

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) Faça o download do **MySQL Installer** [aqui \(https://dev.mysql.com/downloads/installer/\)](https://dev.mysql.com/downloads/installer/) e instale-o em sua máquina, conforme mostrado nos vídeos.

2) Entre no **MySQL Workbench**. Nele, crie um *schema* chamado **bytebank**.

3) Execute os comandos abaixo para criação das tabelas e inclusão de alguns registros:

```
USE bytebank;
```

```
CREATE TABLE Cliente (  
    CPF nvarchar(20) NOT NULL,  
    nome nvarchar(50) NULL,  
    profissao nvarchar(50) NULL,  
    CONSTRAINT PK_Cliente PRIMARY KEY (  
        CPF  
    )  
);
```

```
CREATE TABLE Conta (  
    Agencia int NOT NULL,  
    Conta int NOT NULL,  
    CPF nvarchar(20) NULL,  
    Saldo float NULL,  
    CONSTRAINT PK_Conta PRIMARY KEY (  
        Agencia ASC,  
        Conta ASC  
    )  
);
```

```
INSERT Cliente (CPF, nome, profissao) VALUES (N'00354233233', N'Marcia Andrade', N'Motorista');  
INSERT Cliente (CPF, nome, profissao) VALUES (N'00388894564', N'Victor Martins', N'Engenheiro');  
INSERT Cliente (CPF, nome, profissao) VALUES (N'00400043434', N'Juliana Almeida', N'Médica');  
INSERT Cliente (CPF, nome, profissao) VALUES (N'03485964356', N'Cláudia Alcantara', N'Dona de Casa');  
INSERT Cliente (CPF, nome, profissao) VALUES (N'05423855960', N'Camila Pereira', N'Empresário');  
INSERT Cliente (CPF, nome, profissao) VALUES (N'11111111111', N'Teste Cliente CPF 1111', N'Engenheiro');  
INSERT Cliente (CPF, nome, profissao) VALUES (N'30654981205', N'Flávia Alessandra', N'Empresário');  
INSERT Cliente (CPF, nome, profissao) VALUES (N'33184639950', N'Glória Andrade', N'Dentista');  
INSERT Cliente (CPF, nome, profissao) VALUES (N'33954677183', N'Menendez Grande', N'Médico');  
INSERT Cliente (CPF, nome, profissao) VALUES (N'34950468204', N'Paulo Gomode', N'Padeiro');  
INSERT Cliente (CPF, nome, profissao) VALUES (N'38923401284', N'Rúfio Moraes', N'Engenheiro');  
INSERT Cliente (CPF, nome, profissao) VALUES (N'44054890137', N'Cláudia Silveira', N'Engenheiro');  
INSERT Cliente (CPF, nome, profissao) VALUES (N'44638954305', N'Gomes Carneiro', N'Jornalista');  
INSERT Cliente (CPF, nome, profissao) VALUES (N'55467394659', N'Tina Beatriz', N'Jornalista');  
INSERT Cliente (CPF, nome, profissao) VALUES (N'55643344290', N'Pedro Andrade', N'Músico');  
INSERT Cliente (CPF, nome, profissao) VALUES (N'55843923048', N'Jorge Silveira', N'Músico');  
INSERT Cliente (CPF, nome, profissao) VALUES (N'55890345601', N'Antônio Carlos', N'Médico');  
INSERT Cliente (CPF, nome, profissao) VALUES (N'55943400454', N'Taulo Petra', N'Médico');  
INSERT Cliente (CPF, nome, profissao) VALUES (N'66104395288', N'Lúcio Mauro', N'Padeiro');  
INSERT Cliente (CPF, nome, profissao) VALUES (N'66478590223', N'Daniela Oliveira', N'Dentista');
```

```
INSERT Cliente (CPF, nome, profissao) VALUES (N'66489503679', N'Jussara Floriano', N'Jornalista');
INSERT Cliente (CPF, nome, profissao) VALUES (N'73405463789', N'André Luiz Almeida', N'Empresário');
INSERT Cliente (CPF, nome, profissao) VALUES (N'77389047362', N'Alexandra Teves', N'Engenheiro');
INSERT Cliente (CPF, nome, profissao) VALUES (N'77777777778', N'Teste CPF 99', N'Médico');
INSERT Cliente (CPF, nome, profissao) VALUES (N'77845002843', N'Cris Andrade', N'Médico');
INSERT Cliente (CPF, nome, profissao) VALUES (N'88490493044', N'Márcia Pertiz', N'Engenheiro');
INSERT Cliente (CPF, nome, profissao) VALUES (N'88493045439', N'Mônica Yoná', N'Padeiro');
INSERT Cliente (CPF, nome, profissao) VALUES (N'88493056849', N'Márcia Maria', N'Jornalista');
INSERT Cliente (CPF, nome, profissao) VALUES (N'93043286594', N'Tércio Parreira', N'Jornalista');
INSERT Cliente (CPF, nome, profissao) VALUES (N'99034823946', N'Cláudio Bomtempo', N'Engenheiro');
INSERT Cliente (CPF, nome, profissao) VALUES (N'99304543885', N'Flávio Pedro', N'Dentista');
INSERT Cliente (CPF, nome, profissao) VALUES (N'99367810462', N'Maria de Pedrosa', N'Médico');
INSERT Cliente (CPF, nome, profissao) VALUES (N'99406348677', N'Denise Froz', N'Dentista');
INSERT Cliente (CPF, nome, profissao) VALUES (N'99433454344', N'Zanir Moreira', N'Músico');
```

```
INSERT Conta (Agencia, Conta, CPF, Saldo) VALUES (288, 3352632, N'55467394659', 100);
```

```
ALTER TABLE Conta ADD CONSTRAINT FK_Conta_Cliente FOREIGN KEY(CPF)
REFERENCES Cliente (CPF);
```

4) Modifique a instalação do seu MySQL para que possa admitir que os usuários do banco não utilize senhas criptografadas. Isso irá permitir que acessem os dados através do Visual Basic .NET. Então, reconfigure a instalação do **MySQL Server**, modificando o método de autenticação.

5) Através do **Prompt de Comando**, crie um usuário, digitando os comandos abaixo:

```
CD \
CD "Program Files"
CD MySQL
CD "MySQL Server 8.0"
CD bin
mysql -u root -p -h localhost
```

Após isso, digite a senha do usuário **root** que foi determinada na instalação.

Continuando:

```
CREATE USER 'bytebank'@'localhost' IDENTIFIED BY '12345';
GRANT ALL PRIVILEGES ON bytebank.* TO 'bytebank'@'localhost';
quit;
mysql -u bytebank -p -h localhost
use bytebank
select * from cliente;
quit;
```

Assim, será criado um usuário chamado **bytebank**, com a senha **12345**, no MySQL.

6) No projeto **ByteBank.Bibliotecas** adicione uma nova referência para a DLL **MySQL.Data.dll**, que está localizada em **C:\Arquivos de Programas (x86)\MySQL\Connector NET 8.0\Assemblies\v4.5.2**.

7) Ainda no projeto **Bytebank.Bibliotecas**, crie uma classe dentro da pasta **Classes\Banco de Dados**, chamada **MySQLConn**, com o seguinte conteúdo:

```
Imports MySql.Data.MySqlClient

Namespace Classes.BancoDeDados

    Public Class MySQLConn

        #Region "PROPRIEDADES"

            Public Property StringConn As String
            Public Property connDB As MySqlConnection

        #End Region

        #Region "CONSTRUTOR"

            Public Sub New(Servidor As String, vUser As String, vPassword As String, vDataBase As String)

                'server=<SERVIDOR>;database=<SCHEMA>;User Id=<USUARIO>;Password=SENHA"
                StringConn = "server=" + Servidor + ";database=" + vDataBase +
                    ";User Id=" + vUser + ";Password=" + vPassword + ";"
                connDB = New MySqlConnection(StringConn)
                connDB.Open()

            End Sub

        #End Region

        #Region "MÉTODOS"

            Public Function SQLQuery(SQL As String, ByRef DT As DataTable) As String

                Try
                    Dim myCommand As IDbCommand = New MySqlCommand(SQL, connDB)
                    myCommand.CommandTimeout = 0
                    Dim myReader As IDataReader = myCommand.ExecuteReader
                    DT.Load(myReader)
                    Return ""
                Catch ex As Exception
                    Throw New Exception(ex.Message)
                End Try

            End Function

            Public Function SQLCommand(SQL As String) As String

                Try
                    Dim myCommand As IDbCommand = New MySqlCommand(SQL, connDB)
                    myCommand.CommandTimeout = 0
                    myCommand.ExecuteReader()
                    Return ""
                Catch ex As Exception
                    Throw New Exception(ex.Message)
                End Try

            End Function

            Public Sub Close()
```

```

        connDB.Close()
    End Sub

```

```

#End Region

```

```

End Class

```

```

End Namespace

```

8) No mesmo diretório, crie e implemente a classe `ManutencaoClienteBDMYSQL` :

```

Imports ByteBank.Bibliotecas.Classes.Clientes

```

```

Namespace Classes.BancoDeDados

```

```

    Public Class ManutencaoClienteBDMYSQL

```

```

        Shared Function RetornaClientes() As DataTable

```

```

            Dim Dt As New DataTable
            Try
                Dim DB As New MySQLConn("localhost", "bytebank", "12345", "bytebank")
                Dim Retorno As String =
                    DB.SQLQuery("SELECT CPF,nome,profissao FROM Cliente", Dt)
                DB.Close()
            Catch ex As Exception
                Throw New Exception(ex.Message)
            End Try
            Return Dt

```

```

        End Function

```

```

        Shared Function RetornaContas() As DataTable

```

```

            Dim Dt As New DataTable
            Try
                Dim DB As New MySQLConn("localhost", "bytebank", "12345", "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 EditarCliente(vCPF As String, vNome As String, vProfissao As String) As Strin

```

```

            Try
                Dim DB As New MySQLConn("localhost", "bytebank", "12345", "bytebank")
                Dim Dt As New DataTable
                Dim Retorno As String =
                    DB.SQLQuery("SELECT CPF FROM Cliente WHERE CPF = '" + vCPF + "'", Dt)
                Dim vMensagem As String = ""

```

```

If Dt.Rows.Count = 0 Then
    Dim Comando As String =
        "INSERT INTO CLIENTE (CPF, nome, profissao) VALUES ('" +
            vCPF + "',''" + vNome + "',''" + vProfissao + "')"
    Dim Retorno2 As String = DB.SQLCommand(Comando)
    vMensagem = "Cliente " + vCPF + " incluído com sucesso."
Else
    Dim Comando As String =
        "UPDATE CLIENTE SET nome = '" + vNome + "', profissao = '" +
            vProfissao + "' WHERE CPF = '" + vCPF + "'"
    Dim Retorno2 As String = DB.SQLCommand(Comando)
    DB.Close()
    vMensagem = "Cliente " + vCPF + " alterado com sucesso."
End If
DB.Close()
Return vMensagem
Catch ex As Exception
    Throw New Exception(ex.Message)
End Try
Return ""

```

End Function

Shared Function ExcluiCliente(vCPF As String) As String

```

Try
    Dim DB As New MySQLConn("localhost", "bytebank", "12345", "bytebank")
    Dim Dt As New DataTable
    Dim Retorno As String = DB.SQLQuery("SELECT CPF FROM Cliente WHERE CPF = '" + vCPF + "'")
    Dim vMensagem As String = ""
    If Dt.Rows.Count = 0 Then
        vMensagem = "Cliente não pode ser excluído porque não existe: " + vCPF
    Else
        Dim Comando As String = "DELETE FROM Cliente WHERE CPF = '" + vCPF + "'"
        Dim Retorno2 As String = DB.SQLCommand(Comando)
        DB.Close()
        vMensagem = "Cliente " + vCPF + " excluído com sucesso."
    End If
    DB.Close()
    Return vMensagem
Catch ex As Exception
    Throw New Exception(ex.Message)
End Try
Return ""

```

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 ContaCorr

    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 MySQLConn("localhost", "bytebank", "12345", "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 MySQLConn("localhost", "bytebank", "12345", "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 ""

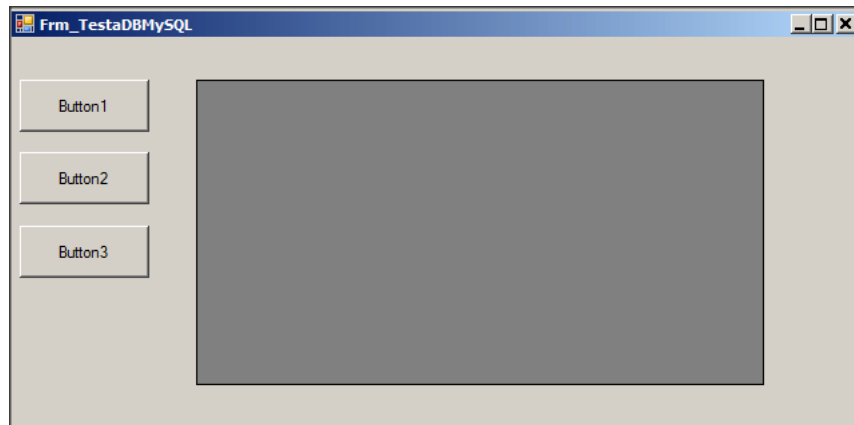
End Function

```

```
End Class
```

```
End Namespace
```

9) No projeto **ByteBank.SistemaAgencia**, crie um formulário chamado **Frm_TestaDBMySQL**. Nele, adicione três *Button*, com *Name* **Button1**, **Button2** e **Button3**, respectivamente, e um *Datagrid*, com *Name* **DTGrid_Resultado**. O layout do formulário ficará assim:



10) Vá no código-fonte deste novo formulário e importe o *Namespace* de acesso ao banco de dados:

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

11) Implemente o código do evento de *Click* dos botões **Button1**, **Button2** e **Button3**:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
```

```
    DtGrid_Resultado.Columns.Clear()
    DtGrid_Resultado.DataSource = Nothing
```

```
    Try
```

```
        Dim Dt As New DataTable
        Dim DB As New MySQLConn("localhost", "bytebank", "12345", "bytebank")
        Dim Retorno As String = DB.SQLQuery("SELECT CPF, NOME, PROFISSAO FROM CLIENTE", Dt)
        DtGrid_Resultado.DataSource = Dt
        DB.Close()
```

```
    Catch ex As Exception
        MsgBox(ex.Message)
```

```
    End Try
```

```
End Sub
```

```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
```

```
    Try
```

```
        Dim Comando As String =
            "INSERT INTO CLIENTE (CPF, nome, profissao) VALUES ('99433454344','Zanir Moreira','Músic"
        Dim DB As New MySQLConn("localhost", "bytebank", "12345", "bytebank")
        Dim Dt As New DataTable
        Dim Retorno As String =
            DB.SQLQuery("SELECT CPF,nome,profissao FROM Cliente WHERE CPF = '99433454344'", Dt)
        If Dt.Rows.Count > 0 Then
```

```

        MsgBox("Cliente não pode ser incluído porque já existe na base")
    Else
        Dim Retorno2 As String = DB.SQLCommand(Comando)
        MsgBox("Cliente incluído com sucesso")
    End If
    DB.Close()
Catch ex As Exception
    MsgBox(ex.Message)
End Try

End Sub

Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click

    Try
        Dim Comando As String = "DELETE FROM CLIENTE WHERE CPF = '99433454344'"
        Dim DB As New MySQLConn("localhost", "bytebank", "12345", "bytebank")
        Dim Retorno As String = DB.SQLCommand(Comando)
        DB.Close()
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try

End Sub

```

12) Altere o formulário `Frm_ManutençãoCCBD`, adicionando um novo *Label*, com `Name` `Lbl_Database`.

13) No código-fonte deste formulário, inicialize a propriedade `Text` do novo *Label*, dentro do método `New`:

```
Lbl_Database.Text = "Banco usado: " + TipoBanco
```

14) No método `InicializacaoContas`, modifique de:

```

Try
    Select Case TipoBanco
        Case "SQLSERVER"
            Cliente =
                ManutencaoClienteBD.RetornaListaClientes()
            Contas =
                ManutencaoClienteBD.RetornaListaContasCorrente(Cliente)
        Case "ORACLE"
            Cliente =
                ManutencaoClienteBDOracle.RetornaListaClientes()
            Contas =
                ManutencaoClienteBDOracle.RetornaListaContasCorrente(Cliente)
    End Select
Catch ex As Exception
    MsgBox(ex.Message)
End Try

```

Para:

```
Try
```

```

Select Case TipoBanco
    Case "SQLSERVER"
        Cliente =
            ManutencaoClienteBD.RetornaListaClientes()
        Contas =
            ManutencaoClienteBD.RetornaListaContasCorrente(Cliente)
    Case "ORACLE"
        Cliente =
            ManutencaoClienteBDOracle.RetornaListaClientes()
        Contas =
            ManutencaoClienteBDOracle.RetornaListaContasCorrente(Cliente)
    Case "MYSQL"
        Cliente =
            ManutencaoClienteBDMySQL.RetornaListaClientes()
        Contas =
            ManutencaoClienteBDMySQL.RetornaListaContasCorrente(Cliente)
End Select
Catch ex As Exception
    MsgBox(ex.Message)
End Try

```

15) No código do evento Click do botão **Salvar** do menu, modifique de:

```

Try
    Dim vRetorno As String = ""
    Select Case TipoBanco
        Case "SQLSERVER"
            vRetorno =
                ManutencaoClienteBD.DescarregaContasCorrentes(Dt)
        Case "ORACLE"
            vRetorno =
                ManutencaoClienteBDOracle.DescarregaContasCorrentes(Dt)
    End Select
    MsgBox(vRetorno)
Catch ex As Exception

    MsgBox(ex.Message)
End Try

```

Para:

```

Try
    Dim vRetorno As String = ""
    Select Case TipoBanco
        Case "SQLSERVER"
            vRetorno =
                ManutencaoClienteBD.DescarregaContasCorrentes(Dt)
        Case "ORACLE"
            vRetorno =
                ManutencaoClienteBDOracle.DescarregaContasCorrentes(Dt)
        Case "MYSQL"
            vRetorno =
                ManutencaoClienteBDMySQL.DescarregaContasCorrentes(Dt)
    End Select

```

```

.....
MsgBox(vRetorno)
Catch ex As Exception
    MsgBox(ex.Message)
End Try

```

16) Altere o formulário `Frm_ManutençãoClienteDB`, adicionando um novo *Label*, com `Name` `Lbl_Database`.

17) No código-fonte deste formulário, inicialize a propriedade `Text` do novo *Label*, dentro do método `New`:

```
Lbl_Database.Text = "Banco usado: " + TipoBanco
```

18) No método `AtualizarListView()` modifique de:

```

Dim Dt As New DataTable
Select Case TipoBanco
    Case "SQLSERVER"
        Dt = ManutencaoClienteBD.RetornaClientes()
    Case "ORACLE"
        Dt = ManutencaoClienteBDOracle.RetornaClientes()
End Select

```

Para:

```

Dim Dt As New DataTable
Select Case TipoBanco
    Case "SQLSERVER"
        Dt = ManutencaoClienteBD.RetornaClientes()
    Case "ORACLE"
        Dt = ManutencaoClienteBDOracle.RetornaClientes()
    Case "MYSQL"
        Dt = ManutencaoClienteBDMySQL.RetornaClientes()
End Select

```

19) No código do evento `Click` do botão **Novo** do menu, modifique de:

```

Dim vRetorno As String = ""
Select Case TipoBanco
    Case "SQLSERVER"
        vRetorno =
            ManutencaoClienteBD.EditaCliente(Txt_CPF.Text, Txt_Nome.Text, txt_Profissao.Text)
    Case "ORACLE"
        vRetorno =
            ManutencaoClienteBDOracle.EditaCliente(Txt_CPF.Text, Txt_Nome.Text, txt_Profissao.Text)
End Select
MsgBox(vRetorno)

```

Para:

```

Dim vRetorno As String = ""
Select Case TipoBanco
    Case "SQLSERVER"
        vRetorno =
            ManutencaoClienteBD.EditaCliente(Txt_CPF.Text, Txt_Nome.Text, txt_Profissao.Text)
    Case "ORACLE"
        vRetorno =
            ManutencaoClienteBDOracle.EditaCliente(Txt_CPF.Text, Txt_Nome.Text, txt_Profissao.Text)
    Case "MYSQL"
        vRetorno =
            ManutencaoClienteBDMySQL.EditaCliente(Txt_CPF.Text, Txt_Nome.Text, txt_Profissao.Text)
End Select
MsgBox(vRetorno)

```

20) No código do evento Click do botão **Excluir**, do menu, modifique de:

```

Dim vRetorno As String = ""
Select Case TipoBanco
    Case "SQLSERVER"
        vRetorno = ManutencaoClienteBDOracle.ExcluiCliente(Txt_CPF.Text)
    Case "ORACLE"
        vRetorno = ManutencaoClienteBDOracle.ExcluiCliente(Txt_CPF.Text)
End Select
MsgBox(vRetorno)

```

Para:

```

Dim vRetorno As String = ""
Select Case TipoBanco
    Case "SQLSERVER"
        vRetorno = ManutencaoClienteBDOracle.ExcluiCliente(Txt_CPF.Text)
    Case "ORACLE"
        vRetorno = ManutencaoClienteBDOracle.ExcluiCliente(Txt_CPF.Text)
    Case "MYSQL"
        vRetorno = ManutencaoClienteBDMySQL.ExcluiCliente(Txt_CPF.Text)
End Select
MsgBox(vRetorno)

```

21) No formulário `Frm_Principal01`, no menu, dentro de **Cadastro BD**, inclua dois novos itens de menu, dentro de **Cadastro BD**, chamado **Manutenção de Clientes BD - MySQL** e **Manutenção de Contas Correntes BD - MySQL**.

22) No clique do item do menu **Manutenção de Clientes BD - MySQL**, inclua o código:

```

Private Sub ManutençãoDeClientesBDMySQLToolStripMenuItem_Click(sender As Object, e As EventArgs)
    Handles ManutençãoDeClientesBDMySQLToolStripMenuItem.Click
    Dim F As New Frm_ManutencaoClienteDB("MYSQL")
    F.MdiParent = Me
    F.Show()
End Sub

```

23) E no clique do item do menu **Manutenção de Contas Correntes BD - MySQL**, inclua o código:

```
Private Sub ManutençãoDeContasCorrentesBDMySQLToolStripMenuItem1_Click(sender As Object, e As EventArgs) Handles ManutençãoDeContasCorrentesBDMySQLToolStripMenuItem1.Click
    Dim F As New Frm_ManutencaoCC_BD("MYSQL")
    F.MdiParent = Me
    F.Show()
End Sub
```