

Data Source=MARCOSRAMIREZPC\SQLEXPRESS; Initial Catalog=BD_FISDL; Integrated Security=true

"Server=Apolo; DataBase=BD_FISDL; Uid=sa; Password=123"

<http://www.microsoft.com/en-us/download/details.aspx?id=435>

TbUsuario

```
-- =====
-- Creado por: Marcos E. Ramirez
-- Fecha Creación: Junio de 2013
-- Modificado por:
-- Fecha Modificación:
-- Descripción: PruebaTecnica en el FISDL
-- =====
-- SP para seleccionar todos los datos de la tabla
-- =====

GO
IF EXISTS (SELECT name FROM sysobjects WHERE name = 'sp_Obtener_BD_FISDL_TbUsuario' AND
type = 'P')
    DROP PROCEDURE [dbo].[sp_Obtener_BD_FISDL_TbUsuario]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[sp_Obtener_BD_FISDL_TbUsuario]
AS
SELECT
    [IdUser]
    , [UserNombre]
    , [UserClave]
    , [UserCrea]
    , [FechaCrea]
    , [UltUserModi]
    , [UltiFechaModi]
```

```

FROM [dbo].[TbUsuario]
ORDER BY UserNombre
--

GO

-- =====
-- Creado por: Marcos E. Ramirez
-- Fecha Creación: Junio de 2013
-- Modificado por:
-- Fecha Modificación:
-- Descripción: PruebaTecnica en el FISDL
-- =====
-- SP para seleccionar un dato de la tabla
-- =====
IF EXISTS (SELECT name FROM sysobjects WHERE name = 'sp_Buscar_BD_FISDL_TbUsuario' AND
type = 'P')
    DROP PROCEDURE [dbo].[sp_Buscar_BD_FISDL_TbUsuario]

GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[sp_Buscar_BD_FISDL_TbUsuario]
    @idUser int
AS
SELECT
    [idUser]
    , [UserNombre]
    , [UserClave]
    , [UserCrea]
    , [FechaCrea]
    , [UltUserModi]
    , [UltiFechaModi]
FROM [dbo].[TbUsuario]
WHERE
    [idUser] = @idUser

-- =====
-- Creado por: Marcos E. Ramirez
-- Fecha Creación: Junio de 2013
-- Modificado por:
-- Fecha Modificación:
-- Descripción: PruebaTecnica en el FISDL
-- =====
-- SP para adicionar un registro a la tabla
-- =====
GO
IF EXISTS (SELECT name FROM sysobjects WHERE name = 'sp_Agregar_BD_FISDL_TbUsuario' AND
type = 'P')
    DROP PROCEDURE [dbo].[sp_Agregar_BD_FISDL_TbUsuario]

GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[sp_Agregar_BD_FISDL_TbUsuario]
    @UserNombre nvarchar(50),
    @UserClave nvarchar(50),

```

```

@UserCrea nvarchar(50),
@FechaCrea DateTime
AS
INSERT [dbo].[TbUsuario]
(
    [UserNombre],
    [UserClave],
    [UserCrea],
    [FechaCrea]

)
VALUES
(
    @UserNombre,
    @UserClave,
    @UserCrea,
@FechaCrea
)

-- =====
-- Creado por: Marcos E. Ramirez
-- Fecha Creación: Junio de 2013
-- Modificado por:
-- Fecha Modificación:
-- Descripción: PruebaTecnica en el FISDL
-- =====
-- SP para actualizar un registro a la tabla
-- =====
GO

IF EXISTS (SELECT name FROM sysobjects WHERE name = 'sp_Actualizar_BD_FISDL_TbUsuario'
AND type = 'P')
    DROP PROCEDURE [dbo].[sp_Actualizar_BD_FISDL_TbUsuario]

GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[sp_Actualizar_BD_FISDL_TbUsuario]
    @idUser int,
    @UserNombre nvarchar(50),
    @UserClave nvarchar(20),
    @UltUserModi nvarchar(20),
@UltiFechaModi DateTime
AS
UPDATE [dbo].[TbUsuario]
SET
    [UserNombre] = @UserNombre,
    [UserClave] = @UserClave,
    [UltUserModi] = @UltUserModi,
    [UltiFechaModi] = GETDATE()
WHERE
    [idUser] = @idUser

-- =====
-- Creado por: Marcos E. Ramirez
-- Fecha Creación: Junio de 2013

```

```

-- Modificado por:
-- Fecha Modificación:
-- Descripción: PruebaTecnica en el FISDL
-- =====
-- SP para borrar un registro a la tabla
-- =====
GO

IF EXISTS (SELECT name FROM sysobjects WHERE name = 'sp_Eliminar_BD_FISDL_TbUsuario' AND
type = 'P')
    DROP PROCEDURE [dbo].[sp_Eliminar_BD_FISDL_TbUsuario]

GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[sp_Eliminar_BD_FISDL_TbUsuario]
    @idUser int
AS
DELETE [dbo].[TbUsuario]
WHERE
    [idUser] = @idUser

--- Fin de archivo SQL

```

TbDepto

```

-- =====
-- Creado por: Marcos E. Ramirez
-- Fecha Creación: Junio de 2013
-- Modificado por:
-- Fecha Modificación:
-- Descripción: PruebaTecnica en el FISDL
-- =====
-- SP para seleccionar todos los datos de la tabla
-- =====
GO

IF EXISTS (SELECT name FROM sysobjects WHERE name = 'sp_Obtener_BD_FISDL_TbDepto' AND
type = 'P')
    DROP PROCEDURE [dbo].[sp_Obtener_BD_FISDL_TbDepto]

GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[sp_Obtener_BD_FISDL_TbDepto]
AS
SELECT
    [IdDepto]
    , [NombreDepto]
    , [UserCrea]

```

```

        , [FechaCrea]
        , [UltUserModi]
        , [UltFechaModi]
FROM [dbo].[TbDepto]
ORDER BY NombreDepto
--

GO
-- =====
-- Creado por: Marcos E. Ramirez
-- Fecha Creación: Junio de 2013
-- Modificado por:
-- Fecha Modificación:
-- Descripción: PruebaTecnica en el FISDL
-- =====
-- SP para seleccionar un dato de la tabla
-- =====
IF EXISTS (SELECT name FROM sysobjects WHERE name = 'sp_Buscar_BD_FISDL_TbDepto' AND type
= 'P')
    DROP PROCEDURE [dbo].[sp_Buscar_BD_FISDL_TbDepto]

GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[sp_Buscar_BD_FISDL_TbDepto]
    @IdDepto int
AS
SELECT
    [IdDepto]
    , [NombreDepto]
    , [UserCrea]
    , [FechaCrea]
    , [UltUserModi]
    , [UltFechaModi]
FROM [dbo].[TbDepto]
WHERE
    [IdDepto] = @IdDepto

-- =====
-- Creado por: Marcos E. Ramirez
-- Fecha Creación: Junio de 2013
-- Modificado por:
-- Fecha Modificación:
-- Descripción: PruebaTecnica en el FISDL
-- =====
-- SP para adicionar un registro a la tabla
-- =====
GO
IF EXISTS (SELECT name FROM sysobjects WHERE name = 'sp_Agregar_BD_FISDL_TbDepto' AND
type = 'P')
    DROP PROCEDURE [dbo].[sp_Agregar_BD_FISDL_TbDepto]

GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[sp_Agregar_BD_FISDL_TbDepto]

```

```

        @NombreDepto nvarchar(50),
        @UserCrea nvarchar(50),
@FechaCrea DateTime,
@UltUserModi nvarchar(50),
@UltFechaModi DateTime
AS
INSERT [dbo].[TbDepto]
(
    [NombreDepto],
    [UserCrea],
    [FechaCrea]
)
VALUES
(
    @NombreDepto,
    @UserCrea,
    @FechaCrea
)

-- =====
-- Creado por: Marcos E. Ramirez
-- Fecha Creación: Junio de 2013
-- Modificado por:
-- Fecha Modificación:
-- Descripción: PruebaTecnica en el FISDL
-- =====
-- SP para actualizar un registro a la tabla
-- =====
GO

IF EXISTS (SELECT name FROM sysobjects WHERE name = 'sp_Actualizar_BD_FISDL_TbDepto' AND
type = 'P')
    DROP PROCEDURE [dbo].[sp_Actualizar_BD_FISDL_TbDepto]

GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[sp_Actualizar_BD_FISDL_TbDepto]
    @IdDepto int,
    @NombreDepto nvarchar(50),
    @UltUserModi nvarchar(50),
    @UltFechaModi DateTime
AS
UPDATE [dbo].[TbUDepto]
SET
    [NombreDepto] = @NombreDepto,
    [UltUserModi] = @UltUserModi,
    [UltiFechaModi] = @UltiFechaModi
WHERE
    [IdDepto] = @IdDepto

-- =====
-- Creado por: Marcos E. Ramirez
-- Fecha Creación: Junio de 2013
-- Modificado por:

```

```

-- Fecha Modificación:
-- Descripción: PruebaTecnica en el FISDL
-- =====
-- SP para borrar un registro a la tabla
-- =====
GO

IF EXISTS (SELECT name FROM sysobjects WHERE name = 'sp_Eliminar_BD_FISDL_TbDepto' AND
type = 'P')
    DROP PROCEDURE [dbo].[sp_Eliminar_BD_FISDL_TbDepto]

GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[sp_Eliminar_BD_FISDL_TbDepto]
    @IdDepto int
AS
DELETE [dbo].[TbDepto]
WHERE
    [IdDepto] = @IdDepto

--- Fin de archivo SQL

```

TbEmpleado

```

-- =====
-- Creado por: Marcos E. Ramirez
-- Fecha Creación: Junio de 2013
-- Modificado por:
-- Fecha Modificación:
-- Descripción: PruebaTecnica en el FISDL
-- =====
-- SP para seleccionar todos los datos de la tabla
-- =====

GO
IF EXISTS (SELECT name FROM sysobjects WHERE name = 'sp_Obtener_BD_FISDL_TbEmpleado' AND
type = 'P')
    DROP PROCEDURE [dbo].[sp_Obtener_BD_FISDL_TbEmpleado]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[sp_Obtener_BD_FISDL_TbEmpleado]
AS
SELECT
[IdEmpleado]
    , [NombreEmpleado]
    , [SexoEmpleado]
    , [FechaNacEmpleado]
    , [EstadCivilEmpleado]

```

```

, [DireccionEmpleado]
, [IdDepto]
, [TelefonoEmpleado]
, [EmailEmpleado]
, [idUser]
, [ImagenEmpleado]
, [UserCrea]
, [FechaCrea]
, [UltUserModi]
, [UltiFechaModi]

FROM [dbo].[TbEmpleado]
ORDER BY NombreEmpleado
--

GO
-- =====
-- Creado por: Marcos E. Ramirez
-- Fecha Creación: Junio de 2013
-- Modificado por:
-- Fecha Modificación:
-- Descripción: PruebaTecnica en el FISDL
-- =====
-- SP para seleccionar un dato de la tabla
-- =====
IF EXISTS (SELECT name FROM sysobjects WHERE name = 'sp_Buscar_BD_FISDL_TbEmpleado' AND
type = 'P')
    DROP PROCEDURE [dbo].[sp_Buscar_BD_FISDL_TbEmpleado]

GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[sp_Buscar_BD_FISDL_TbEmpleado]
    @NombreEmpleado nvarchar(50)
AS
SELECT
    [IdEmpleado]
, [NombreEmpleado]
, [SexoEmpleado]
, [FechaNacEmpleado]
, [EstadCivilEmpleado]
, [DireccionEmpleado]
, [IdDepto]
, [TelefonoEmpleado]
, [EmailEmpleado]
, [idUser]
, [ImagenEmpleado]
, [UserCrea]
, [FechaCrea]
, [UltUserModi]
, [UltiFechaModi] FROM [dbo].[TbEmpleado]
WHERE
    [NombreEmpleado] = @NombreEmpleado

-- =====
-- Creado por: Marcos E. Ramirez
-- Fecha Creación: Junio de 2013
-- Modificado por:
-- Fecha Modificación:
-- Descripción: PruebaTecnica en el FISDL

```

```

-- =====
-- SP para adicionar un registro a la tabla
-- =====
GO
IF EXISTS (SELECT name FROM sysobjects WHERE name = 'sp_Agregar_BD_FISDL_TbEmpleado' AND
type = 'P')
    DROP PROCEDURE [dbo].[sp_Agregar_BD_FISDL_TbEmpleado]

GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[sp_Agregar_BD_FISDL_TbEmpleado]
    @NombreEmpleado nvarchar(50),
    @SexoEmpleado nvarchar(50),
    @FechaNacEmpleado DateTime,
    @EstadCivilEmpleado nvarchar(50),
    @DireccionEmpleado nvarchar(50),
    @IdDepto int,
    @TelefonoEmpleado int,
    @EmailEmpleado nvarchar(50),
    @idUser int,
    @ImagenEmpleado image,
    @UserCrea nvarchar(50),
    @FechaCrea DateTime
AS
INSERT [dbo].[TbEmpleado]
(
    [NombreEmpleado],
    [SexoEmpleado],
    [FechaNacEmpleado],
    [EstadCivilEmpleado],
    [DireccionEmpleado],
    [IdDepto],
    [TelefonoEmpleado],
    [EmailEmpleado],
    [idUser],
    [ImagenEmpleado],
    [UserCrea],
    [FechaCrea]
)
VALUES
(
    @NombreEmpleado,
    @SexoEmpleado,
    @FechaNacEmpleado,
    @EstadCivilEmpleado,
    @DireccionEmpleado,
    @IdDepto,
    @TelefonoEmpleado,
    @EmailEmpleado,
    @idUser,
    @ImagenEmpleado,
    @UserCrea,
    @FechaCrea
)

-- =====
-- Creado por: Marcos E. Ramirez
-- Fecha Creación: Junio de 2013

```

```

-- Modificado por:
-- Fecha Modificación:
-- Descripción: PruebaTecnica en el FISDL
-- =====
-- SP para actualizar un registro a la tabla
-- =====
GO

IF EXISTS (SELECT name FROM sysobjects WHERE name = 'sp_Actualizar_BD_FISDL_TbEmpleado'
AND type = 'P')
    DROP PROCEDURE [dbo].[sp_Actualizar_BD_FISDL_TbEmpleado]

GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[sp_Actualizar_BD_FISDL_TbEmpleado]
    @IdEmpleado int,
    @NombreEmpleado nvarchar(50),
    @SexoEmpleado nvarchar(50),
    @FechaNacEmpleado DateTime,
    @EstadCivilEmpleado nvarchar(50),
    @DireccionEmpleado nvarchar(50),
    @IdDepto int,
    @TelefonoEmpleado int,
    @EmailEmpleado nvarchar(50),
    @idUser int,
    @ImagenEmpleado image,
    @UltUserModi nvarchar(50),
    @UltiFechaModi DateTime

AS
UPDATE [dbo].[TbEmpleado]
SET

[NombreEmpleado] = @NombreEmpleado,
[SexoEmpleado] = @SexoEmpleado,
    [FechaNacEmpleado]= @FechaNacEmpleado,
    [EstadCivilEmpleado]= @EstadCivilEmpleado,
    [DireccionEmpleado]= @DireccionEmpleado,
    [IdDepto]= @IdDepto,
    [TelefonoEmpleado]= @TelefonoEmpleado,
    [EmailEmpleado]= @EmailEmpleado,
    [idUser]= @idUser,
    [ImagenEmpleado]= @ImagenEmpleado,
    [UltUserModi]= @UltUserModi,
    [UltiFechaModi]= @UltiFechaModi

WHERE
    [IdEmpleado] = @IdEmpleado

-- =====
-- Creado por: Marcos E. Ramirez
-- Fecha Creación: Junio de 2013
-- Modificado por:
-- Fecha Modificación:
-- Descripción: PruebaTecnica en el FISDL

```

```

=====
-- SP para borrar un registro a la tabla
=====
GO

IF EXISTS (SELECT name FROM sysobjects WHERE name = 'sp_Eliminar_BD_FISDL_TbEmpleado' AND
type = 'P')
    DROP PROCEDURE [dbo].[sp_Eliminar_BD_FISDL_TbEmpleado]

GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[sp_Eliminar_BD_FISDL_TbEmpleado]
    @IdEmpleado int
AS
DELETE [dbo].[TbEmpleado]
WHERE
    [IdEmpleado] = @IdEmpleado

--- Fin de archivo SQL

```

```

namespace BD_ERP_TabDepartamentos
{
    public class TabDepartamentos : ErpSysClasses.Genericas
    {

        private int _IdDepartamento = -1;
        private string _NomDepartamento = "";
        private int _LongNomDepartamento;

        public int IdDepartamento
        {
            get
            {
                return _IdDepartamento;
            }
        }

        public string NomDepartamento
        {
            get
            {
                return _NomDepartamento;
            }
            set
            {
                _NomDepartamento = value;
            }
        }
    }
}

```

METODOS

- **AgregarDepartamento**

```
public int AddDepartamento(string parmNomDepartamento, string parmUserName)
    {
        int val2ret = -1;
        val2ret = (int)SqlHelper.ExecuteScalar(strCnn,
"sp_Agregar_ERP_TabDepartamentos", parmNomDepartamento, parmUserName);
        _IdDepartamento = val2ret;
        return val2ret;
    }
```

- **ModificarDepartamento**

```
public int ModiDepartamento(int parmIdDepartamento, string parmNomDepartamento, string
parmUserName)
    {
        int val2ret = -1;
        val2ret = (int)SqlHelper.ExecuteNonQuery(strCnn,
"sp_Actualizar_ERP_TabDepartamentos", parmIdDepartamento, parmNomDepartamento,
parmUserName);
        _IdDepartamento = val2ret;
        return val2ret;
    }
```

- **BuscarDepartamento**

```
public DataSet BuscarDepartamento(string parmNomDepartamento)
    {
        DataSet dsTipoClientes = SqlHelper.ExecuteDataset(strCnn,
"sp_Buscar_ERP_TabDepartamentos", parmNomDepartamento);
        return dsTipoClientes;
    }
```

- **EliminarDepartamento**

```

public bool BorrarDepartamento(int parmIdDepartamento)
    {
        try
        {
            int val2ret = -1;
            val2ret = (int)SqlHelper.ExecuteNonQuery(strCnn,
"sp_Eliminar_ERP_TabDepartamentos", parmIdDepartamento);
            if (val2ret > 0)
                return true;
            else
                return false;
        }
        catch
        {
            return false;
        }
    }
}

```

- **LISTARDEPARTAMENTO**

```

public DataSet ListaDepartamentos()
    {
        DataSet dsDepartamento = SqlHelper.ExecuteDataset(strCnn,
CommandType.StoredProcedure, "sp_Obtener_ERP_TabDepartamentos");
        return dsDepartamento;
    }
}

```

+++++

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data;
using System.Data.SqlClient;
using Microsoft.ApplicationBlocks.Data;

```

```

namespace CapaDato
{
    public class Entorno
    {
        protected string strSentencia;
        protected SqlConnection cnnSql = new
SqlConnection(CapaDato.Properties.Settings.Default.cnnFISDL);
        protected string srtCnn = CapaDato.Properties.Settings.Default.cnnFISDL;
    }
}

```

```

namespace SysCapaDato
{
    namespace Tablas
    {
        namespace BD_FISDL_Usuarios
        {
            public class Usuarios : Entorno
            {
                private int _idUser=0;
                private string _UserNombre;
                private string _UserClave;
                private string _UserCrea;
                private DateTime _FechaCrea;
                private string _UltUserModi;
                private DateTime _UltiFechaModi;

                public int IdUser
                {
                    get
                    {
                        return _idUser;
                    }
                }
            }
        }
    }
}

```

.....

```

Imports System.Data
Imports System.Data.SqlClient
' Esta es la referencia o llamada a la biblioteca de clases de acceso a datos
Imports Microsoft.ApplicationBlocks.Data
Imports CapaDato.SysCapaDato.Tablas.BD_FISDL_Depto

```

```

Public Class FrmDepto
    Private clsDatos As New Depto()
    Private intValId As Integer = -1
    Public AddrVrsAct As Integer = 0

    Private Sub FrmDepto_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        'Cargamos el DataGridView en el Load del formulario
        Dim dsorders As DataSet = clsDatos.ListarDepto
        DataGridView1.DataSource = dsorders.Tables("table")
    End Sub

```

```
Private Sub BtnNuevo_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles BtnNuevo.Click
    'Boton Nuevo
    TextBox1.Text = "automatico"
    TextBox2.Text = ""
    TextBox2.Enabled = True
    BtnNuevo.Enabled = False
    BtnGuardar.Enabled = True
    BtnModificar.Enabled = False
    BtnCancelar.Enabled = True
    BtnEliminar.Enabled = False
End Sub
```

```
Private Sub TabControl1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TabControl1.SelectedIndexChanged
    If TabControl1.SelectedTab Is tabPage2 Then
        If intValue < 1 Then
            Else
                End If
        Else
            Dim dsorders As DataSet = clsDatos.ListarDepto
            DataGridView1.DataSource = dsorders.Tables("table")
        End If
    End Sub
```

```
Private Sub DataGridView1_CellEnter(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.DataGridViewCellEventArgs) Handles DataGridView1.CellEnter
    intValue = DataGridView1.Rows(e.RowIndex).Cells(0).Value
    TextBox1.Text = DataGridView1.Rows(e.RowIndex).Cells(0).Value
    TextBox2.Text = DataGridView1.Rows(e.RowIndex).Cells(1).Value
End Sub
```

```
Private Sub DataGridView1_CellDoubleClick(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.DataGridViewCellEventArgs) Handles DataGridView1.CellDoubleClick
    TabControl1.SelectedIndex = 1
End Sub
```

```
Private Sub BtnGuardar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles BtnGuardar.Click
    Dim intResult As Integer
    Dim FechaLocal As String
    FechaLocal = DateTime.Now()
    ErrorProvider1.Clear()
    If TextBox2.Text.Trim.Equals("") Then
        ErrorProvider1.SetError(TextBox2, "No puede quedar en blanco este campo")
        Return
    End If

    If AddvrsAct = 0 Then
        'se ejecuta una insercion de datos
        Try
            intResult = clsDatos.AddDepto(TextBox2.Text,
FrmPrincipal.ToolStripStatusLabel1.Text, FechaLocal.ToString)
```

```

Catch ex As Exception

    MessageBox.Show("Dato almacenado Existosamente")
    'regrando a su estado inicial
    BtnNuevo.Enabled = True
    BtnGuardar.Enabled = False
    BtnModificar.Enabled = True
    BtnCancelar.Enabled = False
    BtnEliminar.Enabled = True
    AdvvrsAct = 0
    TextBox2.Enabled = False
    TabControll1.SelectedIndex = 0
    ErrorProvider1.Clear()
End Try

Else
    'se ejecuta una actualizacion de los datos
    intResult = clsDatos.ActualizarDepto(CInt(TextBox1.Text), TextBox2.Text,
FrmPrincipal.ToolStripStatusLabel1.Text, FechaLocal.ToString)
    MessageBox.Show("Dato Editado Existosamente")
    'regrando a su estado inicial
    TabControll1.SelectedIndex = 0
    TextBox2.Enabled = False
    BtnNuevo.Enabled = True
    BtnGuardar.Enabled = False
    BtnModificar.Enabled = True
    BtnEliminar.Enabled = True
    BtnCancelar.Enabled = False
    AdvvrsAct = 0
End If

End Sub

```

```

Private Sub BtnModificar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnModificar.Click
    TextBox1.Enabled = False
    TextBox2.Enabled = True
    BtnNuevo.Enabled = False
    BtnGuardar.Enabled = True
    BtnModificar.Enabled = False
    BtnCancelar.Enabled = True
    BtnEliminar.Enabled = False
    TextBox2.Enabled = True
    AdvvrsAct = 1
End Sub

```

```

Private Sub BtnCancelar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles BtnCancelar.Click
    BtnNuevo.Enabled = True
    BtnGuardar.Enabled = False
    BtnModificar.Enabled = True
    BtnCancelar.Enabled = False
    BtnEliminar.Enabled = True
    AdvvrsAct = 0
    TextBox2.Enabled = False
    TabControll1.SelectedIndex = 0
    ErrorProvider1.Clear()
End Sub

```

```
Private Sub BtnEliminar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnEliminar.Click
    Dim blnBorrado As Boolean
    blnBorrado = clsDatos.EliminarDepto(CInt(TextBox1.Text))
    MessageBox.Show("Campo Eliminado")
    'regresando a su estado inicial
    BtnNuevo.Enabled = True
    BtnGuardar.Enabled = False
    BtnModificar.Enabled = True
    BtnCancelar.Enabled = False
    BtnEliminar.Enabled = True
    AddvrsAct = 0
    TextBox2.Enabled = False
    TabControl1.SelectedIndex = 0
    ErrorProvider1.Clear()
End Sub
```

```
Private Sub BtnBuscarUno_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnBuscarUno.Click
    Dim strResult As String
    strResult = InputBox("¿Nombre del Departamento a buscar?", "Búsqueda de
información")
    Dim dsorders As DataSet = clsDatos.BuscarDepto(strResult)
    DataGridView1.DataSource = dsorders.Tables("table")
End Sub
```

```
Private Sub BtnListarTodos_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnListarTodos.Click
    Dim dsorders As DataSet = clsDatos.ListarDepto
    DataGridView1.DataSource = dsorders.Tables("table")
End Sub
```

```
Private Sub ValidarCampos()
    ErrorProvider1.Clear()
    If TextBox2.Text.Trim.Equals("") Then
        ErrorProvider1.SetError(TextBox2, "No puede quedar en blanco este campo")
        Return
    End If
    'aquí ponemos más if con sus respectivos textbox
End Sub
```

FORMULARIO EMPLEADO

```
Imports System.Data
Imports System.Data.SqlClient
' Esta es la referencia o llamada a la biblioteca de clases de acceso a datos
Imports Microsoft.ApplicationBlocks.Data
'Imports CapaDato.SysCapaDato.Tablas.BD_FISDL_Empleados
Imports SysCapaDato.Tablas.BD_FISDL_Empleados
Imports System.IO
```

```
Public Class frmEmpleados
    Private clsDatos As New Empleados()
    Private intValId As Integer = -1
    Public AdvvrsAct As Integer = 0
    Private Sub frmEmpleados_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Cargamos el DataGrid en el Load del formulario
        Dim dsorders As DataSet = clsDatos.ListarEmpleado
        DataGridView1.DataSource = dsorders.Tables("table")
    End Sub
```

```
    Private Sub TextBox2_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TxtNombreEmpleado.TextChanged

    End Sub
```

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnSeleccion.Click
    Try
        Dim oFD As New OpenFileDialog
        oFD.Title = "Seleccionar imagen"
        oFD.Filter = "Imágenes|*.jpg;*.gif;*.png;*.bmp|Todos (*.*)|*.*"
        oFD.FileName = Me.txtImagen.Text
        If oFD.ShowDialog = Windows.Forms.DialogResult.OK Then
            Me.txtImagen.Text = oFD.FileName
            Me.PictureBox1.Image = Image.FromFile(Me.txtImagen.Text)
        End If
    Catch ex As Exception
        MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
    End Try
End Sub
```

```
    Private Sub DataGridView1_CellDoubleClick(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.DataGridViewCellEventArgs) Handles DataGridView1.CellDoubleClick
        TabControll1.SelectedIndex = 1
    End Sub
```

```
Private Sub DataGridView1_CellEnter(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.DataGridViewCellEventArgs) Handles DataGridView1.CellEnter
    intValId = DataGridView1.Rows(e.RowIndex).Cells(0).Value
    txtIdEmpleado.Text = DataGridView1.Rows(e.RowIndex).Cells(0).Value
    TxtNombreEmpleado.Text = DataGridView1.Rows(e.RowIndex).Cells(1).Value
    txtSexoEmpleado.Text = DataGridView1.Rows(e.RowIndex).Cells(2).Value
    txtFechaNacEmpleado.Text = DataGridView1.Rows(e.RowIndex).Cells(3).Value
    txtEstaCivilEmpleado.Text = DataGridView1.Rows(e.RowIndex).Cells(4).Value
    txtDireccionEmpleado.Text = DataGridView1.Rows(e.RowIndex).Cells(5).Value
    cmbDepto.Text = DataGridView1.Rows(e.RowIndex).Cells(6).Value
    txtTelefonoEmpleado.Text = DataGridView1.Rows(e.RowIndex).Cells(7).Value
    txtEmailEmpleado.Text = DataGridView1.Rows(e.RowIndex).Cells(8).Value.ToString
    cmbUser.Text = DataGridView1.Rows(e.RowIndex).Cells(9).Value.ToString

Try
    'Pasar de un DATAGRIDVIEW A UN PICTUREBOX
    Dim bits As Byte() = CType(DataGridView1.Rows(e.RowIndex).Cells(11).Value,
Byte())
    Dim memorybits As New MemoryStream(bits)
    Dim bitmap As New Bitmap(memorybits)
    PictureBox1.Image = bitmap

Catch ex As Exception

End Try

End Sub
```

```
Private Sub TabControl1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles TabControl1.SelectedIndexChanged
    If TabControl1.SelectedTab Is tabPage2 Then
        If intValId < 1 Then
            Else
            End If
        Else
            Dim dsorders As DataSet = clsDatos.ListarEmpleado
            DataGridView1.DataSource = dsorders.Tables("table")
        End If
    End Sub
```

```
Private Sub btnNuevo_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnNuevo.Click
    txtDireccionEmpleado.Enabled = True
    txtEmailEmpleado.Enabled = True
    txtEstaCivilEmpleado.Enabled = True
    txtFechaNacEmpleado.Enabled = True
    TxtNombreEmpleado.Enabled = True
    txtImagen.Enabled = True
    txtSexoEmpleado.Enabled = True
    txtTelefonoEmpleado.Enabled = True
    cmbDepto.Enabled = True
```

```
cmbUser.Enabled = True
'Botones
btnSeleccion.Enabled = True
btnCancelar.Enabled = True
btnNuevo.Enabled = False
btnGuardar.Enabled = True

txtIdEmpleado.Text = "Auto"
txtDireccionEmpleado.Text = ""
txtEmailEmpleado.Text = ""
txtEstaCivilEmpleado.Text = ""
txtFechaNacEmpleado.Text = ""
TxtNombreEmpleado.Text = ""
txtImagen.Text = ""
txtSexoEmpleado.Text = ""
txtTelefonoEmpleado.Text = ""
cmbDepto.Text = ""
cmbUser.Text = ""
```

End Sub

```
Private Sub btnCancelar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCancelar.Click
    txtDireccionEmpleado.Enabled = False
    txtEmailEmpleado.Enabled = False
    txtEstaCivilEmpleado.Enabled = False
    txtFechaNacEmpleado.Enabled = False
    TxtNombreEmpleado.Enabled = False
    txtImagen.Enabled = False
    txtSexoEmpleado.Enabled = False
    txtTelefonoEmpleado.Enabled = False
    cmbDepto.Enabled = False
    cmbUser.Enabled = False

    'Botones
    btnSeleccion.Enabled = False
    btnCancelar.Enabled = False
    btnNuevo.Enabled = True
    btnGuardar.Enabled = False

    TabControl1.SelectedIndex = 0
End Sub
```

```
Private Sub btnGuardar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnGuardar.Click

    '
    '
    '
    'txtImagen
    '
    '
    'cmbUser

    Dim dataImg As Byte()
```

```

Dim intResult As Integer
Dim FechaLocal As String
FechaLocal = DateTime.Now()
ErrorProvider1.Clear()

dataImg = ImageHelper.ImageToByteArray(PictureBox1.Image)
If TxtNombreEmpleado.Text.Trim.Equals("") Then
    ErrorProvider1.SetError(TxtNombreEmpleado, "No puede quedar en blanco este
campo")
    Return
End If
If txtSexoEmpleado.Text.Trim.Equals("") Then
    ErrorProvider1.SetError(txtSexoEmpleado, "No puede quedar en blanco este
campo")
    Return
End If

If txtDireccionEmpleado.Text.Trim.Equals("") Then
    ErrorProvider1.SetError(txtDireccionEmpleado, "No puede quedar en blanco este
campo")
    Return
End If
If txtEmailEmpleado.Text.Trim.Equals("") Then
    ErrorProvider1.SetError(txtEmailEmpleado, "No puede quedar en blanco este
campo")
    Return
End If
If txtEstaCivilEmpleado.Text.Trim.Equals("") Then
    ErrorProvider1.SetError(txtEstaCivilEmpleado, "No puede quedar en blanco este
campo")
    Return
End If
If txtFechaNacEmpleado.Text.Trim.Equals("") Then
    ErrorProvider1.SetError(txtFechaNacEmpleado, "No puede quedar en blanco este
campo")
    Return
End If
If txtFechaNacEmpleado.Text.Trim.Equals("") Then
    ErrorProvider1.SetError(txtFechaNacEmpleado, "No puede quedar en blanco este
campo")
    Return
End If
If txtTelefonoEmpleado.Text.Trim.Equals("") Then
    ErrorProvider1.SetError(txtTelefonoEmpleado, "No puede quedar en blanco este
campo")
    Return
End If
If cmbDepto.Text.Trim.Equals("") Then
    ErrorProvider1.SetError(cmbDepto, "No puede quedar en blanco este campo")
    Return
End If
If cmbDepto.Text.Trim.Equals("") Then
    ErrorProvider1.SetError(cmbDepto, "No puede quedar en blanco este campo")
    Return
End If
If cmbUser.Text.Trim.Equals("") Then
    ErrorProvider1.SetError(cmbUser, "No puede quedar en blanco este campo")
    Return
End If

If AddvrsAct = 0 Then
    'se ejecuta una insercion de datos

```

```

        Try
            intResult = clsDatos.AddEmpleado(TxtNombreEmpleado.Text,
            txtSexoEmpleado.Text, txtFechaNacEmpleado.Text, txtEstaCivilEmpleado.Text,
            txtDireccionEmpleado.Text, cmbDepto.Text, txtTelefonoEmpleado.Text,
            txtEmailEmpleado.Text, cmbUser.Text, ImageHelper.ImageToByteArray(PictureBox1.Image),
            FrmPrincipal.ToolStripStatusLabel1.Text, FechaLocal.ToString)
            'intResult = clsDatos.AddEmpleado(TxtNombreEmpleado.Text,
            txtSexoEmpleado.Text, txtFechaNacEmpleado.Text, txtEstaCivilEmpleado.Text,
            txtDireccionEmpleado.Text, cmbDepto.Text, txtTelefonoEmpleado.Text,
            txtEmailEmpleado.Text, cmbUser.Text, dataImg, FrmPrincipal.ToolStripStatusLabel1.Text,
            FechaLocal.ToString)
        Catch ex As Exception

            MessageBox.Show("Dato almacenado Existosamente")
            'regrando a su estado inicial

        End Try

    Else
        'se ejecuta una actualizacion de los datos
        'intResult = clsDatos.ActualizarDepto(CInt(TextBox1.Text), TextBox2.Text,
        FrmPrincipal.ToolStripStatusLabel1.Text, FechaLocal.ToString)
        MessageBox.Show("Dato Editado Existosamente")
        'regrando a su estado inicial

    End If

End Sub

```

```

Private Sub txtTelefonoEmpleado_KeyPress(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txtTelefonoEmpleado.KeyPress
    'Solo Numeros
    If Char.IsDigit(e.KeyChar) Then
        e.Handled = False
    ElseIf Char.IsControl(e.KeyChar) Then
        e.Handled = False
    Else
        e.Handled = True
    End If
End Sub
End Class

```

+++++

CODIGO C#

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data;
using System.Data.SqlClient;
using Microsoft.ApplicationBlocks.Data;

namespace CapaDato
{
    public class Entorno
    {
        protected string strSentencia;
        protected SqlConnection cnnSql = new
SqlConnection(CapaDato.Properties.Settings.Default.cnnFISDL);
        protected string srtCnn = CapaDato.Properties.Settings.Default.cnnFISDL;
    }
}

namespace SysCapaDato
{
    namespace Tablas
    {
        namespace BD_FISDL_Usuarios
        {
            public class Usuarios : CapaDato.Entorno
            {
                private int _idUser=0;
                private string _UserNombre;
                private string _UserClave;
                private string _UserCrea;
                private DateTime _FechaCrea;
                private string _UltUserModi;
                private DateTime _UltiFechaModi;

                public int IdUser
                {
                    get
                    {
                        return _idUser;
                    }
                }

                public string UserNombre
                {
                    get
                    {
                        return _UserNombre;
                    }
                    set
                    {
                        _UserNombre = value;
                    }
                }

                public string UserClave
                {
                    get

```

```

        {
            return _UserClave;
        }
        set
        {
            _UserClave = value;
        }
    }

    public string UserCrea
    {
        get
        {
            return _UserCrea;
        }
        set
        {
            _UserCrea = value;
        }
    }

    public DateTime FechaCrea
    {
        get
        {
            return _FechaCrea;
        }
        set
        {
            _FechaCrea = value;
        }
    }

    public string UltUserModi
    {
        get
        {
            return _UltUserModi;
        }
        set
        {
            _UltUserModi = value;
        }
    }

    public DateTime UltiFechaModi
    {
        get
        {
            return _UltiFechaModi;
        }
        set
        {
            _UltiFechaModi = value;
        }
    }

    public int AddUsuario()
    {
        throw new System.NotImplementedException();
    }

    public DataSet BuscarUsuario(string parmUsuario)

```

```

    {
        DataSet dsUsuario = SqlHelper.ExecuteDataset(srtCnn,
"sp_ObtenerUno_BD_FISDL_TbUsuario", parmUsuario);
        return dsUsuario;
    }

    public bool BorrarUsuario()
    {
        throw new System.NotImplementedException();
    }

    public DataSet ListarUsuario()
    {
        throw new System.NotImplementedException();
    }

    public int ModificarUsuario()
    {
        throw new System.NotImplementedException();
    }
}
}
}
}

```

```

namespace SysCapaDato
{
    namespace Tablas
    {
        namespace BD_FISDL_Empleados
        {
            public class Empleados : CapaDato.Entorno
            {
                private int _IdEmpleado;
                private string _NombreEmpleado;
                private string _SexoEmpleado;
                private DateTime _FechaNacEmpleado;
                private string _EstadCivilEmpleado;
                private string _DireccionEmpleado;
                private int _IdDepto;
                private int _TelefonoEmpleado;
                private string _EmailEmpleado;
                private int _idUser;
                private byte _ImagenEmpleado;
                private string _UserCrea;
                private DateTime _FechaCrea;
                private string _UltUserModi;
                private DateTime _UltiFechaModi;

                public int IdEmpleado
                {
                    get
                    {
                        return _IdEmpleado;
                    }
                }

                public string SexoEmpleado
                {
                    get
                    {

```

```

        return _SexoEmpleado;
    }
    set
    {
        _SexoEmpleado = value;
    }
}

public string NombreEmpleado
{
    get
    {
        return _NombreEmpleado;
    }
    set
    {
        _NombreEmpleado = value;
    }
}

public DateTime FechaNacEmpleado
{
    get
    {
        return _FechaNacEmpleado;
    }
    set
    {
        _FechaNacEmpleado = value;
    }
}

public string EstadCivilEmpleado
{
    get
    {
        return _EstadCivilEmpleado;
    }
    set
    {
        _EstadCivilEmpleado = value;
    }
}

public string DireccionEmpleado
{
    get
    {
        return _DireccionEmpleado;
    }
    set
    {
        _DireccionEmpleado = value;
    }
}

public int IdDepto
{
    get
    {
        return _IdDepto;
    }
    set

```

```
    {
        _IdDepto = value;
    }
}

public int TelefonoEmpleado
{
    get
    {
        return _TelefonoEmpleado;
    }
    set
    {
        _TelefonoEmpleado = value;
    }
}

public string EmailEmpleado
{
    get
    {
        return _EmailEmpleado;
    }
    set
    {
        _EmailEmpleado = value;
    }
}

public int IdUser
{
    get
    {
        return _IdUser;
    }
    set
    {
        _IdUser = value;
    }
}

public byte ImagenEmpleado
{
    get
    {
        return _ImagenEmpleado;
    }
    set
    {
        _ImagenEmpleado = value;
    }
}

public string UserCrea
{
    get
    {
        return _UserCrea;
    }
    set
    {
        _UserCrea = value;
    }
}
```

```

public DateTime FechaCrea
{
    get
    {
        return _FechaCrea;
    }
    set
    {
        _FechaCrea = value;
    }
}

public string UltUserModi
{
    get
    {
        return _UltUserModi;
    }
    set
    {
        _UltUserModi = value;
    }
}

public DateTime UltiFechaModi
{
    get
    {
        return _UltiFechaModi;
    }
    set
    {
        _UltiFechaModi = value;
    }
}

public int AddEmpleado(string parmNombreEmpleado, string parmSexoEmpleado, DateTime
parmFechaNacEmpleado, string parmEstadCivilEmpleado, string parmDireccionEmpleado, int
parmIdDepto, int parmTelefonoEmpleado, string parmEmailEmpleado, int parmIdUser, Array
parmImagenEmpleado, string parmUserCrea, DateTime parmFechaCrea)
{
    int val2ret = -1;
    val2ret = (int)SqlHelper.ExecuteScalar(srtcnn, "sp_Agregar_BD_FISDL_TbEmpleado",
parmNombreEmpleado, parmSexoEmpleado, parmFechaNacEmpleado, parmEstadCivilEmpleado,
parmDireccionEmpleado, parmIdDepto, parmTelefonoEmpleado, parmEmailEmpleado,
parmIdUser, parmImagenEmpleado, parmUserCrea, parmFechaCrea);
    _IdEmpleado = val2ret;
    return val2ret;
}

public DataSet BuscarEmpleado(string parmNombreEmpleado)
{
    DataSet dsTipoEmpleado = SqlHelper.ExecuteDataset(srtcnn,
"sp_Buscar_BD_FISDL_TbEmpleado", parmNombreEmpleado);
    return dsTipoEmpleado;
}

public bool EliminarEmpleado(int parmIdEmpleado)
{
    try
    {

```

```

        int val2ret = -1;
        val2ret = (int)SqlHelper.ExecuteNonQuery(srtCnn,
"sp_Eliminar_BD_FISDL_TbEmpleado", parmIdEmpleado);
        if (val2ret > 0)
            return true;
        else
            return false;
    }
    catch
    {
        return false;
    }
}

public DataSet ListarEmpleado()
{
    DataSet dsEmpleados = SqlHelper.ExecuteDataset(srtCnn,
CommandType.StoredProcedure, "sp_Obtener_BD_FISDL_TbEmpleado");
    return dsEmpleados;
}

public int ModificarEmpleado(int parmIdEmpleado, string parmNombreEmpleado, string
parmSexoEmpleado, DateTime parmFechaNacEmpleado, string parmEstadCivilEmpleado, string
parmDireccionEmpleado, int parmIdDepto, int parmTelefonoEmpleado, string
parmEmailEmpleado, int parmIdUser, Byte parmImagenEmpleado, string parmUserModi, DateTime
parmFechaModi)
{
    int val2ret = -1;
    val2ret = (int)SqlHelper.ExecuteNonQuery(srtCnn,
"sp_Actualizar_BD_FISDL_TbEmpleado", parmIdEmpleado, parmNombreEmpleado,
parmSexoEmpleado, parmFechaNacEmpleado, parmEstadCivilEmpleado, parmDireccionEmpleado,
parmIdDepto, parmTelefonoEmpleado, parmEmailEmpleado, parmIdUser, parmImagenEmpleado,
parmUserModi, parmFechaModi);
    _IdEmpleado = val2ret;
    return val2ret;
}
}
}
}
}

namespace SysCapaDato
{
    namespace Tablas
    {
        namespace BD_FISDL_Depto
        {
            public class Depto : CapaDato.Entorno
            {
                private int _IdDepto=-1;
                private string _NombreDepto="";
                private string _UserCrea="";
                private DateTime _FechaCrea;
                private string _UltUserModi="";
                private DateTime _UltFechaModi;

                public DateTime FechaCrea
                {
                    get
                    {
                        return _FechaCrea;
                    }
                }
            }
        }
    }
}

```

```
    }
    set
    {
        _FechaCrea = value;
    }
}

public int IdDepto
{
    get
    {
        return _IdDepto;
    }
}

public string NombreDepto
{
    get
    {
        return _NombreDepto;
    }
    set
    {
        _NombreDepto = value;
    }
}

public DateTime UltFechaModi
{
    get
    {
        return _UltFechaModi;
    }
    set
    {
        _UltFechaModi = value;
    }
}

public string UltUserModi
{
    get
    {
        return _UltUserModi;
    }
    set
    {
        _UltUserModi = value;
    }
}

public string UserCrea
{
    get
    {
        return _UserCrea;
    }
    set
    {
        _UserCrea = value;
    }
}
```

```

    public int AddDepto(string parmNomDepartamento, string parmUserName, string
parmFechaCrea)
    {
        int val2ret = -1;
        val2ret = (int)SqlHelper.ExecuteScalar(srtcnn, "sp_Agregar_BD_FISDL_TbDepto",
parmNomDepartamento, parmUserName, parmFechaCrea);
        _IdDepto = val2ret;
        return val2ret;
    }

    public DataSet ListarDepto()
    {
        DataSet dsDepartamento = SqlHelper.ExecuteDataset(srtcnn,
CommandType.StoredProcedure, "sp_Obtener_BD_FISDL_TbDepto");
        return dsDepartamento;
    }

    public DataSet BuscarDepto(string parmNomDepartamento)
    {
        DataSet dsTipoDepto = SqlHelper.ExecuteDataset(srtcnn,
"sp_Buscar_BD_FISDL_TbDepto", parmNomDepartamento);
        return dsTipoDepto;
    }

    public bool EliminarDepto(int parmIdDepartamento)
    {
        try
        {
            int val2ret = -1;
            val2ret = (int)SqlHelper.ExecuteNonQuery(srtcnn,
"sp_Eliminar_BD_FISDL_TbDepto", parmIdDepartamento);
            if (val2ret > 0)
                return true;
            else
                return false;
        }
        catch
        {
            return false;
        }
    }

    public int ActualizarDepto(int parmIdDepartamento, string parmNomDepartamento, string
parmUserModi, string parmFechaModi)
    {
        int val2ret = -1;
        val2ret = (int)SqlHelper.ExecuteNonQuery(srtcnn,
"sp_Actualizar_BD_FISDL_TbDepto", parmIdDepartamento, parmNomDepartamento, parmUserModi,
parmFechaModi);
        _IdDepto = val2ret;
        return val2ret;
    }
}
}
}
}

```

ImageHelper.VB

```
Imports System.Collections.Generic
Imports System.Linq
Imports System.Text
Imports System.IO
Imports System.Drawing
Imports System.Drawing.Imaging
Imports System.Reflection

Public Class ImageHelper

    Public Shared Function ByteArrayToImage(ByVal byteArrayIn As Byte()) As Image

        Dim ms As New MemoryStream(byteArrayIn)
        Return Image.FromStream(ms)

    End Function

    Public Shared Function ImageToByteArray(ByVal imageIn As Image) As Byte()

        Dim ms As New MemoryStream()
        imageIn.Save(ms, ImageFormat.Jpeg)
        Return ms.ToArray()

    End Function

    Public Shared Function ObtenerImagenNoDisponible() As Image

        Dim assembly As Assembly = System.Reflection.Assembly.GetExecutingAssembly()

        Dim file As Stream =
assembly.GetManifestResourceStream("Helpers.NoDisponible.jpg")
        Return Image.FromStream(file)

    End Function

End Class
```