

## MANTENIMIENTO DE PRODUCTOS

	CODIGO	PRODUCTO	PRECIO	CANTIDAD	ESTADO
▶	P0001	TECLADO	15.00	250	1
	P0002	MONITOR	150.00	350	1
*					

DATOS DE PRODUCTOS

**CODIGO**

**PRODUCTO**

**PRECIO**

**CANTIDAD**

☒ Estado

### APP.CONFIG

```
<connectionStrings>
  <add name="Conex" connectionString="Server=.;DataBase=DATA;integrated security=true"/>
</connectionStrings>
```

### SCRIPT DE LA BASE DE DATOS

```
--Creación de la base de
--Datos: DATA
Create DataBase DATA
go

--Activar la Base de Datos
Use DATA
go

--Crear la tabla Productos
if OBJECT_ID('Productos') is not null
    drop table Productos
go
CREATE TABLE PRODUCTOS (
CODIGO varchar(5) PRIMARY KEY,
PRODUCTO VARCHAR(50) NOT NULL,
PRECIO NUMERIC(8,2) NOT NULL,
CANTIDAD INT NOT NULL,
ESTADO INT NOT NULL
)
GO

--INSERTAR REGISTROS
INSERT INTO PRODUCTOS VALUES ('P0001', 'TECLADO', 15, 250, 1)
INSERT INTO PRODUCTOS VALUES ('P0002', 'MONITOR', 150, 350, 1)
```

```

--P.A. que lista la tabla Productos
If OBJECT_ID('sp_listaproductos') is not null
    drop proc sp_listaproductos
go
Create Proc sp_listaproductos
as
Select * from Productos
WHERE ESTADO=1
go

--P.A para insertar Registros
If OBJECT_ID('SP_InsertaRegistros') is not null
    drop proc SP_InsertaRegistros
go
Create Proc SP_InsertaRegistros
@Codigo Varchar(5),
@Producto Varchar(50),
@Precio Numeric(8,2),
@Cantidad int,
@ESTADO INT
as
insert into productos
values (@Codigo,@Producto,@Precio,@Cantidad,@ESTADO)
go

--P.A que actualiza registros
If OBJECT_ID('SP_ActualizaRegistro') is not null
    drop proc SP_ActualizaRegistro
go
Create Proc SP_ActualizaRegistro
@Codigo Varchar(5),
@Producto Varchar(50),
@Precio numeric(8,2),
@Cantidad int,
@ESTADO INT
as
Update PRODUCTOS
set Producto=@Producto,PRECIO=@Precio,CANTIDAD=@Cantidad,
ESTADO=@ESTADO
where CODIGO=@Codigo
go

-- Autogenera el codigo del Producto
If OBJECT_ID('sp_genera') is not null
    drop proc sp_genera
go
Create procedure sp_genera
@cod varchar (20) output
as
declare @aux varchar(20)
select @aux = max(right(Codigo,4)) from Productos
select @cod = 'P'+right(str(convert(int,@aux)+10001),4)
--noten que en este procedimiento el parámetro @cod dice output que
quiere decir que es de salida.
go

```

```

--para probar nuestro sp seria de la siguiente manera
--(esto siempre y cuando el sp vote un parámetro de
--salida):
declare @aux varchar(20)
execute sp_genera @aux output
select @aux

--P.A para eliminar registros de
--forma Lógica(No Física)
if OBJECT_ID('usp_DardeBaja') is not null
    drop proc usp_DardeBaja
go
Create Proc usp_DardeBaja
@Codigo char(6)
as
Update PRODUCTOS set Estado=0
where CODIGO=@Codigo
go

```

## CLASE: EntProductos

```

Public Class EntProductos
    Private _Codigo As String
    Private _Producto As String
    Private _Precio As Double
    Private _Cantidad As Integer
    Private _Estado As Integer
    'Metodos Get - Set
    Public Property Codigo As String
        Get
            Return _Codigo
        End Get
        Set(ByVal value As String)
            _Codigo = value
        End Set
    End Property
    Public Property Producto As String
        Get
            Return _Producto
        End Get
        Set(ByVal value As String)
            _Producto = value
        End Set
    End Property
    Public Property Precio As Double
        Get
            Return _Precio
        End Get
        Set(ByVal value As Double)
            _Precio = value
        End Set
    End Property
    Public Property Cantidad As Integer
        Get
            Return _Cantidad
        End Get
        Set(ByVal value As Integer)
            _Cantidad = value
        End Set
    End Property

```

```

        End Get
        Set(ByVal value As Integer)
            _Cantidad = value
        End Set
    End Property
    Public Property Estado As Integer
        Get
            Return _Estado
        End Get
        Set(ByVal value As Integer)
            _Estado = value
        End Set
    End Property
End Class

```

## Clase: Conexion

```

Imports System.Data.SqlClient
Imports System.Configuration.ConfigurationManager
Public Function getConecta() As SqlConnection
    Dim cn As New SqlConnection _
        (ConnectionStrings("Conex").ConnectionString)
    Return cn
End Function

```

## Clase: NegProductos

```

Imports System.Data.SqlClient
Public Class NegProductos
    Dim ObjConex As New Conexion
    Dim cn As New SqlConnection

    Public Function ListaProductos() As DataTable
        cn = ObjConex.getConecta
        Using da As New SqlDataAdapter("sp_listaproductos", cn)
            Using dt As New DataTable
                da.Fill(dt)
                Return dt
            End Using
        End Using
    End Function

    Public Sub GrabarRegistro(ByVal ObjEntProd As EntProductos)
        cn = ObjConex.getConecta
        cn.Open()
        Using tr As SqlTransaction = cn.BeginTransaction(IsolationLevel.Serializable)
            Using cmd As New SqlCommand
                cmd.CommandType = CommandType.StoredProcedure
                cmd.Connection = cn
                cmd.CommandText = "SP_InsertaRegistros"
                cmd.Transaction = tr
                With cmd.Parameters
                    .Add("@Codigo", SqlDbType.VarChar).Value = ObjEntProd.Codigo
                    .Add("@Producto", SqlDbType.VarChar).Value = ObjEntProd.Producto
                    .Add("@Precio", SqlDbType.Decimal).Value = ObjEntProd.Precio
                    .Add("@Cantidad", SqlDbType.Int).Value = ObjEntProd.Cantidad
                End With
            End Using
        End Using
    End Sub

```

```

        .Add("@Estado", SqlDbType.Int).Value = ObjEntProd.Estado
    End With
    Try
        cmd.ExecuteNonQuery()
        tr.Commit()
        MessageBox.Show("SE REGISTRO CORRECTAMENTE!!!", "MENSAJE",
            MessageBoxButtons.OK, MessageBoxIcon.Information)
    Catch ex As Exception
        tr.Rollback()
        MessageBox.Show(ex.Message, "MENSAJE",
            MessageBoxButtons.OK, MessageBoxIcon.Error)
    Finally
        cn.Close()
    End Try
End Using
End Using
End Sub

```

```

Public Sub EliminaRegistro(ByVal ObjEntProd As EntProductos)
    cn = ObjConex.getConecta
    cn.Open()
    Using tr As SqlTransaction = cn.BeginTransaction(IsolationLevel.Serializable)
        Using cmd As New SqlCommand
            cmd.CommandType = CommandType.StoredProcedure
            cmd.Connection = cn
            cmd.CommandText = "usp_DardeBaja"
            cmd.Transaction = tr
            With cmd.Parameters
                .Add("@Codigo", SqlDbType.VarChar).Value = ObjEntProd.Codigo
            End With
            Try
                cmd.ExecuteNonQuery()
                tr.Commit()
                MessageBox.Show("SE ELIMINO CORRECTAMENTE!!!",
                    "MENSAJE", MessageBoxButtons.OK, MessageBoxIcon.Information)
            Catch ex As Exception
                tr.Rollback()
                MessageBox.Show(ex.Message, "MENSAJE",
                    MessageBoxButtons.OK, MessageBoxIcon.Error)
            Finally
                cn.Close()
            End Try
        End Using
    End Using
End Sub

```

```

Public Sub ActualizaRegistro(ByVal ObjEntProd As EntProductos)
    cn = ObjConex.getConecta
    cn.Open()
    Using tr As SqlTransaction = cn.BeginTransaction(IsolationLevel.Serializable)
        Using cmd As New SqlCommand
            cmd.CommandType = CommandType.StoredProcedure
            cmd.Connection = cn
            cmd.CommandText = "SP_ActualizaRegistro"
            cmd.Transaction = tr
            With cmd.Parameters
                .Add("@Codigo", SqlDbType.VarChar).Value = ObjEntProd.Codigo
                .Add("@Producto", SqlDbType.VarChar).Value = ObjEntProd.Producto
                .Add("@Precio", SqlDbType.VarChar).Value = ObjEntProd.Precio
            End With
        End Using
    End Using
End Sub

```

```

        .Add("@Cantidad", SqlDbType.VarChar).Value = ObjEntProd.Cantidad
        .Add("@Estado", SqlDbType.VarChar).Value = ObjEntProd.Estado
    End With
    Try
        cmd.ExecuteNonQuery()
        tr.Commit()
        MessageBox.Show("SE ACTUALIZO CORRECTAMENTE!!!", "MENSAJE",
        MessageBoxButtons.OK, MessageBoxIcon.Information)
    Catch ex As Exception
        MessageBox.Show(ex.Message, "MENSAJE",
        MessageBoxButtons.OK, MessageBoxIcon.Error)
        tr.Rollback()
    Finally
        cn.Close()
    End Try
End Using
End Using
End Sub

```

```

Public Function Autogenera_Codigo() As String
    cn = ObjConex.getConecta
    cn.Open()
    Using cmd As New SqlCommand
        cmd.CommandType = CommandType.StoredProcedure
        cmd.Connection = cn
        cmd.CommandText = "Sp_genera"
        With cmd.Parameters
            .Add("@Cod", SqlDbType.Char, 5).Direction = ParameterDirection.Output
        End With
        cmd.ExecuteNonQuery()
        Dim NewCodigo As String
        NewCodigo = cmd.Parameters(0).Value.ToString
        cn.Close()
        Return NewCodigo
    End Using
End Function

```

End Class

## CAPAPRESENTACION: FORMULARIO

```
Public Class Form1
```

```
#Region "Controles"
```

```
    Sub ListaProductos()
        dgProductos.DataSource = ObjNegProductos.ListaProductos
    End Sub
```

```
    Sub HabilitarTextBox()
        txtProducto.Enabled = True
        txtPrecio.Enabled = True
        txtCantidad.Enabled = True
    End Sub
```

```
    Sub DesHabilitarTextBox()
        txtProducto.Enabled = False
        txtPrecio.Enabled = False
        txtCantidad.Enabled = False
    End Sub
```

```
    Sub LimpiarTextBox()
        txtCodigo.Clear()
        txtCantidad.Clear()
    End Sub

```

```
        txtPrecio.Clear()
        txtProducto.Clear()
    End Sub
#End Region
```

```
Dim ObjNegProductos As New NegProductos
```

```
Private Sub Form1_Load(...) Handles MyBase.Load
    ListaProductos()
End Sub
```

```
Private Sub btnNuevo_Click(..) Handles btnNuevo.Click
    LimpiarTextBox()
    txtCodigo.Text = "Autogenerado"
    HabilitarTextBox()
    txtProducto.Focus()
    chkEstado.Enabled = False
End Sub
```

```
Private Sub btnGrabar_Click(...) Handles btnGrabar.Click
    Dim ObjEntProd As New EntProductos
    ObjEntProd.Codigo = ObjNegProductos.Autogenera_Codigo
    ObjEntProd.Producto = txtProducto.Text
    ObjEntProd.Precio = txtPrecio.Text
    ObjEntProd.Cantidad = txtCantidad.Text
    If chkEstado.Checked = True Then
        ObjEntProd.Estado = 1
    Else
        ObjEntProd.Estado = 0
    End If
    ObjNegProductos.GrabarRegistro(ObjEntProd)
    ListaProductos()
    DesHabilitarTextBox()
    LimpiarTextBox()
End Sub
```

```
Private Sub btnEliminar_Click(...) Handles btnEliminar.Click
    Dim ObjEntProd As New EntProductos
    ObjEntProd.Codigo = txtCodigo.Text
    ObjNegProductos.EliminaRegistro(ObjEntProd)
    ListaProductos()
End Sub
```

```
Private Sub dgProductos_CurrentCellChanged(..) Handles
    Try
        txtCodigo.Text = dgProductos.CurrentRow.Cells(0).Value
        txtProducto.Text = dgProductos.CurrentRow.Cells(1).Value
        txtPrecio.Text = dgProductos.CurrentRow.Cells(2).Value
        txtCantidad.Text = dgProductos.CurrentRow.Cells(3).Value
        If dgProductos.CurrentRow.Cells(4).Value = 1 Then
            chkEstado.Checked = True
        Else
            chkEstado.Checked = False
        End If
    Catch ex As Exception
```

```
End Try
End Sub
```

```
Private Sub btnActualizar_Click(..) Handles btnActualizar.Click
    If btnActualizar.Text = "MODIFICAR" Then
        btnActualizar.Text = "ACTUALIZAR"
        HabilitarTextBox()
    Else
        Dim ObjEntProd As New EntProductos
        ObjEntProd.Codigo = txtCodigo.Text
        ObjEntProd.Producto = txtProducto.Text
        ObjEntProd.Precio = txtPrecio.Text
        ObjEntProd.Cantidad = txtCantidad.Text
        If chkEstado.Checked = True Then
            ObjEntProd.Estado = 1
        Else
            ObjEntProd.Estado = 0
        End If
        ObjNegProductos.ActualizaRegistro(ObjEntProd)
        ListaProductos()
        btnActualizar.Text = "MODIFICAR"
        DesHabilitarTextBox()
    End If
End Sub
End Class
```