



Programación Orientada A Objetos II

Guía Práctica N° 09

LinQ

COMPETENCIAS:

- El alumno implementara aplicaciones de Mantenimiento de Registros usando LINQ to SQL.

CASO 1: Mantenimiento de Productos

Implemente una aplicación que permita realizar un mantenimiento de la tabla Productos.

Tener en cuenta:

- Implementar procedimientos almacenados que permitan listar los productos, proveedores y categorías.
- Deberá agregar productos con el código de producto de forma automática.
- Deberá modificar los productos previamente seleccionado desde el control DataGridView.
- Deberá eliminar un determinado producto seleccionado desde el control DataGridView.

	IdProducto	NomProducto	IdProveedor	IdCategoria	CantxUnidad	Pre
1	1	Te Dharamsala	1	1	10 cajas x 20 bolsas	18
2	2	Cerveza tibetana Barley	1	1	24 - bot. 12 l	19
3	3	Sirope de regaliz	1	2	12 - bot. 550 ml	10
4	4	Espicias Cajun del chef Anton	2	2	48 - frascos 6 l	22
5	5	Mezcla Gumbo del chef Anton	2	2	36 cajas	21
6	6	Mermelada de grosellas de la abuela	3	2	12 - frascos 8 l	25
7	7	Peras secas organicas del tio Bob	3	7	12 - paq. 1 kg	30

Pasos:

1. Implementar los siguientes procedimientos almacenados en SQL Server 2012.

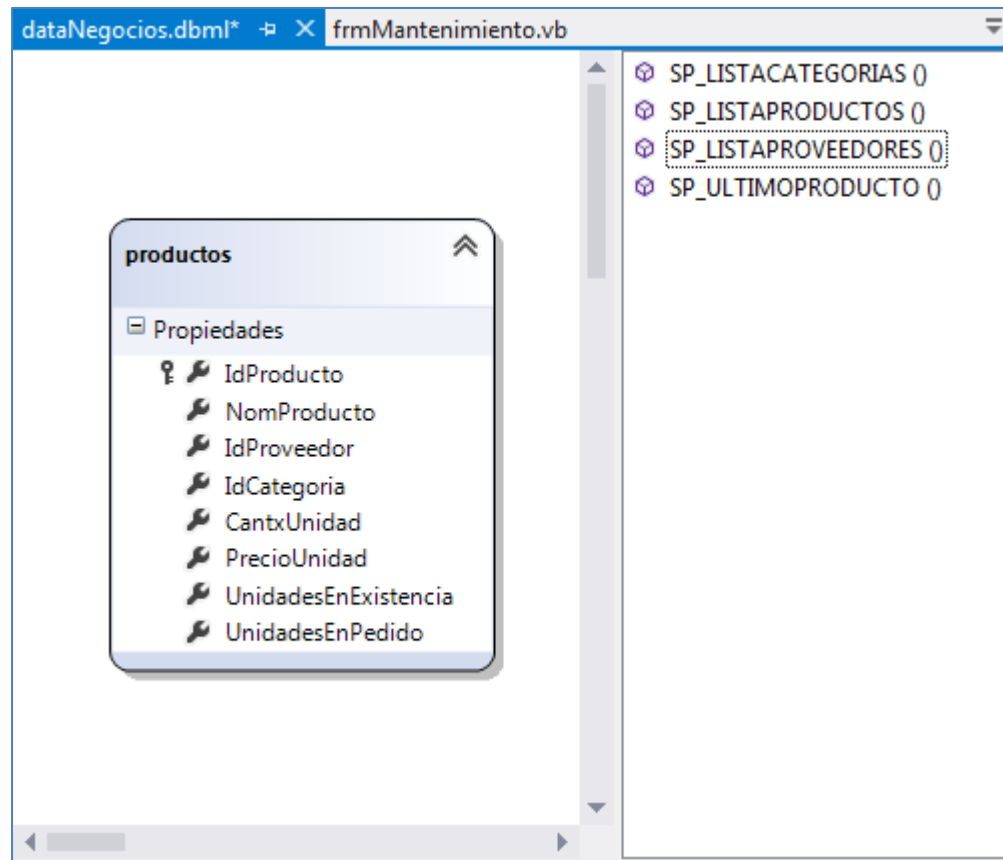
```
IF OBJECT_ID('SP_LISTAPRODUCTOS') IS NOT NULL
BEGIN
    DROP PROC SP_LISTAPRODUCTOS
END
GO
CREATE PROC SP_LISTAPRODUCTOS
AS
    SELECT * FROM Compras.productos
GO

IF OBJECT_ID('SP_LISTAPROVEEDORES') IS NOT NULL
BEGIN
    DROP PROC SP_LISTAPROVEEDORES
END
GO
CREATE PROC SP_LISTAPROVEEDORES
AS
    SELECT * FROM Compras.proveedores
GO

IF OBJECT_ID('SP_LISTACATEGORIAS') IS NOT NULL
BEGIN
    DROP PROC SP_LISTACATEGORIAS
END
GO
CREATE PROC SP_LISTACATEGORIAS
AS
    SELECT * FROM Compras.categorias
GO

IF OBJECT_ID('SP_ULTIMOPRODUCTO') IS NOT NULL
BEGIN
    DROP PROC SP_ULTIMOPRODUCTO
END
GO
CREATE PROC SP_ULTIMOPRODUCTO
AS
    SELECT TOP 1 IdProducto FROM Compras.productos
        ORDER BY IdProducto DESC
GO
```

2. Crear un nuevo proyecto llamado pjMantenimientoProductos
3. Agregar un nuevo elemento del tipo Clases de LinQ to SQL, llamarlo dNegocios y agregar los procedimientos almacenados implementados además de la tabla Productos de tal forma que el entorno se muestre de la siguiente manera:



4. Diseñar la siguiente GUI

5. Script de la aplicación:

```
Public Class frmMantenimiento
    Dim dNegocios As New dataNegociosDataContext
    Private Sub frmMantenimiento_Load(sender As Object, e As EventArgs)
        Handles MyBase.Load
            Call muestraProductos()
            Call muestraProveedores()
            Call muestraCategoria()
            Call generaCodigo()
        End Sub

        Sub muestraProductos()
            dgProductos.DataSource = dNegocios.SP_LISTAPRODUCTOS
        End Sub

        Sub muestraProveedores()
            cboProveedor.DataSource = dNegocios.SP_LISTAPROVEEDORES
            cboProveedor.DisplayMember = "nomProveedor"
            cboProveedor.ValueMember = "idProveedor"
        End Sub

        Sub muestraCategoria()
            cboCategoria.DataSource = dNegocios.SP_LISTACATEGORIAS
            cboCategoria.DisplayMember = "nombreCategoria"
        End Sub
    End Class
```

```
cboCategoria.ValueMember = "idCategoria"
End Sub

Sub generaCodigo()
    lblCodigo.Text = dNegocios.SP_ULTIMOPRODUCTO.First.IdProducto + 1
End Sub

Sub limpiarControles()
    Dim misObjetos As Control

    For Each misObjetos In Me.Controls
        If TypeOf misObjetos Is System.Windows.Forms.TextBox Then
            misObjetos.Text = ""
        End If
    Next
End Sub

Private Sub tsAgregar_Click(sender As Object, e As EventArgs)
    Handles tsAgregar.Click

    Call generaCodigo()
    Dim objPro As New productos
    objPro.IdProducto = lblCodigo.Text
    objPro.NomProducto = txtDescripcion.Text
    objPro.IdProveedor = cboProveedor.SelectedValue
    objPro.IdCategoria = cboCategoria.SelectedValue
    objPro.CantxUnidad = txtCantidad.Text
    objPro.PrecioUnidad = txtPrecio.Text
    objPro.UnidadesEnExistencia = txtExistencia.Text
    objPro.UnidadesEnPedido = txtPedido.Text

    dNegocios.productos.InsertOnSubmit(objPro)
    dNegocios.SubmitChanges()
    Call muestraProductos()
    Call generaCodigo()
    Call limpiarControles()
End Sub

Private Sub dgProductos_CellContentClick(sender As Object, e As
DataGridViewCellEventArgs) Handles dgProductos.CellContentClick
    lblCodigo.Text = dgProductos.CurrentRow.Cells(0).Value
    txtDescripcion.Text = dgProductos.CurrentRow.Cells(1).Value
    cboProveedor.SelectedValue=dgProductos.CurrentRow.Cells(2).Value
    cboCategoria.SelectedValue=dgProductos.CurrentRow.Cells(3).Value
    txtCantidad.Text = dgProductos.CurrentRow.Cells(4).Value
    txtPrecio.Text = dgProductos.CurrentRow.Cells(5).Value
    txtExistencia.Text = dgProductos.CurrentRow.Cells(6).Value
    txtPedido.Text = dgProductos.CurrentRow.Cells(7).Value
End Sub

Private Sub tsModificar_Click(sender As Object, e As EventArgs)
    Handles tsModificar.Click
```

```
Dim producto = From pro In dNegocios.productos
                Where pro.IdProducto = lblCodigo.Text

Dim objPro As New productos
objPro = producto.First
objPro.IdProducto = lblCodigo.Text
objPro.NomProducto = txtDescripcion.Text
objPro.IdProveedor = cboProveedor.SelectedValue
objPro.IdCategoria = cboCategoria.SelectedValue
objPro.CantxUnidad = txtCantidad.Text
objPro.PrecioUnidad = txtPrecio.Text
objPro.UnidadesEnExistencia = txtExistencia.Text
objPro.UnidadesEnPedido = txtPedido.Text
dNegocios.SubmitChanges()
Call muestraProductos()
End Sub

Private Sub tsEliminar_Click(sender As Object, e As EventArgs)
    Handles tsEliminar.Click
    Dim producto = From pro In dNegocios.productos
                    Where pro.IdProducto = lblCodigo.Text

    dNegocios.productos.DeleteOnSubmit(producto.First)
    dNegocios.SubmitChanges()
    Call muestraProductos()
End Sub

Private Sub tsSalir_Click(sender As Object, e As EventArgs)
    Handles tsSalir.Click
    Me.Close()
End Sub
End Class
```