



Programacion Orientada A Objetos II

Guia Practica N° 03

COMPETENCIAS:

- Implementa conexión a la base de datos con SQL Server usando el pool de conexiones.
- Implementa consultas a SQL Server usando la clase SqlCommand.
- Implementa el mantenimiento a la tabla Cliente usando un menu de opciones (Listado-Nuevo-Actualiza y Elimina)

Caso N° 01: Mantenimiento de Clientes

Aplicación que permite realizar el mantenimiento de los Clientes de la base de datos Negocios, usando la capanegocios.

Procedimientos Almacenados para la aplicación:

```
USE NEGOCIOS2014
GO
--1. LISTADO DE CLIENTES
IF OBJECT_ID('SP_LISTACLIENTES') IS NOT NULL
    DROP PROC SP_LISTACLIENTES
GO

CREATE PROC SP_LISTACLIENTES
AS
    SELECT C.IDCLIENTE AS CODIGO,
           C.NOMCLIENTE AS CLIENTE,
           C.DIRCLIENTE AS DIRECCION,
           P.NOMBREPAIS AS PAIS,
           C.FONOCIENTE AS FONO
    FROM VENTAS.CLIENTES C
    JOIN VENTAS.PAISES P ON C.IDPAIS=P.IDPAIS
GO

--2. LISTADO DE PAISES
IF OBJECT_ID('SP_LISTAPAIS') IS NOT NULL
    DROP PROC SP_LISTAPAIS
GO

CREATE PROC SP_LISTAPAIS
AS
    SELECT * FROM VENTAS.PAISES
```

```
GO
--3. INSERTAR UN NUEVO CLIENTE
IF OBJECT_ID('SP_INSERTACLIENTE') IS NOT NULL
    DROP PROC SP_INSERTACLIENTE
GO

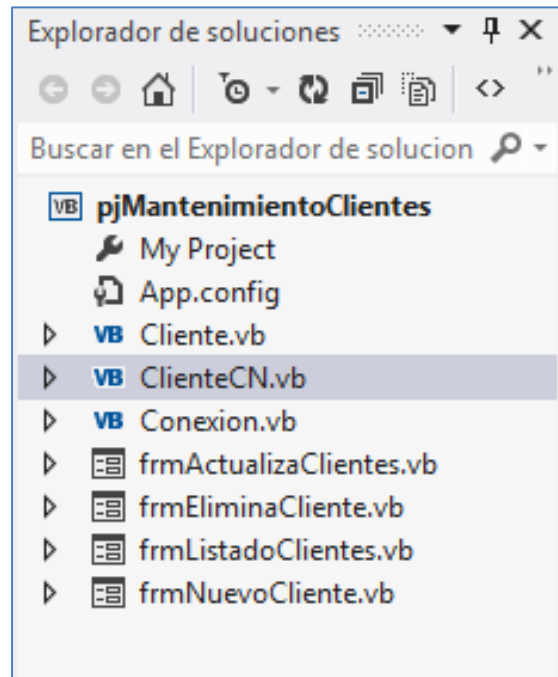
CREATE PROC SP_INSERTACLIENTE(@IDC CHAR(5),@NOM VARCHAR(40),
                             @DIR VARCHAR(40),@PAI CHAR(3),
                             @TEL CHAR(15))
AS
    INSERT INTO VENTAS.CLIENTES VALUES (@IDC,@NOM,@DIR,@PAI,@TEL)
GO
--4. BUSCAR UN CLIENTE
IF OBJECT_ID('SP_BUSCACLIENTE') IS NOT NULL
    DROP PROC SP_BUSCACLIENTE
GO

CREATE PROC SP_BUSCACLIENTE(@IDC CHAR(5))
AS
    SELECT * FROM VENTAS.CLIENTES
           WHERE IDCLIENTE=@IDC
GO
--5. ACTUALIZACION DE CLIENTE
IF OBJECT_ID('SP_ACTUALIZACLIENTE') IS NOT NULL
    DROP PROC SP_ACTUALIZACLIENTE
GO

CREATE PROC SP_ACTUALIZACLIENTE(@IDC CHAR(5),@NOM VARCHAR(40),
                                @DIR VARCHAR(40),@PAI CHAR(3),
                                @TEL CHAR(15))
AS
    UPDATE VENTAS.CLIENTES
           SET NOMCLIENTE=@NOM,DIRCLIENTE=@DIR,
               IDPAIS=@PAI,FONOCIENTE=@TEL
           WHERE IDCLIENTE=@IDC
GO
--6. ELIMINAR UN CLIENTE
IF OBJECT_ID('SP_ELIMINACLIENTE') IS NOT NULL
    DROP PROC SP_ELIMINACLIENTE
GO

CREATE PROC SP_ELIMINACLIENTE(@IDC CHAR(5))
AS
    DELETE VENTAS.CLIENTES
           WHERE IDCLIENTE=@IDC
GO
```

El proyecto debe contener los siguientes archivos:



Donde:

App.Config	Configura la conexión a la BD Negocios.
Cliente	Clase que contiene los atributos privados y metodos GET-SET de los Clientes, este se puede implementar creando un Diagrama de Clase (ClassDiagrama1.cd).
ClienteCN	Clase que contiene los metodos de los Clientes (ListaClientes, NuevoCliente, BuscaCliente, EliminaCliente, ActualizaCliente, etc).
Conexión	Clase que contiene el metodo getConecta que se asocia con app.config para determinar con que base de datos se esta trabajando, para nuestro caso Negocios.
frmListadoClientes	Formulario que permite mostrar una lista de los clientes.
frmNuevoCliente	Formulario que permite registrar al nuevo cliente.
frmActualizaClientes	Formulario que permite actualizar los valores de un determinado cliente.
frmEliminaCliente	Formulario que permite eliminar un determinado cliente.

Script del archivo: App.Config

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <connectionStrings>
    <add name="cn"
          connectionString="server=.;database=negocios2014;uid=sa;pwd=sql"/>
  </connectionStrings>
</configuration>
```

Agregar la librería System.Configuration al Proyecto

Script de la clase Conexión.vb

```
Imports System.Data.SqlClient
Imports System.Configuration.ConfigurationManager
Public Class Conexion
    Public Function getConecta() As SqlConnection
        Dim cn As New SqlConnection(ConnectionString("cn").ConnectionString)
        Return cn
    End Function
End Class
```

Script de la clase Cliente.vb

```
Public Class Cliente
    Private _Codigo As String
    Private _Cliente As String
    Private _Direccion As String
    Private _Pais As String
    Private _Fono As String

    Public Property Codigo As String
        Get
            Return _Codigo
        End Get
        Set(value As String)
            _Codigo = value
        End Set
    End Property

    Public Property Cliente As String
        Get
            Return _Cliente
        End Get
        Set(value As String)
            _Cliente = value
        End Set
    End Property

    Public Property Direccion As String
        Get
            Return _Direccion
        End Get
        Set(value As String)
            _Direccion = value
        End Set
    End Property
```

```
Public Property Pais As String
    Get
        Return _Pais
    End Get
    Set(value As String)
        _Pais = value
    End Set
End Property

Public Property Fono As String
    Get
        Return _Fono
    End Get
    Set(value As String)
        _Fono = value
    End Set
End Property
End Class
```

GUI para Listado de los Clientes



Script de la clase ClienteCN.vb

```
Imports System.Data.SqlClient
Public Class ClienteCN
    Dim objCon As New Conexion
    Dim cn As New SqlConnection

    'Listar los clientes
```

```
Public Function listadoClientes() As DataTable
    cn = objCon.getConecta
    Using da As New SqlDataAdapter("sp_listaclientes", cn)
        Dim dt As New DataTable
        da.Fill(dt)
        Return dt
    End Using
End Function
End Sub
```

Script de la clase frmListadoClientes

```
Public Class frmListadoClientes
    Dim objNeg As New ClienteCN

    Private Sub frmListadoClientes_Load(...) Handles MyBase.Load
        dgClientes.DataSource = objNeg.listadoClientes
        lblTotal.Text = objNeg.listadoClientes.Rows.Count
        dgClientes.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.AllCells
    End Sub
End Class
```

GUI para el registro del nuevo Cliente

Script de la clase ClienteCN.vb: Se debe agregar los metodos ListadoPais e InsertaCliente a la clase ClienteCN.

```
'Listado de Países
Public Function listadoPais() As DataTable
    cn = objCon.getConecta
    Using cmd As New SqlCommand
        cmd.CommandType = CommandType.StoredProcedure
        cmd.Connection = cn
```

```
cmd.CommandText = "SP_LISTAPAIS"

Using da As New SqlDataAdapter
    da.SelectCommand = cmd
    Dim dt As New DataTable
    da.Fill(dt)
    Return dt
End Using
End Using
End Function

'Insercion del nuevo Cliente
Public Sub insertaCliente(ByVal objCli As Cliente)
    cn = objCon.getConecta
    cn.Open()
    Using cmd As New SqlCommand
        cmd.Connection = cn
        cmd.CommandType = CommandType.StoredProcedure
        cmd.CommandText = "SP_INSERTACLIENTE"
        With cmd.Parameters
            .Add("@idc", SqlDbType.Char).Value = objCli.Codigo
            .Add("@nom", SqlDbType.VarChar).Value = objCli.Cliente
            .Add("@dir", SqlDbType.VarChar).Value = objCli.Direccion
            .Add("@pai", SqlDbType.Char).Value = objCli.Pais
            .Add("@tel", SqlDbType.Char).Value = objCli.Fono
        End With
        Try
            cmd.ExecuteNonQuery()
            MessageBox.Show("Cliente registrado correctamente", "Sistema",
                MessageBoxButtons.OK, MessageBoxIcon.Information)
        Catch ex As Exception
            MessageBox.Show(ex.Message, "Sistema",
                MessageBoxButtons.OK, MessageBoxIcon.Error)
        Finally
            cn.Close()
        End Try
    End Using
End Sub
```

Script de la clase frmNuevoCliente

```
Public Class frmNuevoCliente
    Dim objNeg As New ClienteCN

    Private Sub frmNuevoCliente_Load(...) Handles MyBase.Load
        Call llenaPais()
    End Sub

    Sub llenaPais()
        cboPais.DataSource = objNeg.listadoPais
    End Sub
End Class
```

```

        cboPais.DisplayMember = "nombrePais"
        cboPais.ValueMember = "idpais"
    End Sub

    Private Sub btnRegistrar_Click(...) Handles btnRegistrar.Click
        'Datos enviados a la clase Cliente
        Dim objCli As New Cliente
        objCli.Codigo = txtCodigo.Text
        objCli.Cliente = txtCliente.Text
        objCli.Direccion = txtDireccion.Text
        objCli.Pais = cboPais.SelectedValue
        objCli.Fono = txtFono.Text

        'Registrarlo
        objNeg.insertaCliente(objCli)
    End Sub
End Class

```

GUI para la Actualización de datos del Cliente

Script de la clase ClienteCN.vb: Se debe agregar los metodos buscaCliente y ActualizaCliente en la clase ClienteCN.

```

Public Function buscaCliente(ByVal objCli As Cliente) As DataTable
    cn = objCon.getConecta
    cn.Open()
    Using cmd As New SqlCommand
        cmd.Connection = cn
        cmd.CommandType = CommandType.StoredProcedure
        cmd.CommandText = "SP_BUSCACLIENETE"
        cmd.Parameters.Add("@IDC", SqlDbType.Char).Value = objCli.Codigo
    Using da As New SqlDataAdapter
        da.SelectCommand = cmd
    Using dt As New DataTable

```



```
        da.Fill(dt)
        Return dt
    End Using
End Using
End Using
End Function

'Metodo que actualiza los datos del cliente
Public Sub actualizaCliente(ByVal objCli As Cliente)
    cn = objCon.getConecta
    cn.Open()
    Using cmd As New SqlCommand
        cmd.Connection = cn
        cmd.CommandType = CommandType.StoredProcedure
        cmd.CommandText = "SP_ACTUALIZACLIENTE"
        With cmd.Parameters
            .Add("@idc", SqlDbType.Char).Value = objCli.Codigo
            .Add("@nom", SqlDbType.VarChar).Value = objCli.Cliente
            .Add("@dir", SqlDbType.VarChar).Value = objCli.Direccion
            .Add("@pai", SqlDbType.Char).Value = objCli.Pais
            .Add("@tel", SqlDbType.Char).Value = objCli.Fono
        End With
        Try
            cmd.ExecuteNonQuery()
            MessageBox.Show("Cliente actualizado correctamente",
                "Sistema", MessageBoxButtons.OK, MessageBoxIcon.Information)
        Catch ex As Exception
            MessageBox.Show(ex.Message, "Sistema",
                MessageBoxButtons.OK, MessageBoxIcon.Error)
        Finally
            cn.Close()
        End Try
    End Using
End Sub
```

Script de la clase frmActualizaClientes

```
Public Class frmActualizaClientes
    Dim objNeg As New ClienteCN
    Dim objCli As New Cliente

    Private Sub btnBuscar_Click(...) Handles btnBuscar.Click

        objCli.Codigo = txtCodigo.Text

        txtCliente.Text = objNeg.buscaCliente(objCli).
            Rows(0).Item("nomCliente")
        txtDireccion.Text = objNeg.buscaCliente(objCli).
            Rows(0).Item("dirCliente")
        cboPais.SelectedValue = objNeg.buscaCliente(objCli).
```

```
Rows(0).Item("idPais")
txtFono.Text = objNeg.buscaCliente(objCli).
Rows(0).Item("fonoCliente")

End Sub

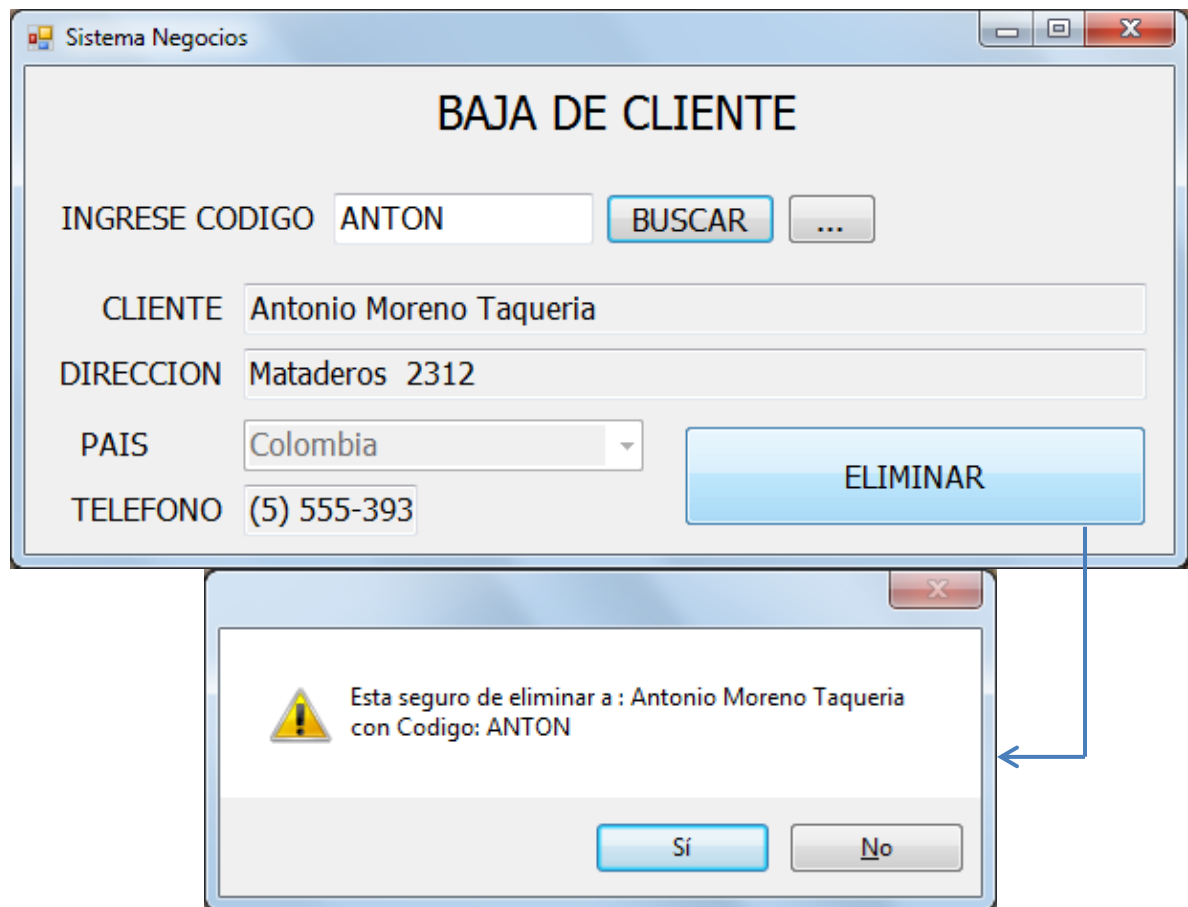
Private Sub frmActualizaClientes_Load(...) Handles MyBase.Load
    Call llenaPais()
End Sub

Sub llenaPais()
    cboPais.DataSource = objNeg.listadoPais
    cboPais.DisplayMember = "nombrePais"
    cboPais.ValueMember = "idPais"
End Sub

Private Sub btnActualizar_Click(...) Handles btnActualizar.Click
    'Datos enviados a la clase Cliente
    objCli.Codigo = txtCodigo.Text
    objCli.Cliente = txtCliente.Text
    objCli.Direccion = txtDireccion.Text
    objCli.Pais = cboPais.SelectedValue
    objCli.Fono = txtFono.Text

    'Registrar la actualizacion
    objNeg.actualizaCliente(objCli)
End Sub
End Class
```

GUI para la eliminacion del Cliente



Script de la clase ClienteCN.vb: Se debe agregar el metodo EliminaCliente a la clase ClienteCN.

```
'Baja de Cliente
Public Sub EliminaCliente(ByVal objCli As Cliente)
    cn = objCon.getConecta
    cn.Open()
    Using cmd As New SqlCommand
        cmd.Connection = cn
        cmd.CommandType = CommandType.StoredProcedure
        cmd.CommandText = "SP_ELIMINACLIENTE"
        cmd.Parameters.Add("@idc", SqlDbType.Char).Value = objCli.Codigo
    Try
        cmd.ExecuteNonQuery()
        MessageBox.Show("Cliente eliminado correctamente",
                        "Sistema", MessageBoxButtons.OK,
                        MessageBoxIcon.Information)
    Catch ex As Exception
        MessageBox.Show(ex.Message, "Sistema",
                        MessageBoxButtons.OK, MessageBoxIcon.Information)
```

```
Finally
    cn.Close()
End Try
End Using
End Sub
```

Script de la clase frmEliminaCliente

```
Dim objNeg As New ClienteCN
Dim objCli As New Cliente

Private Sub frmEliminaCliente_Load(...) Handles MyBase.Load
    cboPais.DataSource = objNeg.listadoPais
    cboPais.DisplayMember = "nombrePais"
    cboPais.ValueMember = "idPais"
End Sub

Private Sub btnBuscar_Click(...) Handles btnBuscar.Click
    objCli.Codigo = txtCodigo.Text

    txtCliente.Text = objNeg.buscaCliente(objCli).Rows(0).Item("nomCliente")
    txtDireccion.Text = objNeg.buscaCliente(objCli).Rows(0).Item("dirCliente")
    cboPais.SelectedValue = objNeg.buscaCliente(objCli).Rows(0).Item("idPais")
    txtFono.Text = objNeg.buscaCliente(objCli).Rows(0).Item("fonoCliente")
End Sub

Private Sub btnEliminar_Click(...) Handles btnEliminar.Click
    Dim r As Integer
    r = MessageBox.Show("Esta seguro de eliminar a : " & txtCliente.Text &
        vbNewLine & "con Codigo: " & txtCodigo.Text, "",
        MessageBoxButtons.YesNo, MessageBoxIcon.Warning)

    If r = 6 Then
        objCli.Codigo = txtCodigo.Text
        objNeg.EliminaCliente(objCli)
    End If
End Sub
```