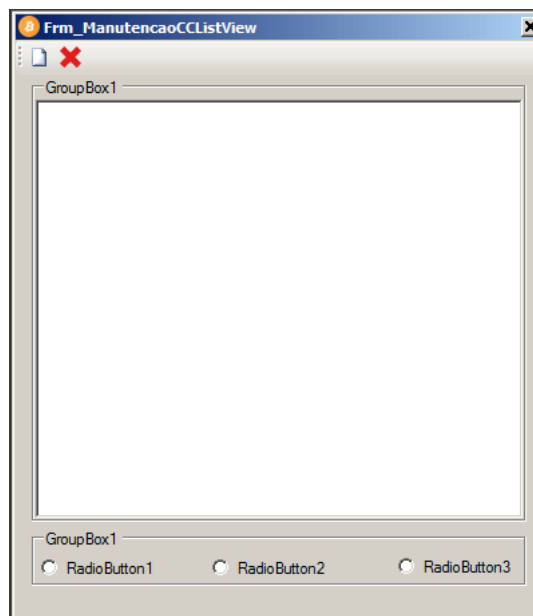


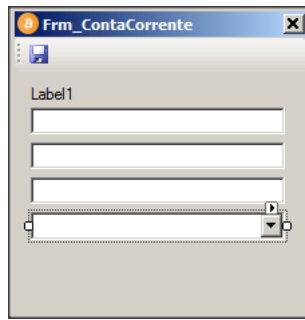
Mãos na massa

Chegou a hora de você pôr em prática o que foi visto na aula. Para isso, execute os passos listados abaixo.

- 1) Crie um formulário espelho do formulário `Frm_MantencaoCCListView` e chame-o de `Frm_MantencaoCCDatatable`. Então esse novo formulário terá um `ToolStrip`, com `Name ToolStrip1`, e dois `GroupBox`, com `Name Grp_Lista` e `Grp_CritérioOrdenacao`, respectivamente.
- 2) Dentro do `GroupBox Grp_Lista`, ele terá um `ListView`, com `Name Lvw_Contas`. Dentro do `GroupBox Grp_CritérioOrdenacao`, ele terá três `RadioButton`, com `Name Rb_Saldo`, `Rb_Nome` e `Rb_Conta`.
- 3) No componente `ToolStrip1`, selecione todos os botões padrões. Para isso clique na seta no canto superior direito da área do componente e escolha a opção **Inserir Itens Padrão**. Em seguida, apague todos os itens, exceto o item **Novo**.
- 4) Crie um novo botão na `ToolStrip1` e inclua a imagem **delete.png**, que você pode baixar [aqui \(https://caelum-online-public.s3.amazonaws.com/1016-vb-net-lambda-linq-dataset/03/delete.zip\)](https://caelum-online-public.s3.amazonaws.com/1016-vb-net-lambda-linq-dataset/03/delete.zip). O formulário ficará com o seguinte layout:



- 5) Crie outro formulário, chamado `Frm_ContaCorrenteDT`, que será parecido com o formulário `Frm_ContaCorrente`, mas com um componente novo a mais. Nesse novo formulário, adicione um `ToolStrip`, com `Name ToolStrip1`, um `Label`, com `Name Lbl_ContaCorrente`, três `TextBox`, com `Name Txt_Nome` (esse também terá a propriedade `ReadOnly` com o valor `True`), `Txt_Conta` e `Txt_Agencia`, e um `ComboBox`, com `Name Cmb_CPF`.
- 6) No componente `ToolStrip1`, selecione todos os botões padrões. Para isso clique na seta no canto superior direito da área do componente e escolha a opção **Inserir Itens Padrão**. Em seguida, apague todos os itens, exceto o item **Salvar**. O formulário ficará com o seguinte layout:



7) No código-fonte do formulário `Frm_ContaCorrenteDT`, insira as seguintes propriedades:

```
#Region "PROPRIEDADES"

Public Property Agencia As Integer
Public Property Conta As Integer
Public Property CPF As String
Public Property Retorno As Boolean

#End Region
```

8) No seu construtor, faça:

```
Public Sub New(DtClientes As DataTable)

    ' Esta chamada é requerida pelo designer.
    InitializeComponent()

    ' Adicione qualquer inicialização após a chamada InitializeComponent().

    Me.Text = "Conta Corrente"
    Retorno = False
    Lbl_ContaCorrente.Text = "Conta Corrente"
    Txt_CPF.ReadOnly = True
    Cmb_CPF.Items.Clear()
    For I As Integer = 0 To DtClientes.Rows.Count - 1

        Cmb_CPF.Items.Add(DtClientes.Rows(I)("CPF"))
    Next

End Sub
```

9) Implemente o código do evento de `Click` do item **Salvar** do menu:

```
Private Sub SalvarToolStripButton_Click(sender As Object, e As EventArgs)
    Handles SalvarToolStripButton.Click

    Retorno = True
    Agencia = Txt_Agencia.Text
    Conta = Txt_Conta.Text
    CPF = Txt_CPF.Text
    Me.Close()

End Sub
```

10) E inclua o código a ser executado quando a opção do *ComboBox* for alterada:

```
Private Sub Cmb_CPF_SelectedIndexChanged(sender As Object, e As EventArgs)
    Handles Cmb_CPF.SelectedIndexChanged
    Txt_CPF.Text = Cmb_CPF.Text
End Sub
```

11) No código-fonte do formulário *Frm_MantencaoCCDatatable*, insira a importação do *Namespace*:

```
Imports ByteBank.Bibliotecas.Classes.Clientes
```

12) Declare as variáveis globais:

```
Public DS As New DataSet
Public DtListView As New DataTable
```

13) Na sub-rotina *New()*, abaixo da *InitializeComponent()*, inicialize a propriedade *Text* dos *GroupBox*, *RadioButton* e formulário. Além disso, chame os métodos *InicializacaoContas()*, *FormatarListView()*, *CriarColunas()*, *CriarDataTableListView()*, *AtualizarListView1()* e *AtualizarListView2()*:

```
Me.Text = "Manutenção Conta Corrente"
Grp_Lista.Text = "Lista de Contas Correntes"
Grp_CriterioOrdenacao.Text = "Critério de Ordenação"
Rb_Conta.Text = "Conta"
Rb_Nome.Text = "Nome"
Rb_Saldo.Text = "Saldo"

InicializacaoContas()
FormatarListView()
CriarColunas()
CriarDataTableListView()
' AtualizarListView()
AtualizarListView1()
AtualizarListView2()
```

14) Crie e implemente os métodos *CriarDatatableListView()* e *InicializacaoContas()*:

```
Sub CriarDataTableListView()
    DtListView.Columns.Add(CriarColuna("Conta Corrente", "System.String"))
    DtListView.Columns.Add(CriarColuna("Correntista", "System.String"))
    DtListView.Columns.Add(CriarColuna("Saldo em R$", "System.Double"))
End Sub

Sub InicializacaoContas()

    ' Previamente configurar algumas contas

    ' #####
    ' Criar uma lista de contas correntes e clientes
    ' #####
```

```

Dim Cliente As New List(Of Cliente)
Cliente.Add(CriarCliente("123123123123", "João da Silva", "Engenheiro"))
Cliente.Add(CriarCliente("123123129999", "Pedro Alcantara", "Médico"))
Cliente.Add(CriarCliente("123123888888", "Luzia Medeiros", "Mecânico"))
Cliente.Add(CriarCliente("125656565656", "Márcia Andrade", "Engenheiro"))
Cliente.Add(CriarCliente("186868686868", "Joana Silveira", "Empresário"))
Cliente.Add(CriarCliente("139393939393", "Marcelo Almeida", "Engenheiro"))
Cliente.Add(CriarCliente("186868686869", "Vitor Junior", "Analista de Sistemas"))

```

```

Dim Contas As New List(Of ContaCorrente)
Contas.Add(New ContaCorrente(277, 123221))
Contas(0).titular = Cliente(0)
Contas.Add(New ContaCorrente(277, 133323))
Contas(1).titular = Cliente(0)
Contas.Add(New ContaCorrente(277, 244323))
Contas(2).titular = Cliente(2)
Contas.Add(New ContaCorrente(277, 124543))
Contas(3).titular = Cliente(1)
Contas.Add(New ContaCorrente(277, 543434))
Contas(4).titular = Cliente(1)
Contas.Add(New ContaCorrente(277, 653443))
Contas(5).titular = Cliente(3)
Contas.Add(New ContaCorrente(277, 126544))
Contas(6).titular = Cliente(4)
Contas.Add(New ContaCorrente(277, 748383))
Contas(7).titular = Cliente(5)

```

```

' #####
' criando a tabela de contas correntes
' #####

' Conta Corrente : Agencia (Int), Conta (Int), cfp (string), saldo (double)

```

```

Dim Dt As New DataTable
Dt.Columns.Add(CriarColuna("Agencia", "System.Int32"))
Dt.Columns.Add(CriarColuna("Conta", "System.Int32"))
Dt.Columns.Add(CriarColuna("CPF", "System.String"))
Dt.Columns.Add(CriarColuna("Saldo", "System.Double"))

```

```

' definir chave primária

```

```

Dim vArrayChavesContas(1) As DataColumn
Dt.Columns("Agencia").AllowDBNull = False
Dt.Columns("Conta").AllowDBNull = False
vArrayChavesContas(0) = Dt.Columns("Agencia")
vArrayChavesContas(1) = Dt.Columns("Conta")
Dt.PrimaryKey = vArrayChavesContas

```

```

' Nome da tabela

```

```

Dt.TableName = "Contas Correntes"

```

```

' #####
' Criar a tabela de Clientes
' #####

```

```

Dim Dt2 As New DataTable

```

```

Dim Dt2 As New DataColumn
Dt2.Columns.Add(CriarColuna("CPF", "System.String"))
Dt2.Columns.Add(CriarColuna("Nome", "System.String"))
Dt2.Columns.Add(CriarColuna("Profissao", "System.String"))

' Criando chave primária para cliente

Dim vArrayChavesCliente(0) As DataColumn
Dt2.Columns("CPF").AllowDBNull = False
vArrayChavesCliente(0) = Dt2.Columns("CPF")
Dt2.PrimaryKey = vArrayChavesCliente

' Nome da tabela

Dt2.TableName = "Clientes"

' #####
' Criar o DataSET
' #####

DS.Tables.Add(Dt)
DS.Tables.Add(Dt2)

' #####
' Criar a chave estrangeira
'
' Clientes -----> Contas Correntes (FK)
' #####

Dim FKClientesCC As New ForeignKeyConstraint("FKClientesCC",
    DS.Tables("Clientes").Columns("CPF"),
    DS.Tables("Contas Correntes").Columns("CPF"))

FKClientesCC.DeleteRule = Rule.None

DS.Tables("Contas Correntes").Constraints.Add(FKClientesCC)

' #####
' Incluir dados da tabela de clientes do DataSET
' #####

For I As Integer = 0 To Cliente.Count - 1
    Dim ClienteAtual As Cliente = Cliente(I)
    Dim Dr As DataRow = DS.Tables("Clientes").NewRow

    Dr(0) = ClienteAtual.cpf
    Dr(1) = ClienteAtual.nome
    Dr(2) = ClienteAtual.profissao

    DS.Tables("Clientes").Rows.Add(Dr)
Next

' #####
' Incluir dados da tabela de contas corrente do DataSET
' #####

For I As Integer = 0 To Contas.Count - 1
    Dim ContaAtual As ContaCorrente = Contas(I)
    Dim Dr As DataRow = DS.Tables("Contas Correntes").NewRow

```

```

Dim Dr As DataRow = DS.Tables("Contas Correntes").NewRow

Dr(0) = ContaAtual.agencia
Dr(1) = ContaAtual.numero
Dr(2) = ContaAtual.titular.cpf
Dr(3) = ContaAtual.saldo

DS.Tables("Contas Correntes").Rows.Add(Dr)
Next

End Sub

```

15) Crie as funções `CriarCliente` e `CriarColuna` :

```

Function CriarCliente(cpf As String, nome As String, profissao As String) As Cliente

    Dim C As New Cliente
    C.cpf = cpf
    C.nome = nome
    C.profissao = profissao

    Return C

End Function

Function CriarColuna(NomeColuna As String, TipColuna As String) As DataColumn

    Dim Dc As New DataColumn
    Dc.ColumnName = NomeColuna
    Dc.DataType = System.Type.GetType(TipColuna)

    Return Dc

End Function

```

16) Crie e implemente os métodos `FormatarListView()`, `CriarColunas()` e `AtualizarListView()` :

```

Sub FormatarListView()
    ' Formatar o ListView

    Lvw_Contas.View = View.Details
    Lvw_Contas.LabelEdit = False
    Lvw_Contas.AllowColumnReorder = False
    Lvw_Contas.FullRowSelect = True
    Lvw_Contas.GridLines = True

End Sub

Sub CriarColunas()
    ' Configurar colunas

    Lvw_Contas.Columns.Add("Conta Corrente", 100, HorizontalAlignment.Center)
    Lvw_Contas.Columns.Add("Correntista", 150, HorizontalAlignment.Center)
    Lvw_Contas.Columns.Add("Saldo em R$", 100, HorizontalAlignment.Center)

End Sub

```

```
Sub AtualizarListView()  
  
    ' Inclusao dos elementos  
  
    Lvw_Contas.Items.Clear()  
  
    For I As Integer = 0 To DS.Tables("Contas Correntes").Rows.Count - 1  
        Dim vContaNumero As String = DS.Tables("Contas Correntes").Rows(I)("Agencia").ToString +  
            "-" + DS.Tables("Contas Correntes").Rows(I)("Conta").ToString  
        Dim Linha As New ListViewItem  
        Linha.Name = vContaNumero  
        Linha.Text = vContaNumero  
        Linha.SubItems.Add(BuscaCliente(DS.Tables("Contas Correntes").Rows(I)("CPF").ToString))  
        Linha.SubItems.Add(DS.Tables("Contas Correntes").Rows(I)("saldo").ToString)  
        Lvw_Contas.Items.Add(Linha)  
    Next  
  
End Sub
```

17) Crie as rotinas AtualizarListView1() e AtualizarListView2() :

```
Sub AtualizarListView1()  
  
    ' Inclusao dos elementos  
  
    DtListView.Rows.Clear()  
  
    For I As Integer = 0 To DS.Tables("Contas Correntes").Rows.Count - 1  
        Dim vContaNumero As String = DS.Tables("Contas Correntes").Rows(I)("Agencia").ToString +  
            "-" + DS.Tables("Contas Correntes").Rows(I)("Conta").ToString  
        Dim Linha As DataRow = DtListView.NewRow  
        Linha(0) = vContaNumero  
        Linha(1) = BuscaCliente(DS.Tables("Contas Correntes").Rows(I)("CPF").ToString)  
        Linha(2) = DS.Tables("Contas Correntes").Rows(I)("saldo")  
        DtListView.Rows.Add(Linha)  
    Next  
  
End Sub
```

```
Sub AtualizarListView2()  
  
    ' Inclusao dos elementos  
  
    Lvw_Contas.Items.Clear()  
  
    For I As Integer = 0 To DtListView.Rows.Count - 1  
        Dim Linha As New ListViewItem  
        Linha.Name = DtListView.Rows(I)(0).ToString  
        Linha.Text = DtListView.Rows(I)(0).ToString  
        Linha.SubItems.Add(DtListView.Rows(I)(1).ToString)  
        Linha.SubItems.Add(DtListView.Rows(I)(2))  
        Lvw_Contas.Items.Add(Linha)  
    Next  
  
End Sub
```

18) Crie a função `BuscaCliente`, com o seguinte código:

```
Function BuscaCliente(CPF) As String

    Dim vCritério As String = "CPF = '" + CPF + "'"
    Dim DtFiltro As DataTable = DS.Tables("Clientes").Clone
    Dim DrLinhas As DataRow() = DS.Tables("Clientes").Select(vCritério)
    For Each row As DataRow In DrLinhas
        DtFiltro.ImportRow(row)
    Next
    If DtFiltro.Rows.Count > 0 Then
        Return DtFiltro.Rows(0)("nome")
    End If
    Return ""

End Function
```

19) No código do evento de `Click` do botão **Novo** do menu, insira o seguinte código:

```
Private Sub NovoToolStripButton_Click(sender As Object, e As EventArgs)
    Handles NovoToolStripButton.Click

    Dim F As New Frm_ContaCorrenteDT(DS.Tables("Clientes"))
    F.ShowDialog()
    If F.Retorno Then
        Dim Dr As DataRow = DS.Tables("Contas Correntes").NewRow
        Dr(0) = F.Agencia
        Dr(1) = F.Conta
        Dr(2) = F.CPF
        Dr(3) = 100
        DS.Tables("Contas Correntes").Rows.Add(Dr)
        ' AtualizarListView()
        AtualizarListView1()
        AtualizarListView2()
    End If

End Sub
```

20) No código do evento de `Click` do botão **Apagar** do menu, insira o seguinte código:

```
Private Sub ToolStripButton1_Click(sender As Object, e As EventArgs)
    Handles ToolStripButton1.Click

    For I As Integer = 0 To Lvw_Contas.Items.Count - 1
        If Lvw_Contas.Items(I).Selected Then
            Dim vAgenciaConta As String = Lvw_Contas.Items(I).SubItems(0).Text
            Dim vPos As Integer = InStr(vAgenciaConta, "-")
            Dim vAgencia As Integer = Val(Mid(vAgenciaConta, 1, vPos))
            Dim VConta As Integer = Val(Mid(vAgenciaConta, vPos + 1))
```

```

        Dim vCritério2 As String = "AGENCIA = " + vAgencia.ToString + " AND CONTA = " + VConta.1
        Dim DrLinhas2 As DataRow() = DS.Tables("Contas Correntes").Select(vCritério2)
        For Each row As DataRow In DrLinhas2
            DS.Tables("Contas Correntes").Rows.Remove(row)
        Next
    End If
Next
' AtualizarListView()
AtualizarListView1()
AtualizarListView2()

End Sub

```

21) Implemente o código da opção de ordenação por conta:

```

Private Sub Rb_Conta_CheckedChanged(sender As Object, e As EventArgs)
    Handles Rb_Conta.CheckedChanged

    AtualizarListView1()

    Dim DtView As DataView
    DtView = DtListView.DefaultView
    Dim vCritérioOrdenacao As String = "Conta Corrente"
    DtView.Sort = vCritérioOrdenacao
    DtListView = DtView.ToTable

    AtualizarListView2()

End Sub

```

22) Implemente o código na opção de ordenação por nome:

```

Private Sub Rb_Nome_CheckedChanged(sender As Object, e As EventArgs)
    Handles Rb_Nome.CheckedChanged

    AtualizarListView1()

    Dim DtView As DataView
    DtView = DtListView.DefaultView
    Dim vCritérioOrdenacao As String = "Correntista"
    DtView.Sort = vCritérioOrdenacao
    DtListView = DtView.ToTable

    AtualizarListView2()

End Sub

```

23) Implemente o código na opção de ordenação por saldo:

```

Private Sub Rb_Saldo_CheckedChanged(sender As Object, e As EventArgs)
    Handles Rb_Saldo.CheckedChanged

    AtualizarListView1()

```

```
Dim DtView As DataView
DtView = DtListView.DefaultView
Dim vCritérioOrdenacao As String = "Saldo em R$"
DtView.Sort = vCritérioOrdenacao
DtListView = DtView.ToTable

AtualizarListView2()
```

End Sub

24) Por fim, no formulário `Frm_Principal01`, adicione mais uma opção no menu, chamada **Manutenção ListView DT**. Em seguida, implemente o código do evento de `Click` deste novo item do menu:

```
Private Sub ManutençãoListViewDTToolStripMenuItem_Click(sender As Object, e As EventArgs)
    Handles ManutençãoListViewDTToolStripMenuItem.Click
    Dim F As New Frm_ManutencaoCCDataTable
    F.MdiParent = Me
    F.Show()
End Sub
```