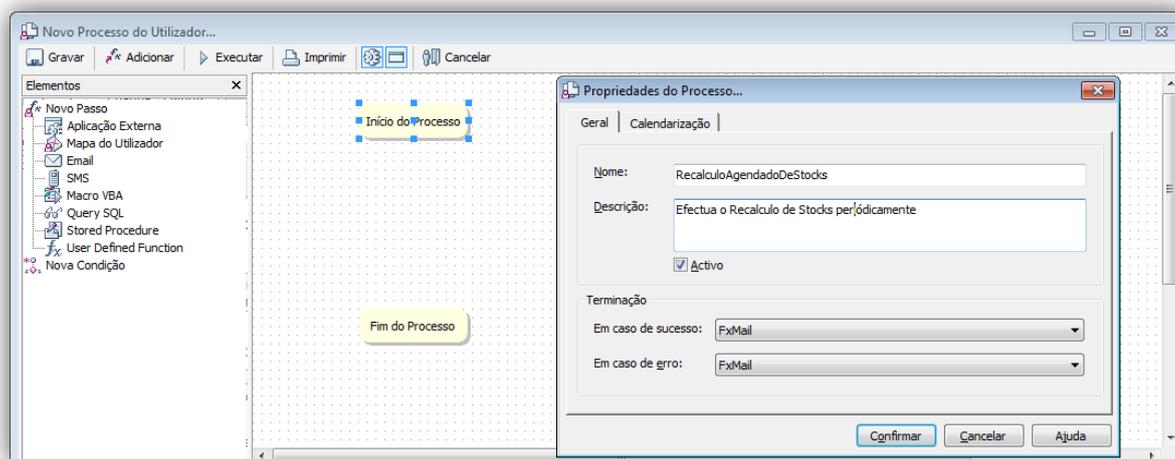
    strErro = vbNullString
```

```
intRet = Aplicacao.BSO.Comercial.Stocks.RecalculoStocks(trcRecalculo,
strArtigoInicial, strArtigoFinal, _
strArmazemInicial,
strArmazemFinal, _
blnArtNecRecalcPCM:=blnRecalculaMarcados, _
blnRecalcQtdReservada:=blnRecalculaReservas, _
blnExecPCMDocs:=blnAtualizaDocsVenda, strMsgErro:=strErro)
If (intRet = 0) Then
    strMensagem = "Recálculo de Stocks efetuado com sucesso."
Else
    strMensagem = "Ocorreu o erro abaixo no Recálculo de Stocks:" & vbCrLf
    strMensagem = strMensagem & strErro
End If
MsgBox strMensagem, vbOKOnly, Aplicacao.Nome

Exit Sub

Erro:
Screen.MousePointer = vbDefault
If Not WaitDialog Is Nothing Then
    WaitDialog.Termina
    Set WaitDialog = Nothing
End If
GCPMostraErro Plataforma.Localizacao.DaResString(cABRV_APL, 5296), PRI_Critico,
Err.Description
End Sub
```

Após disponibilizarmos o código, é necessário criar o Processo de Utilizador que periodicamente executará a Macro criada (consulte o capítulo “Processos do Utilizador”).



### Criação de um documento de liquidação

Este exemplo pretende exemplificar a forma de efetuar uma liquidação através da extensibilidade. O exemplo descrito liquida um dos documentos pendentes para o cliente SOFRIO de um exercício DEMO com sede no continente.

Para executar o exemplo abaixo devem-se verificar as seguintes condições:

1. Existir as seguintes referências no Projeto VBA:
  - PRIMAVERA EXECUTIVE 8.00;
  - PRIMAVERA GcpBE 8.00.

Apresenta-se de seguida o código que permite implementar a funcionalidade descrita.

```
Public Sub Liquida_Documento()
'Este procedimento pretende exemplificar a forma de efetuar uma liquidação
'através da extensibilidade. O exemplo descrito liquida um dos documentos pendentes
'para o cliente SOFRIO de um exercício DEMO com sede no continente.

Dim objDocLiq      As GcpBEDocumentoLiq
Dim ObjLinhaLiq   As GcpBELinhaLiq
Dim strAvisos     As String

On Error GoTo Erro

    Set objDocLiq = New GcpBEDocumentoLiq
```

```
'Preenchimento do cabeçalho do documento de liquidação
With objDocLiq

    'Filial onde está a ser efetuada a liquidação
    .Filial = BSO.Comercial.Filiais.CodigoFilial

    'Tipo de documento de liquidação a utilizar
    .Tipodoc = "RE"

    'Série do documento de liquidação a utilizar
    .Serie = "2"

    'Próximo numero de documento disponível para a série usada
    .NumDoc = BSO.Comercial.Series.ProximoNumero("M", .Tipodoc, .Serie)

    'Tipo de entidade para quem vai ser emitido o documento de liquidação
    'Neste exemplo um cliente
    .TipoEntidade = "C"

    'Entidade para quem vai ser emitido o documento de liquidação
    .Entidade = "SOFRIO"

    'Tipo de conta (Conta-Corrente)
    .TipoConta = "C"

    'Data de liquidação (data atual)
    .DataDoc = Date

    'Moeda utilizada na liquidação
    .Moeda = "EUR"
End With

'Preenchimento dos dados em falta
BSO.Comercial.Liquidacoes.PreencheDadosRelacionados objDocLiq, cctDadosTodos

'Identificação do documento a liquidar (criação de uma linha de liquidação)
Set ObjLinhaLiq = New GcpBELinhaLiq
With ObjLinhaLiq
    'Filial onde foi emitido o documento a liquidar
    .FilialOrig = "000"

    'Módulo a que pertence o documento (Vendas)
    .ModuloOrig = "V"

    'Tipo de documento a liquidar
```

```
.TipoDocOrig = "FA"

'Série do documento a liquidar
.SerieOrig = "2006"

'Numero do documento a liquidar
'No caso de ser por exemplo um documento do módulo de compras
'ser o numero do documento do fornecedor
.NumDocOrig = "13"

'Numero interno do documento a liquidar
'No caso de ser por exemplo um documento do módulo de compras
'podero não ser coincidentes
.NumDocOrigInt = 13

'Numero da prestao do documento a liquidar
'No caso da condio de pagamento do documento
'no ser particionada (prestaes) ser sempre 1
.NumPrestacao = 1

'Data de emisso do documento a liquidar
.DataDoc = "21-11-2006"

'Numero de Transferência do documento
'Este valor poder ser diferente no caso do
'documento ter sofrido uma operação de transferência
'de conta/estado com alteração da entidade do documento
.NumTransferencia = 0

'Tipo de entidade a que pertence o documento a liquidar
'Neste exemplo um cliente
.TipoEntidade = "C"

'Entidade a que pertence o documento a liquidar
.Entidade = "SOFRIO"

'Moeda em que foi emitido o documento a liquidar
.MoedaDocOrig = "EUR"

'Tipo de Conta em que se encontra o documento a liquidar
'Neste exemplo conta-corrente de cliente
.TipoConta = "CCC"

'Estado em que se encontra o documento a liquidar
'Neste exemplo o estado pendentes da conta-corrente de cliente
```

```
.Estado = "PEN"

'Valor a liquidar (na moeda de liquidao)
.ValorRec = 2458.3

'Valor que vai ficar pendente (na moeda de liquidao)
.ValorPend = 0

'Valor original do documento (na moeda de liquidacao)
.ValorOrig = 2458.3

'Valor de desconto (na moeda de liquidacao)
.ValorDesconto = 0

'Valor a liquidar (na moeda de emissao do documento)
.ValorRecMoedaOrig = 2458.3

'Valor de desconto (na moeda de emissao do documento)
.ValorDescMoedaOrig = 0

End With

'Associação da linha de liquidação ao documento
objDocLiq.LinhasLiquidacao.Insere ObjLinhaLiq

'Validação do documento
If BSO.Comercial.Liquidacoes.ValidaActualizacao(objDocLiq, strAvisos) Then
    'Gravação do documento
    BSO.Comercial.Liquidacoes.Actualiza objDocLiq, strAvisos
    PlataformaPRIMAVERA.Dialogos.MostraAviso "Documento criado com sucesso.",
PRI_Informativo
Else
    PlataformaPRIMAVERA.Dialogos.MostraErro "Erro", PRI_Exclama, strAvisos
End If

'Eliminação dos objetos
Set objDocLiq = Nothing
Set ObjLinhaLiq = Nothing
Exit Sub

Erro:
Set objDocLiq = Nothing
Set ObjLinhaLiq = Nothing
If Len(strAvisos) > 0 Then
    PlataformaPRIMAVERA.Dialogos.MostraErro "Erro", PRI_Exclama, strAvisos
```

```
Else
    PlataformaPRIMAVERA.Dialogos.MostraErro "Erro", PRI_Exclama, Err.Number & " - "
    & Err.Description
End If
End Sub
```

### Criação de um documento de transferência de conta/estado

Este exemplo pretende exemplificar a forma de efetuar uma transferência de conta/estado através da extensibilidade. O exemplo descrito transfere um dos documentos pendentes para o fornecedor F0001 de um exercício DEMO com sede no continente. A transferência retira o documento do estado AGP e coloca-o no estado APR.

1. Existir as seguintes referências no Projeto VBA:

- PRIMAVERA EXECUTIVE 8.00;
- PRIMAVERA GcpBE 8.00.

Apresenta-se de seguida o código que permite implementar a funcionalidade descrita.

```
Public Sub Transfere_Documento()
'Este procedimento pretende exemplificar a forma de efetuar uma transferência de
conta/estado
'através da extensibilidade. O exemplo descrito transfere um dos documentos pendentes
'para o fornecedor F0001 de um exercício DEMO com sede no continente.
'A transferência retira o documento do estado AGP e coloca-o no estado APR.

Dim objDocTrf As GcpBEDocumentoTransf
Dim ObjLinhaTrf As GcpBELinhaTransf
Dim strAvisos As String
Dim Mensagem As String

On Error GoTo Erro:

    Set objDocTrf = New GcpBEDocumentoTransf

    'Preenchimento do cabeçalho do documento de transferência
    With objDocTrf
        'Filial onde está a ser efetuada a transferência
        .Filial = BSO.Comercial.Filiais.CodigoFilial

        'Tipo de documento de transferência a utilizar
        .Tipodoc = "APR"

        'Série do documento de transferência a utilizar
```

```
.Serie = "2"

'Próximo numero de documento disponível para a série usada
.NumDoc = BSO.Comercial.Series.ProximoNumero("M", .Tipodoc, .Serie)

'Tipo de entidade para quem vai ser emitido o documento de transferência
'Neste exemplo um fornecedor
.TipoEntidade = "F"

'Entidade para quem vai ser emitido o documento de liquidação
.Entidade = "F0001"

'Tipo de entidade destino dos documentos transferidos
'Neste exemplo um fornecedor
.TipoEntidadeDestino = "F"

'Entidade destino dos documentos transferidos
.EntidadeDestino = "F0001"

'Tipo de Conta em que vão ser colocados os documentos transferidos
'Neste exemplo a conta-corrente de fornecedor
.TipoContaDestino = "CCF"

'Estado em que vão ser colocados os documentos transferidos
'Neste exemplo o estado Aprovado na conta-corrente de fornecedor
.EstadoDestino = "APR"

'Tipo de conta (Conta-Corrente)
.TipoConta = "C"

'Data da transferência (data atual)
.DataDoc = Date

'Moeda da transferência (tem de ser a mesma dos documentos a transferir)
.Moeda = "EUR"

'Modulo do cabealho do documento de Transferncia (contas-correntes)
.Modulo = "M"
End With

Set ObjLinhaTrf = New GcpBELinhaTransf

'Identificao do documento a transferir (criação de uma linha de transferência)
With ObjLinhaTrf
```

```
'Filial onde foi emitido o documento a transferir
.FilialOrig = "000"

'Mdulo a que pertence o documento (Compras)
.ModuloOrig = "C"

'Tipo de documento a transferir
.TipoDocOrig = "VFP"

'Srie do documento a transferir
.SerieOrig = "2"

'Numero do documento a transfeir
'Neste exemplo um documento (mdulo de compras)
'deve ser o numero de documento do fornecedor
.NumDocOrig = 4567

'Numero interno do documento a liquidar
'No caso de ser por exemplo um documento do módulo de compras
'podero no ser coincidentes
.NumDocOrigInt = 1

'Moeda em que foi emitido o documento a transferir
.MoedaDocOrig = "EUR"

'Tipo de Conta em que se encontra o documento a liquidar
'Neste exemplo conta-corrente de fornecedor
.TipoConta = "CCF"

'Estado em que se encontra o documento a transferir
'Neste exemplo o estado Aguarda Aprovao da conta-corrente de fornecedor
.Estado = "AGP"

'Numero da prestao do documento a transferir
'No caso da condio de pagamento do documento
'no ser particionada (prestaes) ser sempre 1
.NumPrestacaoOrig = 1

'Numero de Transferencia do documento
'Este valor poder ser diferente no caso do
'documento ter sofrido uma operação de transferência
'de conta/estado com alteração da entidade do documento
.NumTransferencia = 0

'No caso da transferência no ser uma transferência de entidade
```

```
.TransfereEntidade = False

'Valor a transferir (como se trata de um documento a pagar deve ser
negativo)
.ValorRec = -5808

'Valor original do documento (como se trata de um documento a pagar deve ser
negativo)
.ValorOrig = -5808

'Tipo de entidade original do documento
.TipoEntidadeOrigem = "F"

'Entidade Original do documento
.EntidadeOrigem = "F0001"
End With

'Associação da linha de transferência ao documento
objDocTrf.DocumentosTransferir.Insere ObjLinhaTrf

If BSO.Comercial.Transferencias.ValidaActualizacao(objDocTrf, strAvisos) Then
'Gravação do Documento
BSO.Comercial.Transferencias.Actualiza objDocTrf, strAvisos
PlataformaPRIMAVERA.Dialogos.MostraAviso "Documento transferido com sucesso.",
PRI_Informativo
Else
PlataformaPRIMAVERA.Dialogos.MostraErro "Erro", PRI_Exclama, strAvisos
End If

Set objDocTrf = Nothing
Set ObjLinhaTrf = Nothing
Exit Sub

Erro:
Set objDocTrf = Nothing
Set ObjLinhaTrf = Nothing
If Len(strAvisos) > 0 Then
PlataformaPRIMAVERA.Dialogos.MostraErro "Erro", PRI_Exclama, strAvisos
Else
PlataformaPRIMAVERA.Dialogos.MostraErro "Erro", PRI_Exclama, Err.Number & " -
" & Err.Description
End If
End Sub
```

## Criação de um documento de transferência bancária

Este exemplo pretende exemplificar a forma de criar um documento de tesouraria através da extensibilidade. O exemplo descrito cria uma transferência bancária assumindo estar a ser executado num exercício DEMO com sede no continente.

Para executar o exemplo abaixo devem-se verificar as seguintes condições:

1. Existir as seguintes referências no Projeto VBA:
  - PRIMAVERA EXECUTIVE 8.00;
  - PRIMAVERA GcpBE 8.00.

Apresenta-se de seguida o código que permite implementar a funcionalidade descrita.

```
Public Sub InsereDocTransferencia()  
'Este procedimento pretende exemplificar a forma de criar um documento de tesouraria  
'através da extensibilidade. O exemplo descrito cria uma transferência bancária  
assumindo  
'estar a ser executado num exercício DEMO com sede no continente.  
Dim objDocTesouraria As GcpBEDocumentoTesouraria  
Dim strErro As String  
  
On Error GoTo Erro  
  
Set objDocTesouraria = New GcpBEDocumentoTesouraria  
  
'Tipo do documento (TRANS - Transf. Bancária)  
objDocTesouraria.Tipodoc = "TRANS"  
  
'Preenchimento dos dados da transferência. Conta Origem, Conta Destino, Valor  
Set objDocTesouraria =  
BSO.Comercial.Tesouraria.PreencheDadosTransferencia(objDocTesouraria, "CXREC",  
"CXFIX", 200)  
  
'Gravação do Documento  
BSO.Comercial.Tesouraria.Actualiza objDocTesouraria, strErro  
  
Set objDocTesouraria = Nothing  
Exit Sub  
  
Erro:  
Set objDocTesouraria = Nothing  
If Len(strErro) > 0 Then  
PlataformaPRIMAVERA.Dialogos.MostraErro "Erro", PRI_Exclama, strErro
```

```
Else
    PlataformaPRIMAVERA.Dialogos.MostraErro "Erro", PRI_Exclama, Err.Number & " - "
    & Err.Description
End If
End Sub
```

### Criação de um documento de pagamento de cartão de crédito

Este exemplo pretende exemplificar a forma de criar um documento de tesouraria através da extensibilidade. O exemplo descrito cria um documento de pagamento de cartão de crédito assumindo estar a ser executado num exercício DEMO com sede no continente.

Para executar o exemplo abaixo devem-se verificar as seguintes condições:

1. Existir as seguintes referências no Projeto VBA:
  - PRIMAVERA EXECUTIVE 8.00;
  - PRIMAVERA GcpBE 8.00.

Apresenta-se de seguida o código que permite implementar a funcionalidade descrita.

```
Public Sub InsereDocPagamentoCartao()
    'Este procedimento pretende exemplificar a forma de criar um documento de tesouraria
    'através da extensibilidade. O exemplo descrito cria um documento de pagamento de
    'cartão
    'assumindo estar a ser executado num exercício DEMO com sede no continente.
    Dim objDocTesouraria As GcpBEDocumentoTesouraria
    Dim strErro As String

    On Error GoTo Erro

    Set objDocTesouraria = New GcpBEDocumentoTesouraria

    'Tipo do documento (PAG - Pagamento de Cartão)
    objDocTesouraria.Tipodoc = "PAG"

    'Preenchimento dos dados de pagamento. Conta Cartão, Conta DO, Valor
    Set objDocTesouraria =
    BSO.Comercial.Tesouraria.PreencheDadosPagamentoCartao(objDocTesouraria, "BPU-4",
    "BPU-1", 300)

    'Gravação do Documento
    BSO.Comercial.Tesouraria.Actualiza objDocTesouraria, strErro
    Set objDocTesouraria = Nothing
    Exit Sub
```

```
Erro:
  Set objDocTesouraria = Nothing
  If Len(strErro) > 0 Then
    PlataformaPRIMAVERA.Dialogos.MostraErro "Erro", PRI_Exclama, strErro
  Else
    PlataformaPRIMAVERA.Dialogos.MostraErro "Erro", PRI_Exclama, Err.Number & " - "
    & Err.Description
  End If
End Sub
```

### Criação de um documento de aplicação financeira

Este exemplo pretende exemplificar a forma de criar um documento de tesouraria através da extensibilidade. O exemplo descrito cria uma aplicação financeira assumindo estar a ser executado num exercício DEMO com sede no continente.

Para executar o exemplo abaixo devem-se verificar as seguintes condições:

1. Existir as seguintes referências no Projeto VBA:
  - PRIMAVERA EXECUTIVE 8.00;
  - PRIMAVERA GcpBE 8.00.

Apresenta-se de seguida o código que permite implementar a funcionalidade descrita.

```
Public Sub InsereDocAplicacaoFinanceira()
'Este procedimento pretende exemplificar a forma de criar um documento de tesouraria
'através da extensibilidade. O exemplo descrito cria uma aplicação financeira
'assumindo estar a ser executado num exercício DEMO com sede no continente.
Dim objDocTesouraria As GcpBEDocumentoTesouraria
Dim strErro As String

On Error GoTo Erro

  Set objDocTesouraria = New GcpBEDocumentoTesouraria

  'Tipo do documento (APL - Aplicação Financeira)
  objDocTesouraria.Tipodoc = "APL"

  'Preenchimento dos dados da aplicação. Conta Aplicação, Conta DO, Valor, Taxa de
  Juro, N de dias
  Set objDocTesouraria =
  BSO.Comercial.Tesouraria.PreencheDadosAplicacaoFinanceira(objDocTesouraria, "BPU-3",
  "BPU-1", 6000, 3.5, 60)
```

```
'Gravação do Documento
BSO.Comercial.Tesouraria.Actualiza objDocTesouraria, strErro
Set objDocTesouraria = Nothing
Exit Sub

Erro:
Set objDocTesouraria = Nothing
If Len(strErro) > 0 Then
    PlataformaPRIMAVERA.Dialogos.MostraErro "Erro", PRI_Exclama, strErro
Else
    PlataformaPRIMAVERA.Dialogos.MostraErro "Erro", PRI_Exclama, Err.Number & " - "
& Err.Description
End If
End Sub
```

### Criação de um documento de capitalização de aplicação financeira

Este exemplo pretende exemplificar a forma de criar um documento de tesouraria através da extensibilidade. O exemplo descrito cria um documento de capitalização de aplicação financeira assumindo estar a ser executado num exercício DEMO com sede no continente.

Para executar o exemplo abaixo devem-se verificar as seguintes condições:

1. Existir as seguintes referências no Projeto VBA:
  - PRIMAVERA EXECUTIVE 8.00;
  - PRIMAVERA GcpBE 8.00.

Apresenta-se de seguida o código que permite implementar a funcionalidade descrita.

```
Public Sub InsereDocCapitalizacaoAplFinanceira()
'Este procedimento pretende exemplificar a forma de criar um documento de tesouraria
'através da extensibilidade. O exemplo descrito cria um documento de capitalização
'de aplicação financeira assumindo estar a ser executado num exercício DEMO com sede
no continente.
Dim objDocTesouraria As GcpBEDocumentoTesouraria
Dim strErro As String
On Error GoTo Erro

Set objDocTesouraria = New GcpBEDocumentoTesouraria

'Tipo do documento (APL - Aplicação Financeira)
objDocTesouraria.Tipodoc = "CAPL"

'Preenchimento dos dados da capitalização. Conta Aplicação, Conta DO,
Identificação da aplicação a capitalizar
```

```

Set objDocTesouraria =
BSO.Comercial.Tesouraria.PreencheDadosCapitalizacaoAplFinanceira(objDocTesouraria,
"BPU-3", "BPU-1", "APL", "2", "000", 2, "DVC", "DVD")

'Gravação do documento
BSO.Comercial.Tesouraria.Actualiza objDocTesouraria, strErro
Set objDocTesouraria = Nothing
Exit Sub

Erro:
Set objDocTesouraria = Nothing
If Len(strErro) > 0 Then
    PlataformaPRIMAVERA.Dialogos.MostraErro "Erro", PRI_Exclama, strErro
Else
    PlataformaPRIMAVERA.Dialogos.MostraErro "Erro", PRI_Exclama, Err.Number & " - "
& Err.Description
End If
End Sub

```

### Criação de um novo pendente

Este exemplo pretende exemplificar a forma de criar um novo pendente em conta-corrente através da extensibilidade. O exemplo descrito cria um documento de tipo N/Fatura (série 2) para o cliente SOFRIO, assumindo estar a ser executado num exercício DEMO com sede no continente.

Para executar o exemplo abaixo devem-se verificar as seguintes condições:

- Existir as seguintes referências no Projeto VBA:
  - PRIMAVERA EXECUTIVE 8.00;
  - PRIMAVERA GcpBE 8.00.

Apresenta-se de seguida o código que permite implementar a funcionalidade descrita.

```

Public Sub InsereNovoPendente()
'Este procedimento pretende exemplificar a forma de criar um novo pendente em conta-
corrente
'através da extensibilidade. O exemplo descrito cria um documento de tipo N/Fatura
(série 2)
'para o cliente SOFRIO, assumindo estar a ser executado num exercício DEMO com sede
no continente.
Dim objPendente           As GcpBEPendente
Dim objLinPendente       As GCPBELinhaPendente
Dim dblTaxaIva           As Double
Dim strErro              As String

```

```
On Error GoTo Erro
```

```
Set objPendente = New GcpBEPendente
```

```
With objPendente
```

```
    'Tipo de Entidade do Pendente (Cliente)
```

```
    .TipoEntidade = "C"
```

```
    'Entidade do Pendente
```

```
    .Entidade = "SOFRIO"
```

```
    'Tipo de Documento do Pendente
```

```
    .Tipodoc = "FAC"
```

```
End With
```

```
'Preenchimento de todos os dados em falta
```

```
BSO.Comercial.pendentes.PreencheDadosRelacionados objPendente, pdDadosTodos
```

```
Set objLinPendente = New GCPBELinhaPendente
```

```
'Preenchimento dos dados da linha do pendente
```

```
With objLinPendente
```

```
    'Descrição da linha
```

```
    .Descricao = "Comisses"
```

```
    'Valor da linha (sem IVA)
```

```
    .Incidencia = 755
```

```
    'Código da taxa de IVA aplicar na linha
```

```
    .CodIva = "21"
```

```
    'Taxa de IVA associada ao código
```

```
    dblTaxaIva = BSO.Comercial.Iva.DaValorAtributo("21", "Taxa")
```

```
    'Cálculo do Valor de IVA
```

```
    .ValorIva = (dblTaxaIva / 100) * objLinPendente.Incidencia
```

```
    'Cálculo do Total da linha
```

```
    .Total = objLinPendente.Incidencia + objLinPendente.ValorIva
```

```
End With
```

```
'Associação da Linha ao documento
```

```
objPendente.Linhas.Insere objLinPendente
```

```
'Gravação do Documento
BSO.Comercial.pendentes.ActualizaEx objPendente, strErro

Set objPendente = Nothing
Set objLinPendente = Nothing
Exit Sub

Erro:
Set objPendente = Nothing
Set objLinPendente = Nothing
If Len(strErro) > 0 Then
    PlataformaPRIMAVERA.Dialogos.MostraErro "Erro", PRI_Exclama, strErro
Else
    PlataformaPRIMAVERA.Dialogos.MostraErro "Erro", PRI_Exclama, Err.Number & " - "
& Err.Description
End If
End Sub
```

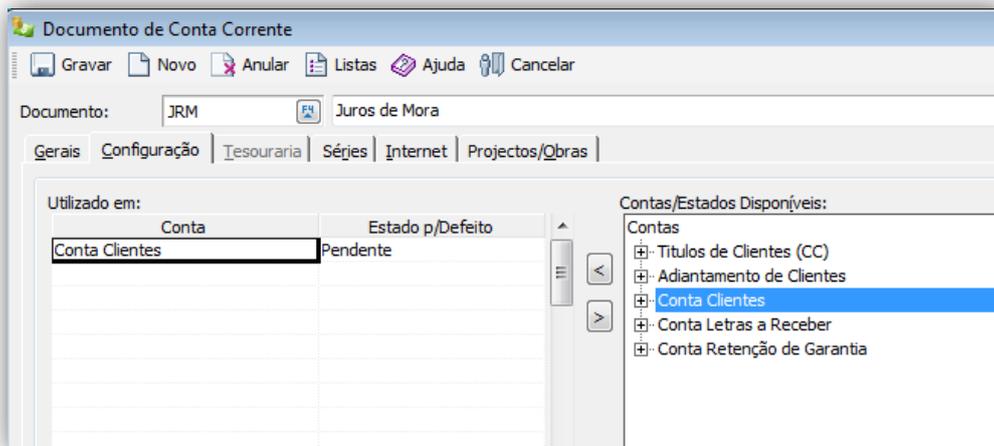
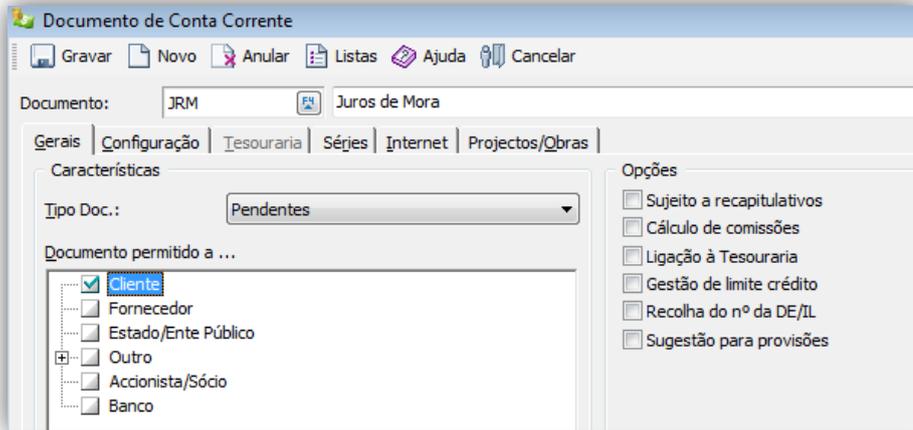
### Inserção de um pendente de juros de mora

Com o exemplo seguinte pretende-se realizar as seguintes operações:

- 1) Verificar se a data de vencimento dos pendentes que estão a ser liquidados foi ultrapassada em 30 dias;
- 2) Caso a condição anterior se verifique inserir um pendente com um valor de 1% dos valores liquidados com a data de vencimento ultrapassada em 30 dias. Para esta implementação vai ser usado código no evento DepoisDeGravar das operações sobre contas-correntes.

Para executar o exemplo abaixo devem-se verificar as seguintes condições:

1. Existir as seguintes referências no Projeto VBA:
  - PRIMAVERA EXECUTIVE 8.00;
  - PRIMAVERA GcpBE 8.00.
2. Existir um documento de conta corrente com as seguintes configurações:





```
        dblValor = dblValor +  
Me.DocumentoLiquidacao.LinhasLiquidacao(lngLinha).ValorRec  
        End If  
    Next  
  
    'Se existe valor para cálculo de juros inserir um pendente  
    If Valor > 0 Then InserePendenteJuros Me.DocumentoLiquidacao.Entidade,  
dblValor  
        End If  
    End If  
End Sub  
  
Private Sub InserePendenteJuros(Entidade As String, Valor As Double)  
Dim objPendente        As GcpBEPendente  
Dim objLinPendente     As GCPBELinhaPendente  
Dim dblTaxaIva         As Double  
Dim strErro            As String  
  
On Error GoTo Erro  
  
    Set objPendente = New GcpBEPendente  
  
    With objPendente  
        'Tipo de Entidade do Pendente (Cliente)  
        .TipoEntidade = "C"  
  
        'Entidade do Pendente  
        .Entidade = Entidade  
  
        'Tipo de Documento do Pendente  
        .Tipodoc = "JRM"  
    End With  
  
    'Preenchimento de todos os dados em falta  
    BSO.Comercial.pendentes.PreencheDadosRelacionados objPendente, pdDadosTodos  
  
    Set objLinPendente = New GCPBELinhaPendente  
  
    'Preenchimento dos dados da linha do pendente  
    With objLinPendente  
        'Descrição da linha  
        .Descricao = "Juros de Mora"  
  
        'Valor da linha (sem IVA)  
        .Incidencia = 0.01 * Valor
```

```

        'Código da taxa de IVA aplicar linha (0%)
        .CodIva = "00"

        'Valor de IVA
        .ValorIva = 0

        'Cálculo do Total da linha
        .Total = 0.01 * Valor
    End With

    'Associação da linha ao documento
    objPendente.Linhas.Insere objLinPendente

    'Gravação do Documento
    BSO.Comercial.pendentes.ActualizaEx objPendente, strErro

    Set objPendente = Nothing
    Set objLinPendente = Nothing
    Exit Sub

Erro:
    Set objPendente = Nothing
    Set objLinPendente = Nothing
    If Len(strErro) > 0 Then
        PlataformaPRIMAVERA.Dialogos.MostraErro "Erro", PRI_Exclama, strErro
    Else
        PlataformaPRIMAVERA.Dialogos.MostraErro "Erro", PRI_Exclama, Err.Number & " -
" & Err.Description
    End If
End Sub
    
```

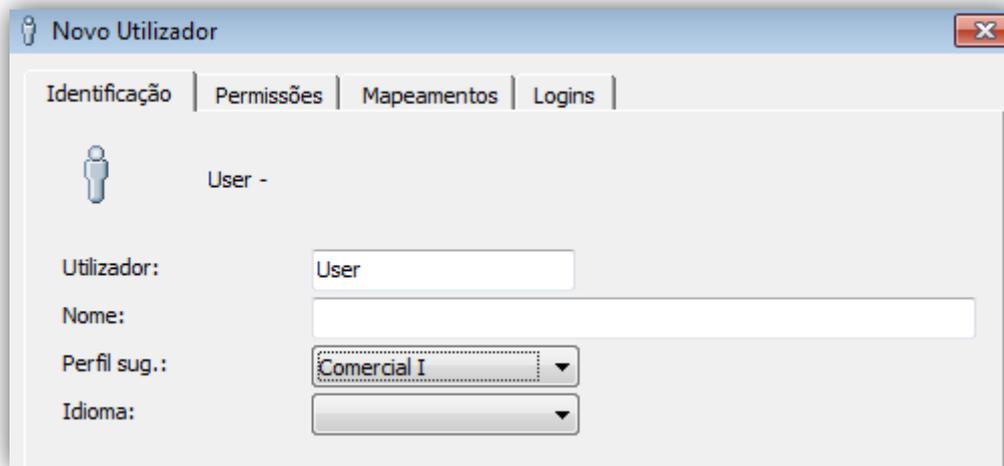
### Restrição às liquidações de uma determinada entidade a um determinado utilizador

Pretende-se exemplificar através do uso do método EntidadeIdentificada das operações sobre contas-correntes, a forma de restringir o acesso às liquidações de uma determinada entidade. Assumindo estar a ser executado num exercício DEMO com sede no continente, este exemplo permite que as liquidações ao cliente SOFRIO apenas possam ser efetuadas pelo utilizador "User".

Para executar o exemplo abaixo devem-se verificar as seguintes condições:

1. Existir as seguintes referências no Projeto VBA:
  - PRIMAVERA EXECUTIVE 8.00;
  - PRIMAVERA GcpBE 8.00.

Existir um utilizador conforme ilustrado:



Apresenta-se de seguida o código que permite implementar a funcionalidade descrita.

```
Private Sub EditorCCorrentes_EntidadeIdentificada(TDocumento As
GcpBE800.TE_DocCCorrentes, TipoEntidade As String, Entidade As String, Cancel As
Boolean)
'Pretende-se implementar as seguintes restrições:
'
'    1. As liquidações ao cliente SOFRIO apenas poderão ser efetuadas por um
determinado utilizador.
'Este exemplo descreve a forma de o fazer. Assume-se que a operação está a ser
efetuada num exercício DEMO
'com sede no continente e que existem utilizadores registados no sistema.

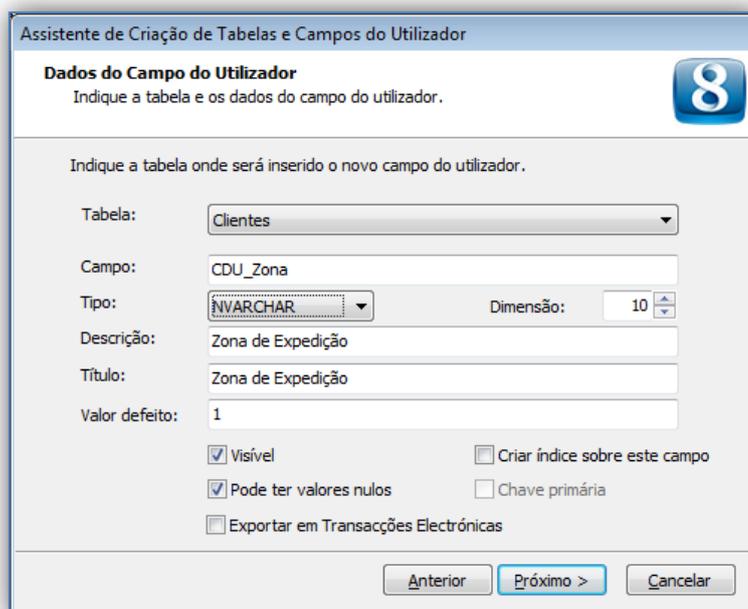
    If TDocumento = TE_DocLiquidacao Then
        If Aplicacao.Utilizador.Utilizador <> "User" And TipoEntidade = "C" And
Entidade = "ALCAD" Then
            Cancel = True
        End If
    End If
End Sub
```

## Preenchimento de um campo de utilizador nas liquidações

Pretende-se exemplificar usando o método AntesDeGravar das operações sobre contas-correntes, a forma de efetuar o preenchimento de forma automática de um campo de utilizador no cabeçalho do documento de liquidação. Esse campo vai identificar a zona para onde deve ser enviada a liquidação a partir de um campo de utilizador da tabela de clientes.

Para executar o exemplo abaixo devem-se verificar as seguintes condições:

1. Existir as seguintes referências no Projeto VBA:
  - PRIMAVERA EXECUTIVE 8.00;
  - PRIMAVERA GcpBE 8.00.
  
2. Existir um campo de utilizador na tabela de clientes conforme ilustrado:



3. Existir um campo de utilizador na tabela cabliq conforme ilustrado:

Assistente de Criação de Tabelas e Campos do Utilizador

**Dados do Campo do Utilizador**  
Indique a tabela e os dados do campo do utilizador.

Indique a tabela onde será inserido o novo campo do utilizador.

Tabela: CabLiq

Campo: CDU\_Zona

Tipo: NVARCHAR Dimensão: 10

Descrição: Zona de Expedição

Título: Zona de Expedição

Valor defeito:

Visível  Criar índice sobre este campo

Pode ter valores nulos  Chave primária

Exportar em Transacções Electrónicas

Anterior Próximo > Cancelar

Apresenta-se de seguida o código que permite implementar a funcionalidade descrita.

```
Private Sub EditorCorrentes_AntesDeGravar(TDocumento As GcpBE800.TE_DocCorrentes,
Cancel As Boolean)
Dim strZona As String

If TDocumento = TE_DocLiquidacao Then
If Me.DocumentoLiquidacao.Tipodoc = "RE" And
Me.DocumentoLiquidacao.TipoEntidade = "C" Then
strZona =
BSO.Comercial.Clientes.DaValorAtributo(Me.DocumentoLiquidacao.Entidade, "CDU_Zona")
Me.DocumentoLiquidacao.CamposUtil("CDU_Zona") = strZona
End If
End If
End Sub
```

### Restrição à edição de novos pendentes para uma determinada entidade a um determinado utilizador

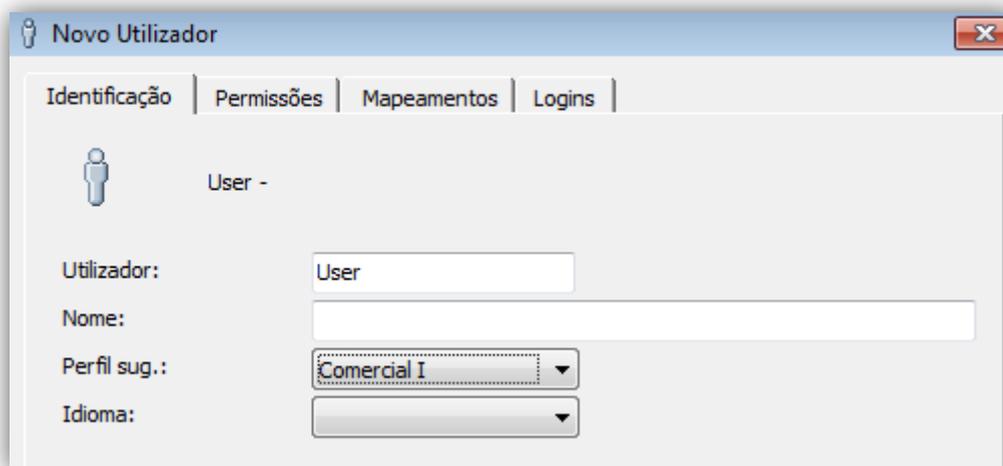
Pretende-se exemplificar através do uso do método AntesDeEditar dos novos documentos de conta-corrente, a forma de restringir o acesso para edição aos documentos de uma determinada entidade. Assumindo estar a ser executado num exercício DEMO com sede no continente, este exemplo permite que a edição dos documentos do cliente SOFRIO apenas possam ser efetuadas pelo utilizador "User".

Para executar o exemplo abaixo devem-se verificar as seguintes condições:

1. Existir a seguinte referência no Projeto VBA:

- PRIMAVERA EXECUTIVE 8.00.

- Existir um utilizador conforme ilustrado:



Apresenta-se de seguida o código que permite implementar a funcionalidade descrita.

```
Private Sub EditorPendentes_AntesDeEditar(Filial As String, Serie As String, Tipo As String, NumDoc As Long, NumPrestacao As Integer, Cancel As Boolean)
    If Aplicacao.Utilizador.Utilizador <> "User" And Me.DocumentoPendente.TipoEntidade = "C" And Me.DocumentoPendente.Entidade = "SOFRIO" Then
        Cancel = True
    End If
End Sub
```

### Adicionar uma linha fixa aos novos pendentes

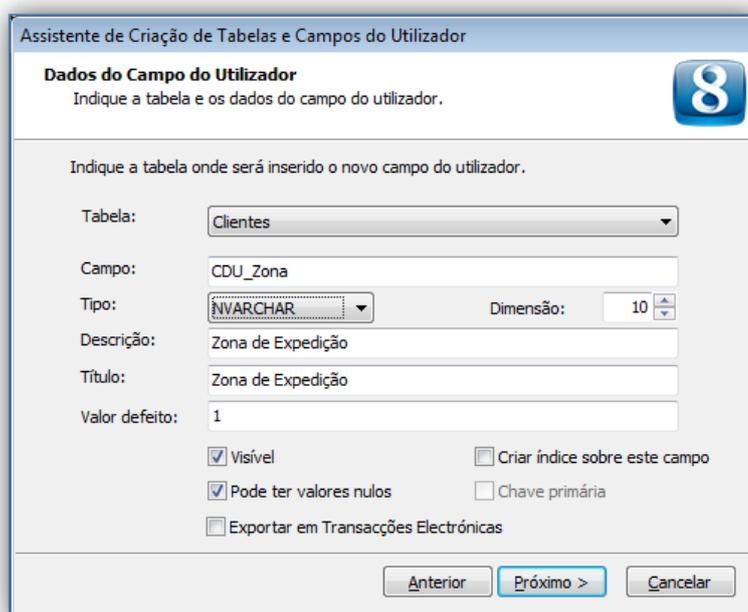
Pretende-se exemplificar usando o método AntesDeGravar dos novos documentos de conta corrente, a forma de efetuar o preenchimento de forma automática de uma linha fixa a determinados documentos (FAC) de conta corrente para entidades do tipo cliente. O valor para essa linha vai ser inserido tendo em conta o valor existente num campo do utilizador da tabela de clientes, assumindo estar a ser executado num exercício DEMO com sede no continente. Essa linha pretenderá simular uma situação de adicionar despesas de envio ao valor da fatura.

Para executar o exemplo abaixo devem-se verificar as seguintes condições:

- Existir as seguintes referências no Projeto VBA:

- PRIMAVERA EXECUTIVE 8.00;
- PRIMAVERA GcpBE 8.00.

2. Existir um campo de utilizador na tabela de clientes conforme ilustrado:



Apresenta-se de seguida o código que permite implementar a funcionalidade descrita.

```
Private Sub EditorPendentes_AntesDeGravar(Cancel As Boolean)
Dim dblTaxaIva           As Double
Dim intLinha            As Integer
Dim intZona              As Integer
Dim objLinhaDespesa     As GCPBELinhaPendente
Dim dblIncidencia       As Double

On Error GoTo Erro:

    If Me.DocumentoPendente.TipoEntidade = "C" And Me.DocumentoPendente.Tipodoc =
    "FAC" And Not Me.DocumentoPendente.EmModoEdicao Then

        'Determinação da zona do cliente
        intZona =
        BSO.Comercial.Clientes.DaValorAtributo(Me.DocumentoPendente.Entidade, "CDU_Zona")

        Select Case Zona
```

```
Case 1:
    dblIncidencia = 100
Case 2:
    dblIncidencia = 125
Case 3:
    dblIncidencia = 150
End Select

Set objLinhaDespesa = New GCPBELinhaPendente

'Descrição da Linha
objLinhaDespesa.Descricao = "Despesas de Envio"

objLinhaDespesa.Incidencia = dblIncidencia

'Cálculo do Valor de Iva (taxa de 21%)
objLinhaDespesa.ValorIva = 0.21 * dblIncidencia
objLinhaDespesa.CodIva = "21"

'Cálculo do Valor Total (Incidência + Iva)
objLinhaDespesa.Total = dblIncidencia + objLinhaDespesa.ValorIva

'Associação da linha ao documento
Me.DocumentoPendente.Linhas.Insere objLinhaDespesa
Set objLinhaDespesa = Nothing

End If
Exit Sub

Erro:
    Set objLinhaDespesa = Nothing
    PlataformaPRIMAVERA.Dialogos.MostraErro "Erro", PRI_Exclama, Err.Number & " - " &
Err.Description
End Sub
```

O resultado será a inserção de uma linha conforme ilustrado:

Descritivo	Incidência	IVA	Taxa	Valor IVA	Total	%IVA Ded.
Linha	7.800,00	21	21,00	1.638,00	9.438,00	100,00
Despesas de Envio	125,00	21	21,00	26,25	151,25	0,00

### Informar o crédito disponível na inserção de documentos

Pretende-se exemplificar o uso do método `EntidadeIdentificada` dos novos documentos de conta corrente. No exemplo através do uso desse método informa-se o valor de crédito disponível quando parametrizado para entidades do tipo cliente ou fornecedor.

Para executar o exemplo abaixo devem-se verificar as seguintes condições:

- Existir as seguintes referências no Projeto VBA:
  - PRIMAVERA EXECUTIVE 8.00;
  - PRIMAVERA StdBE 8.00.

Apresenta-se de seguida o código que permite implementar a funcionalidade descrita.

```
Private Sub EditorPendentes_EntidadeIdentificada(TipoEntidade As String, Entidade As String, Cancel As Boolean)
    Dim objCampos As StdBECampos
    Dim strMensagem As String
    Dim dblCredito As Double

    On Error GoTo Erro:

    Set objCampos = New StdBECampos
    strMensagem = vbNullString

    Select Case TipoEntidade
        Case "C":
            Set objCampos = BSO.Comercial.Clientes.DaValorAtributos(Entidade, "LimiteCredValor", "LimiteCred", "TotalDeb")
            If Not objCampos Is Nothing Then
                If objCampos("LimiteCredValor") Then
                    dblCredito = objCampos("LimiteCred") - objCampos("TotalDeb")
                    If dblCredito > 0 Then
                        strMensagem = "O crédito disponível para esta entidade é de " & CStr(dblCredito) & " EUR."
                    End If
                End If
            End If
    End Select
End Sub
```

```
Else
    strMensagem = "A entidade já não tem crédito disponível."
End If
End If
End If

Case "F":
    Set objCampos =
BSO.Comercial.Fornecedores.DaValorAtributos(Entidade, "LimiteCredValor",
"LimiteCred", "TotalDeb")
    If Not objCampos Is Nothing Then
        If objCampos("LimiteCredValor") Then
            dblCredito = objCampos("LimiteCred") - objCampos("TotalDeb")
            If dblCredito < 0 Then
                strMensagem = "O crédito disponível para esta entidade é de
" & CStr(Abs(dblCredito)) & " EUR."
            Else
                strMensagem = "A entidade já não tem crédito disponível."
            End If
        End If
    End If
End If

End Select

If Len(strMensagem) > 0 Then PlataformaPRIMAVERA.Dialogos.MostraAviso
strMensagem, PRI_Informativo

Exit Sub

Erro:
Set objCampos = Nothing
PlataformaPRIMAVERA.Dialogos.MostraErro "Erro", PRI_Exclama, Err.Number & " - " &
Err.Description
End Sub
```

### Limitar o valor da transferência bancária

Pretende-se exemplificar o uso do método AntesDeGravar dos novos documentos de caixa/bancos para impedir a gravação dos documentos do tipo transferência bancária sempre que excedam um determinado valor.

Para executar o exemplo abaixo devem-se verificar as seguintes condições:

1. Existir as seguintes referências no Projeto VBA:
  - PRIMAVERA EXECUTIVE 8.00;
  - PRIMAVERA GcpBE 8.00.

Apresenta-se de seguida o código que permite implementar a funcionalidade descrita.

```
Private Sub EditorTesouraria_AntesDeGravar(TDocumento As GcpBE800.TE_DocTesouraria,
Cancel As Boolean)
    If TDocumento = TES_Transferencia Then
        If Me.DocumentoTesouraria.ValorMovimento >= 10000 And
Me.DocumentoTesouraria.Moeda = "EUR" Then
            PlataformaPRIMAVERA.Dialogos.MostraAviso "Não é possível efetuar
transferências de valor superior a 10.000 EUR", PRI_Informativo
            Cancel = True
        End If
    End If
End Sub
```

### Adicionar documento de despesas bancárias sempre que se efetua uma transferência

Pretende-se exemplificar o uso do método AntesDeGravar dos novos documentos de caixa/bancos para impedir a gravação dos documentos do tipo transferência bancária sempre que excedam um determinado valor.

Para executar o exemplo abaixo devem-se verificar as seguintes condições:

1. Existir as seguintes referências no Projeto VBA:
  - PRIMAVERA EXECUTIVE 8.00;
  - PRIMAVERA GcpBE 8.00.

Apresenta-se de seguida o código que permite implementar a funcionalidade descrita.

```
Private Sub EditorTesouraria_DepoisDeGravar(TDocumento As GcpBE800.TE_DocTesouraria,
Filial As String, Tipodoc As String, Serie As String, NumDoc As Long)
    If TDocumento = TES_Transferencia Then
        InsereDocumentoDespesa Me.DocumentoTesouraria.ContaOrigem
    End If
End Sub

Private Sub InsereDocumentoDespesa(Conta As String)
Dim objDocTesouraria      As New GcpBEDocumentoTesouraria
Dim objLinTes             As New GcpBELinhaDocTesouraria
Dim strErro               As String

    Set objDocTesouraria = New GcpBEDocumentoTesouraria

    'Filial onde está a ser efetuada a operação
    objDocTesouraria.Filial = "000"

    'Tipo de documento (Movimento em conta)
    objDocTesouraria.Tipodoc = "MOV"

    'Atribuição da série usado por defeito
    objDocTesouraria.Serie = BSO.Comercial.Series.DaSerieDefeito("B", "MOV")

    'Data documento de tesouraria (data atual)
    objDocTesouraria.Data = Date

    'Conta a debitar
    objDocTesouraria.ContaOrigem = Conta

    'Moeda da operação
    objDocTesouraria.Moeda = "EUR"

    'Cambios
    objDocTesouraria.Cambio = 1
    objDocTesouraria.CambioMAlt = 1
    objDocTesouraria.CambioMBase = 1

    'Módulo que gerou o documento (bancos)
    objDocTesouraria.ModuloOrigem = "B"

    'Criação da linha de tesouraria
    Set objLinTes = New GcpBELinhaDocTesouraria

    'Identificação do movimento bancário
    objLinTes.MovimentoBancario = "DSB"
```

```
'Conta a movimentar
objLinTes.Conta = Conta

'Moeda do movimento
objLinTes.Moeda = "EUR"

'Valor a debitar
objLinTes.Debito = 1.5

'Cambios
objLinTes.Cambio = 1
objLinTes.CambioMAlt = 1
objLinTes.CambioMBase = 1

'Data do movimento
objLinTes.DataMovimento = Date

'Data Valor
objLinTes.DataValor = Date

'Natureza do movimento
objLinTes.Natureza = "D"

'Associação da linha ao documento
objDocTesouraria.Linhas.Insere objLinTes

If BSO.Comercial.Tesouraria.ValidaActualizacao(objDocTesouraria, strErro) Then
  'Gravação do documento
  BSO.Comercial.Tesouraria.Actualiza objDocTesouraria, strErro
End If
End Sub
```

## Área de Contabilidade

### Criação de um lançamento na contabilidade com reflexão para centros de custo

Este exemplo pretende exemplificar a forma de efetuar um lançamento para a contabilidade através da extensibilidade. O exemplo descrito faz um lançamento também uma reflexão para centros de custo.

Para executar o exemplo abaixo devem-se verificar as seguintes condições:

1. Existir as seguintes referências no Projeto VBA:
  - PRIMAVERA EXECUTIVE 8.00;
  - PRIMAVERA CbIBE 8.00.

Apresenta-se de seguida o código que permite implementar a funcionalidade descrita.

```
Public Sub InserirMovimentoContabilidade()  
  
Dim objDoc As CblBEDocumento  
Dim objLinhasDoc As CblBELinhaDocGeral  
Dim objLinhasCentro As CblBELinhaDocCentros  
Dim strErro As String  
  
'Criação do cabeçalho do documento  
Set objDoc = New CblBEDocumento  
With objDoc  
    .Modulo = "L"  
    .Doc = "411"  
    .NumDoc = -1  
    .Diario = "41"  
    .NumDiario = -1  
    .Ano = 2009  
    .Mes = DatePart("m", Date)  
    .Dia = DatePart("d", Date)  
    .Moeda = "EUR"  
End With  
  
Set objLinhasDoc = New CblBELinhaDocGeral  
  
'Atribuição dos valores da 1ª linha  
With objLinhasDoc  
    .Conta = "22110001"  
    .Descricao = "Fornecedor 1"  
    .Natureza = "C"
```

```
.TipoLinha = "F"  
.Valor = 100000  
.ValorAlt = 100000  
.ValorOrigem = 100000  
.Lote = 0  
.Cambio = 1  
.CambioMAlt = 1  
.CambioOrigem = 1  
objDoc.LinhasGeral.Insere objLinhasDoc  
End With  
Set objLinhasDoc = Nothing  
  
Set objLinhasDoc = New CblBELinhaDocGeral  
  
'Atribuição dos valores da 2ª linha  
With objLinhasDoc  
  .Conta = "24321110511"  
  .Descricao = "Iva"  
  .Natureza = "D"  
  .TipoLinha = "F"  
  .Valor = 4762  
  .ValorAlt = 4762  
  .ValorOrigem = 4762  
  .Lote = 0  
  .Cambio = 1  
  .CambioMAlt = 1  
  .CambioOrigem = 1  
  objDoc.LinhasGeral.Insere objLinhasDoc  
End With  
Set objLinhasDoc = Nothing  
  
Set objLinhasDoc = New CblBELinhaDocGeral  
'Atribuição dos valores da 3ª linha  
With objLinhasDoc  
  .Conta = "31611"  
  .Descricao = "Compras-Mat.Prim.-Mercado Nac."  
  .Natureza = "D"  
  .TipoLinha = "F"  
  .Valor = 95238  
  .ValorAlt = 95238  
  .ValorOrigem = 95238  
  .Iva = "121000111"  
  .Lote = 0  
  .Cambio = 1  
  .CambioMAlt = 1
```

```
.CambioOrigem = 1
objDoc.LinhasGeral.Insere objLinhasDoc
End With
Set objLinhasDoc = Nothing

' Criação de uma linha para relaxação em centros de custo
Set objLinhasCentro = New CblBELinhaDocCentros

'Atribuição dos valores para relaxação
With objLinhasCentro
    .Centro = "110030"
    .ContaOrigem = "31611"
    .Descricao = "Reflexão"
    .Natureza = "D"
    .TipoLinha = "0"
    .Valor = 95238
    .ValorAlt = 95238
    .ValorOrigem = 95238
    .Lote = 0
    .Cambio = 1
    .CambioMAlt = 1
    .CambioOrigem = 1
objDoc.LinhasCentros.Insere objLinhasCentro
End With
Set objLinhasDoc = Nothing

BSO.Contabilidade.Documentos.Actualiza objDoc, strErro

PlataformaPRIMAVERA.Dialogos.MostraAviso "Foi gravado o Numero de Diário " &
objDoc.NumDiario & " no diário " & objDoc.Diario, PRI_Informativo
Set objDoc = Nothing
Set objLinhasDoc = Nothing
Exit Sub

Erro:
Set objDoc = Nothing
Set objLinhasDoc = Nothing
If Len(strErro) > 0 Then
    PlataformaPRIMAVERA.Dialogos.MostraErro "Erro", PRI_Exclama, strErro
Else
    PlataformaPRIMAVERA.Dialogos.MostraErro "Erro", PRI_Exclama, Err.Number & " -
" & Err.Description
End If
End Sub
```



designed  
for excellence

