

## **Estrutura da aula**

### **Noções básicas sobre macros no Access 97**

Construção de algumas macros simples

### **Noções básicas sobre procedimentos VBA no Access 97**

Exemplo de um procedimento VBA

### **Apresentação e discussão de propostas para os trabalhos de grupo**

**Macros:** conjunto de instruções simples que permitem automatizar respostas a determinados acontecimentos

- Abrir e fechar tabelas, consultas, formulários e relatórios
- Imprimir dados
- Executar uma consulta
- Testar condições e controlar o fluxo da acção
- Definir valores
- Procurar dados
- Construir um menu personalizado e executar comandos de menus
- Controlar a apresentação e o foco
- Informar o utilizador sobre as acções
- Mudar o nome, copiar, eliminar, guardar, importar e exportar objectos
- Executar outra aplicação

Fazendo duplo clique no nome de um titular será possível visualizar a informação completa sobre esse cliente

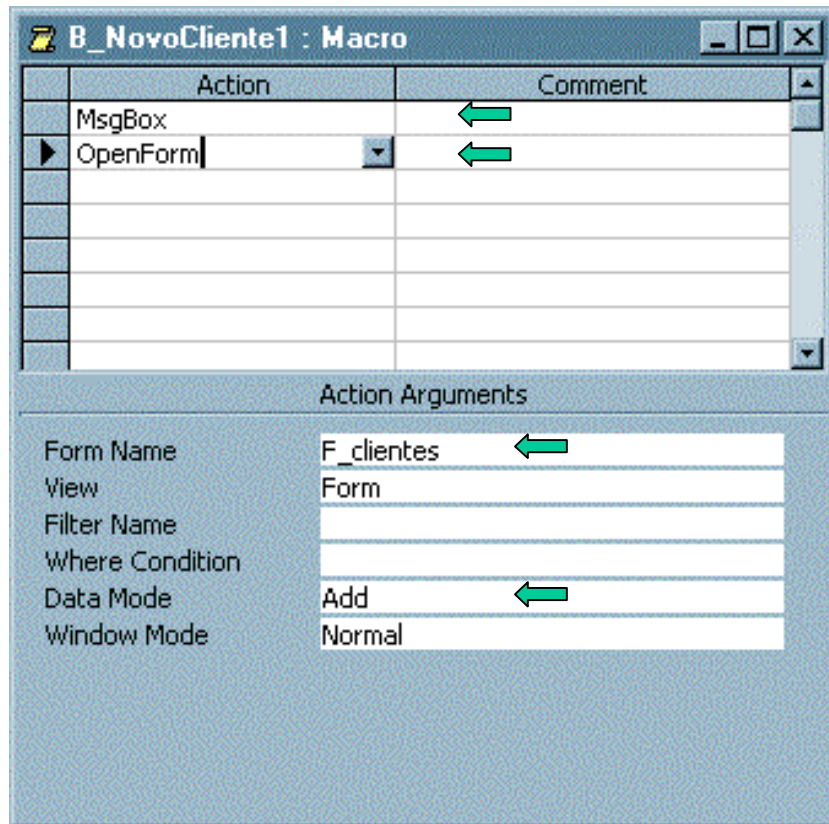
**Botão Movimentos:** permite visualizar os movimentos efectuados na conta actual

**Botão Novo cliente:** permite acrescentar um novo cliente, abindo o formulário F\_Cliente

### Criação do botão de comando Botao\_NovoCliente

The image shows the MS Access 97 interface during the creation of a command button. The main window displays a form titled "F\_Contas\_Botao : Form" with a grid layout. The grid includes a "Form Header" with the title "Contas", a "Detail" section with fields for "Nº da Conta" (containing "#Conta") and "Tipo" (containing "#Tipo"), and a "Footers" section with buttons for "Movimentos" and "Novo cliente". A "Toolbox" is open, showing various controls, with an arrow pointing to the "Command Button" icon. Three dialog boxes are overlaid on the form:

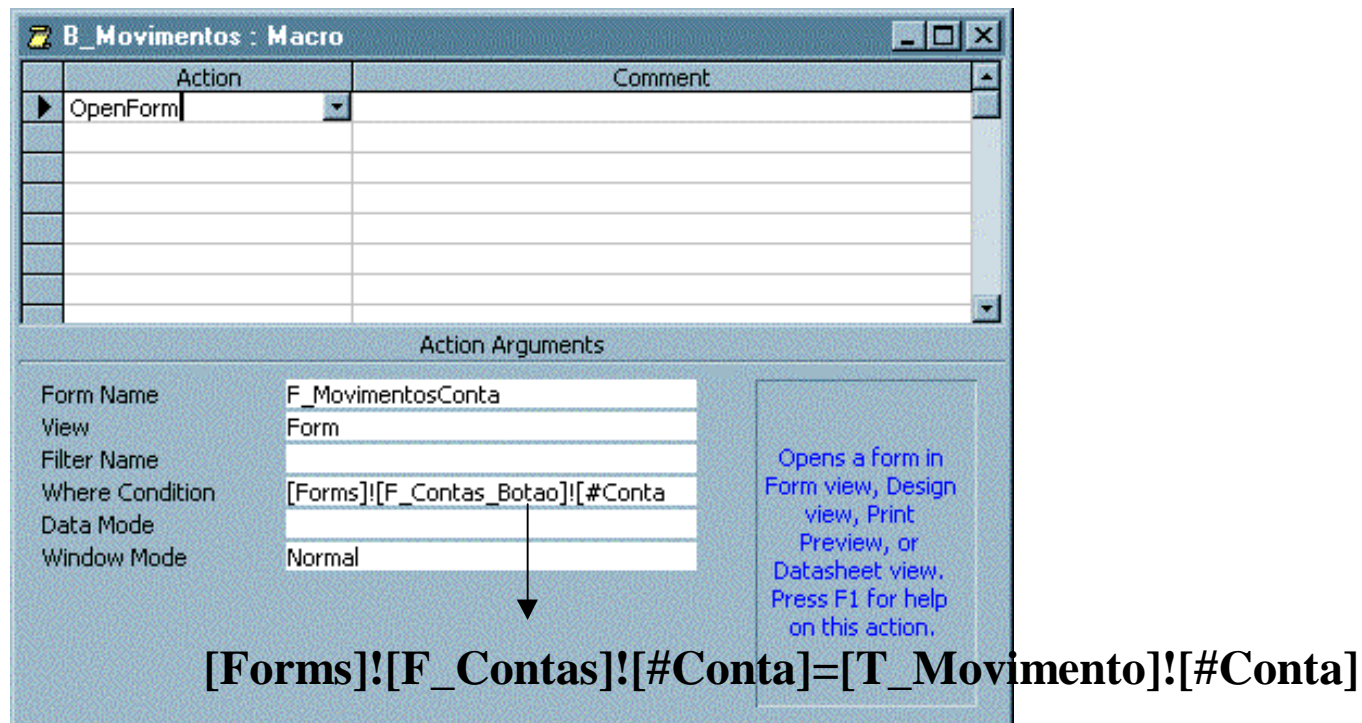
- Command Button: Botao\_NovoCliente (Format tab):** Shows properties for the button, including Name: Botao\_NovoCliente, Caption: Novo cliente, Picture: (none), Picture Type: Embedded, Transparent: No, Default: No, and Cancel: No. Green arrows point to the Name and Caption fields.
- Command Button: Botao\_NovoCliente (Event tab):** Shows event properties for the button, including On Enter, On Exit, On Got Focus, On Lost Focus, On Click, On Dbl Click, and On Mouse Down. An arrow points to the On Click event field.
- Choose Builder:** A dialog box with a list box containing "Expression Builder", "Macro Builder", and "Code Builder". The "Macro Builder" option is selected. Arrows point to the "OK" and "Cancel" buttons.

Macro associada ao botão **Botao\_NovoCliente**

- Mensagem a avisar que um novo cliente vai ser introduzido
- Escolher a acção pretendida: abrir um formulário
- Seleccionar o formulário F\_clientes
- Seleccionar o modo de introdução de novos dados

### Para o botão **Botao\_Movimentos**:

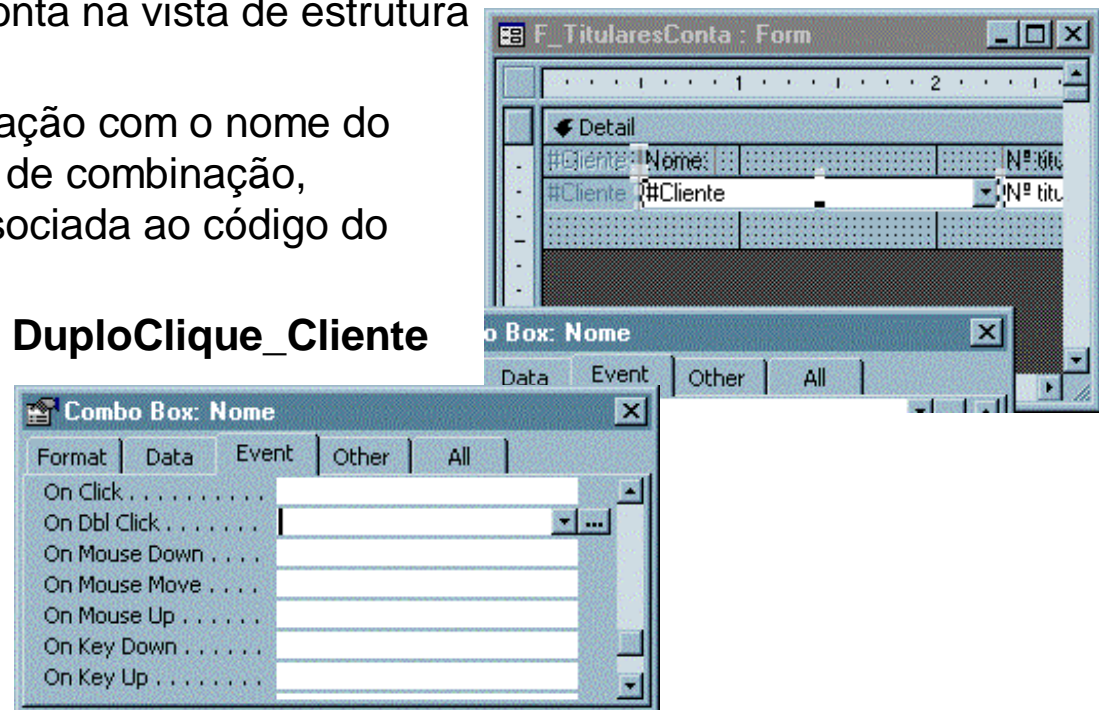
1. É necessário um formulário com os movimentos (se não existir, crie um muito simples, baseado na tabela **T\_Movimentos**)
2. Crie o botão **Botao\_Movimentos**
3. No acontecimento “*On click*” crie uma nova macro e dê-lhe o nome **B\_Movimentos**



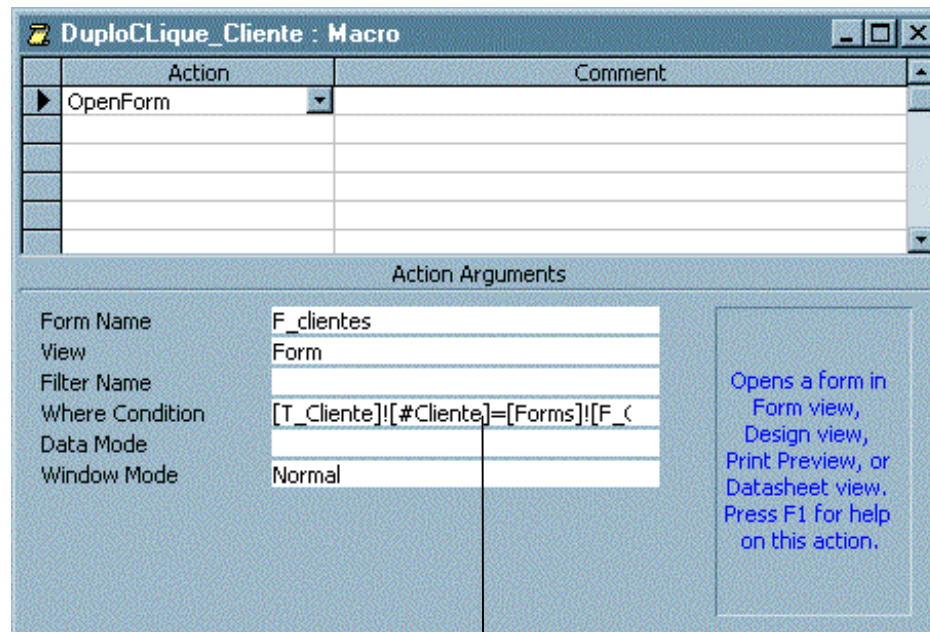
Macro associada ao acontecimento “***On Dbl Click***” (duplo clique) no nome de um titular do formulário **F\_Contas**

**Nota:** O nome dos titulares encontra-se num subformulário, logo a macro deve ser criada dentro desse subfromulário

1. Abrir o formulário F\_TitularesConta na vista de estrutura
2. Seleccionar a Caixa de combinação com o nome do titular ( Se não tiver uma caixa de combinação, seleccione a caixa de texto associada ao código do titular).
3. Crie uma nova macro chamada **DuploClique\_Cliente**



## Macro DuploClique\_Clientes



[T\_Cliente]![#Cliente]=[Forms]![F\_Contas]![Titulares].[Form]![#Cliente]



**Deve utilizar macros quando:**

- Não necessitar de tratar erros
- Não necessitar de avaliar ou definir parâmetro passados por acontecimentos, tais como *Confirmar*, *Confirmar eliminação*, *Erro*, *Se não estiver na lista*, etc.
- A aplicação contiver poucos formulários e relatórios
- A aplicação se destinar a utilizadores não programadores (mas que gostariam de a poder alterar e melhorar)
- Estiver a desenvolver um protótipo de uma aplicação e pretender automatizar rapidamente algumas tarefas para demonstração

**Certos casos só podem ser resolvidos através de macros:**

- Definição de acções alternativas para determinados comandos de teclas: *AutoKeys*
- Criar uma rotina de arranque: *Autoexec*

O **VBA** (***Visual Basic for Applications***) é a linguagem de programação comum às aplicações do MS Office: Access, Excel e Powerpoint

O código VBA é guardado em módulos, que podem ser criados de duas formas:

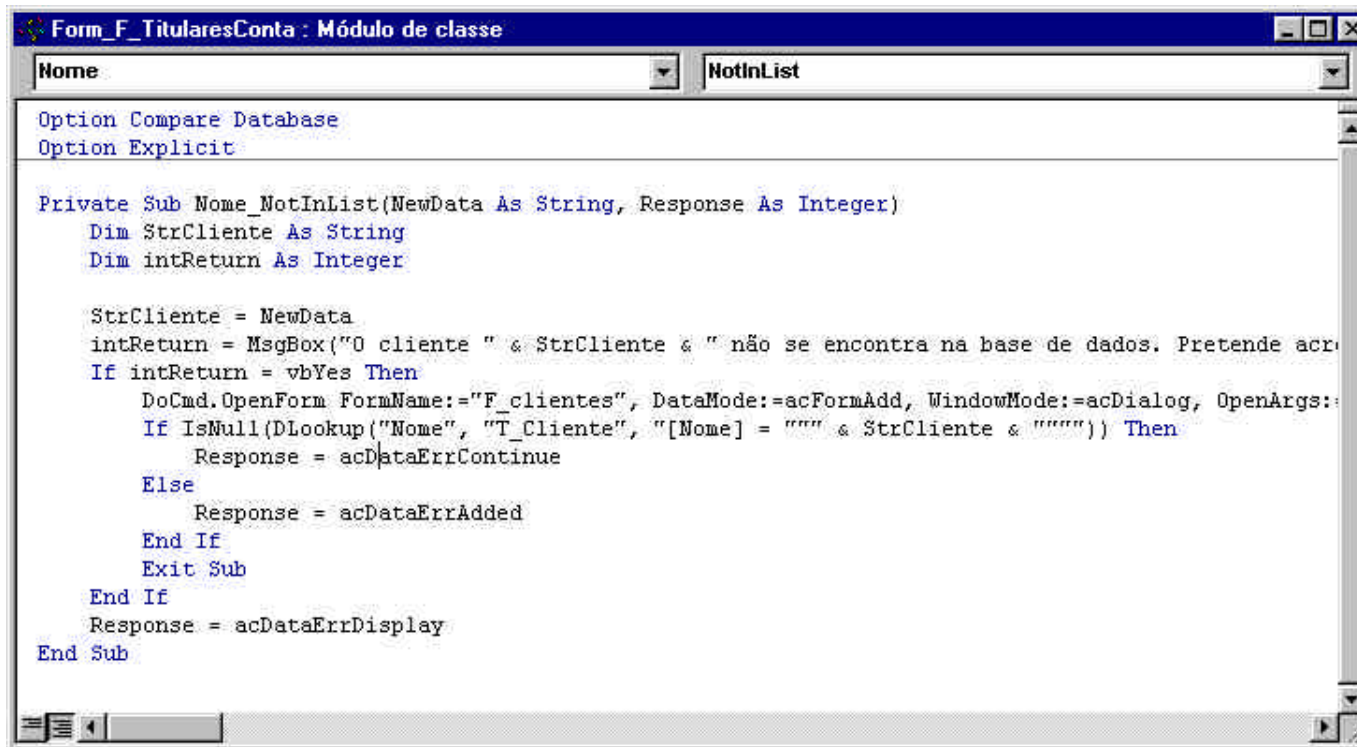
- como um objecto Módulo
- como parte de um objecto Formulário ou Relatório

### **Deve utilizar procedimentos VBA quando:**

- Necessitar de lidar com erros numa aplicação
- Pretender definir uma nova função
- Necessitar de criar novos objectos que passem parâmetros ou aceitem valores de retorno
- Necessitar de criar novos objectos (tabelas, consultas, formulários ou relatórios) a partir da aplicação
- A aplicação necessitar de interagir com outras aplicações em Windows
- ...

**Exemplo:** Um procedimento VBA para permitir acrescentar um novo cliente quando se escreve um nome que não se encontra na lista

1 - Na caixa de combinação “**Nome**” do formulário **F\_TitularesConta**, seleccionar o acontecimento “*NotInList*”



```
Form_F_TitularesConta : Módulo de classe
Nome NotInList

Option Compare Database
Option Explicit

Private Sub Nome_NotInList(NewData As String, Response As Integer)
    Dim StrCliente As String
    Dim intReturn As Integer

    StrCliente = NewData
    intReturn = MsgBox("O cliente " & StrCliente & " não se encontra na base de dados. Pretende acrescentar este cliente à base de dados?")
    If intReturn = vbYes Then
        DoCmd.OpenForm FormName:="F_clientes", DataMode:=acFormAdd, WindowMode:=acDialog, OpenArgs:=StrCliente
        If IsNull(DLookup("Nome", "T_Cliente", "[Nome] = '" & StrCliente & "'")) Then
            Response = acDataErrContinue
        Else
            Response = acDataErrAdded
        End If
        Exit Sub
    End If
    Response = acDataErrDisplay
End Sub
```

## Código do procedimento **Nome\_NotInList**

(para utilizar em conjunto com o procedimento “OnLoad” do formulário F\_Clientes)

```
Private Sub Nome_NotInList(NewData As String, Response As Integer)
    Dim StrCliente As String
    Dim intReturn As Integer

    StrCliente = NewData
    intReturn = MsgBox("O cliente " & StrCliente & " não se encontra na base de dados. Pretende acrescentá-lo?",_
        vbQuestion + vbYesNo, "Sonota")
    If intReturn = vbYes Then
        DoCmd.OpenForm FormName:="F_clientes", DataMode:=acFormAdd, WindowMode:=acDialog, OpenArgs:=StrCliente
        If IsNull(DLookup("Nome", "T_Cliente", "[Nome] = "" & StrCliente & """)) Then
            Response = acDataErrContinue
        Else
            Response = acDataErrAdded
        End If
    End Sub
End If
Response = acDataErrDisplay
End Sub
```

## Código do procedimento **Nome\_NotInList** - versão alternativa (mais simples)

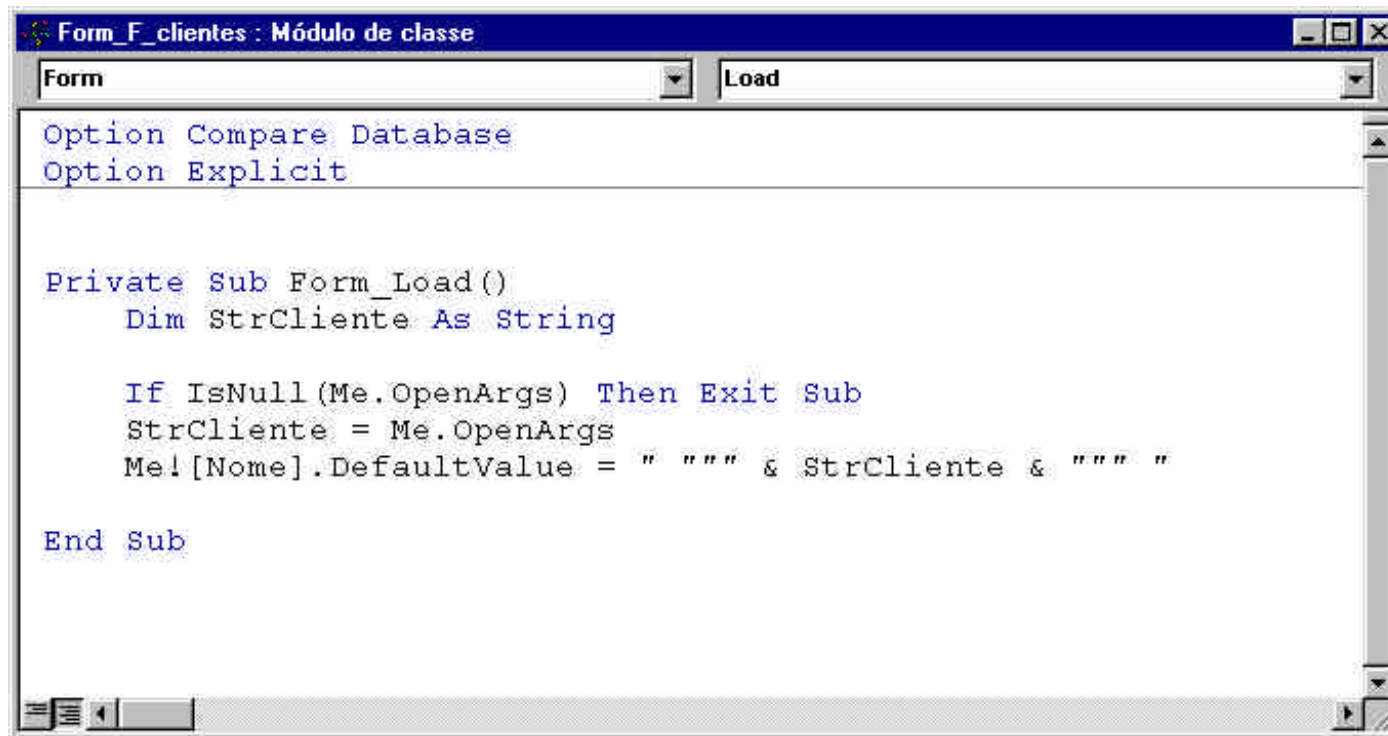
```
Private Sub Nome_NotInList(NewData As String, Response As Integer)
    Dim StrCliente As String
    Dim intReturn, int1 As Integer

    StrCliente = NewData
    intReturn = MsgBox("O cliente " & StrCliente & " não se encontra na base de dados. Pretende acrescentá-lo?",
        vbQuestion + vbYesNo, "Sonota")
    If intReturn = vbYes Then
        DoCmd.OpenForm FormName:="F_clientes", DataMode:=acFormAdd, WindowMode:=acDialog, OpenArgs:=StrCliente
        Response = acDataErrAdded
        Exit Sub
    End If
    If intReturn = vbNo Then
        intReturn = MsgBox("Operação cancelada!", vbOKOnly, "Informação")
        Response = acDataErrContinue
    End If

End Sub
```

Se o utilizador pretender acrescentar o novo cliente, o formulário F\_clientes deverá aparecer com o novo nome já preenchido.

Para isso, deverá acrescentar um novo procedimento ao acontecimento "OnLoad":



```
Form_F_clientes : Módulo de classe
Form Load
Option Compare Database
Option Explicit

Private Sub Form_Load()
    Dim StrCliente As String

    If IsNull(Me.OpenArgs) Then Exit Sub
    StrCliente = Me.OpenArgs
    Me![Nome].DefaultValue = " " & StrCliente & " "

End Sub
```

**Problemas:**

1. Criar uma macro *Autoexec* (executada automaticamente quando é aberta a aplicação e pode abrir um formulário, apresentar uma mensagem, etc.)
2. Construir uma macro com a seguinte funcionalidade: no formulário *F\_clientes*, sempre que o utilizador escrever o nome de uma cidade que já exista, preenche automaticamente o campo do Código Postal