

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) No projeto **ByteBank.Bibliotecas**, edite a classe **ManutencaoClienteBD**, mudando a importação do Namespace para:

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

2) Implemente também os métodos `RetornaContas()`, `RetornaListaClientes()`, `RetornaListaContasCorrente()` e `DescarregaContasCorrentes()`:

```
Shared Function RetornaContas() As DataTable
```

```
    Dim Dt As New DataTable
    Try
        Dim DB As New SqlConnection("DESKTOP-JFHLUPE", "BYTEBANK")
        Dim Retorno As String =
            DB.SqlQuery("SELECT [Agencia],[Conta],[CPF],[Saldo] FROM [Conta]", Dt)
        DB.Close()
    Catch ex As Exception
        Throw New Exception(ex.Message)
    End Try
    Return Dt
```

```
End Function
```

```
Shared Function RetornaListaClientes() As List(Of Cliente)
```

```
    Dim Cli As New List(Of Cliente)
    Try
        Dim dt As New DataTable
        dt = RetornaClientes()
        For i As Integer = 0 To dt.Rows.Count - 1
            Dim cliAux As New Cliente
            cliAux.cpf = dt.Rows(i)("cpf").ToString
            cliAux.nome = dt.Rows(i)("nome").ToString
            cliAux.profissao = dt.Rows(i)("profissao").ToString
            Cli.Add(cliAux)
        Next
        Return Cli
    Catch ex As Exception
        Throw New Exception(ex.Message)
    End Try
```

```
End Function
```

```
Shared Function RetornaListaContasCorrente(Cliente As List(Of Cliente)) As List(Of ContaCorrente)
```

```
    Dim CC As New List(Of ContaCorrente)
    Try
        Dim dt As New DataTable
```

```

dt = RetornaContas()
For i As Integer = 0 To dt.Rows.Count - 1
    Dim contaAux As New ContaCorrente(dt.Rows(i)("agencia"), dt.Rows(i)("conta"))
    contaAux.Depositar(dt.Rows(i)("saldo") - 100)
    Dim vCpf As String = dt.Rows(i)("cpf")
    Dim ClienteFiltro As List(Of Cliente) =
        Cliente.FindAll(Function(c As Cliente) c.cpf = vCpf)
    If Not (ClienteFiltro Is Nothing) AndAlso ClienteFiltro.Count > 0 Then
        contaAux.titular = ClienteFiltro(0)
    End If
    CC.Add(contaAux)
Next
Return CC
Catch ex As Exception
    Throw New Exception(ex.Message)
End Try

```

End Function

Shared Function DescarregaContasCorrentes(Dt As DataTable) As String

```

Try
    Dim DB As New SqlConnection("DESKTOP-JFHLUPE", "BYTEBANK")
    Dim Comando As String = "DELETE FROM CONTA"
    Dim Retorno2 As String = DB.SqlCommand(Comando)
    DB.Close()

    For I As Integer = 0 To Dt.Rows.Count - 1
        DB = New SqlConnection("DESKTOP-JFHLUPE", "BYTEBANK")
        Dim vAgencia As Integer = Dt.Rows(I)("Agencia")
        Dim vConta As Integer = Dt.Rows(I)("Conta")
        Dim vCPF As String = Dt.Rows(I)("CPF")
        Dim vSaldo As Double = Dt.Rows(I)("Saldo")
        Comando = "INSERT INTO CONTA (Agencia, Conta, CPF, Saldo) "
        Comando += "VALUES (" + vAgencia.ToString + "," + vConta.ToString +
            "," + vCPF + "," + vSaldo.ToString + ")"
        Retorno2 = DB.SqlCommand(Comando)
        DB.Close()
    Next
    Return "Contas atualizadas com sucesso"
Catch ex As Exception
    Throw New Exception(ex.Message)
End Try
Return ""

```

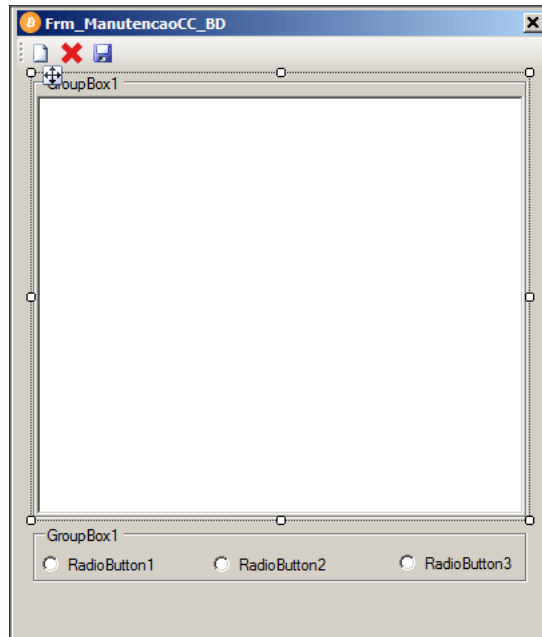
End Function

3) No projeto **ByteBank.SistemaAgencia**, crie um novo formulário, chamado **Frm_MantencaoCC_BD**. Nesse novo formulário, inclua o ícone do ByteBank. Depois, adicione um *ToolStrip*, com Name *ToolStrip1*, mantendo apenas o ícone **Novo**, do padrão do objeto. Em seguida inclua um ícone que representa a exclusão.

4) Ainda no formulário, adicione dois *GroupBox*, com Name *Grp_Lista* e *Grp_CritérioOrdenacao*, respectivamente.

5) Dentro do *GroupBox Grp_Lista*, adicione um *ListView*, com Name *Lvw_Contas*.

6) Dentro do *GroupBox* Grp_CritérioOrdenacao , adicione três *RadioButton*, com Name Rb_Saldo , Rb_Nome e Rb_Conta , respectivamente. O formulário ficará com o seguinte layout:



7) No código-fonte do formulário, inclua a importação dos Namespace :

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

8) Instancie globalmente um *DataSet* e um *DataTable* :

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

9) Adicione o método *New()* e inclua a inicialização da propriedade *Text* do formulário, dos *GroupBox* e dos *RadioButton*. Além disso, chame os métodos *InicializacaoContas()*, *FormatarListView()*, *CriarColunas()*, *CriarDataTableListView()*, *AtualizarListView1()* e *AtualizarListView2()* :

```
Public Sub New()

    ' Esta chamada é requerida pelo designer.
    InitializeComponent()

    ' Adicione qualquer inicialização após a chamada InitializeComponent().

    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()

```

```
End Sub
```

10) 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)
    Dim Contas As New List(Of ContaCorrente)

    Try
        Cliente = ManutencaoClienteBD.RetornaListaClientes()
        Contas = ManutencaoClienteBD.RetornaListaContasCorrente(Cliente)
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try

    ' #####
    ' 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
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

    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

```

11) Agora, implemente os métodos `CriarCliente`, `CriarColuna` e `CriarColunas`:

```

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

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

```

12) Por fim, implemente também os métodos `FormatarListView()`, `AtualizarListView()`, `AtualizarListView1()` e `AtualizarListView2()`:

```

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 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

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

```

13) E implemente o método `BuscarCPF` :

```
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

```

14) No clique do botão **Novo** da barra de ferramentas, inclua:

```

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

```

15) No clique de exclusão, na barra de ferramentas, inclua:

```

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.ToString
            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
End Sub

```

```
' AtualizarListView()  
AtualizarListView1()  
AtualizarListView2()
```

```
End Sub
```

16) No evento de `Click` do botão **Salvar**, na barra de ferramentas, inclua:

```
Private Sub SalvarToolStripButton_Click(sender As Object, e As EventArgs)  
    Handles SalvarToolStripButton.Click  
  
    Dim Dt As New DataTable  
    Dt = DS.Tables("Contas Correntes")  
    Try  
        Dim vRetorno As String = ManutencaoClienteBD.DescarregaContasCorrentes(Dt)  
        MsgBox(vRetorno)  
    Catch ex As Exception  
        MsgBox(ex.Message)  
    End Try  
    Me.Close()  
  
End Sub
```

17) Faltava agora implementar a ordenação por nome, saldo e conta, que será realizada ao clicar nos seus respectivos *RadioButton*:

```
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  
  
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

```
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
```

18) Ainda no projeto **ByteBank.SistemaAgencia**, vá no formulário **Frm_Principal01** e inclua um novo item de menu, dentro de **Cadastro BD**, chamado **Manutenção de Contas Correntes BD**.

19) No clique deste novo item do menu, inclua a chamada ao novo formulário:

```
Private Sub ManutençãoDeContasCorrentesBDToolStripMenuItem_Click(sender As Object, e As EventArgs)
    Handles ManutençãoDeContasCorrentesBDToolStripMenuItem.Click
    Dim F As New Frm_ManutencaoCC_BD
    F.MdiParent = Me
    F.Show()
End Sub
```

