Clase conexión:

```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Data;
using System.Linq;
using System.Web;

namespace Ejemplo_Api2.Data
{

    public class ConexionBD
    {
        #region "Atributos"
        private string strError;
        private bool blnBDAbierta;
        private string strCadenaCnx;
        private SqlConnection objCnnBD;
        private SqlCommand objCmdBD;
        private SqlDataReader objReader;
        private SqlDataAdapter dapGenerico; //Intermediario para llenar el dataset
        private DataSet dts;
        private string strVrUnico;
        private SqlParameter objParametro;
        #endregion
        #region "Constructor"
        public ConexionBD()
        {
            objCnnBD = new SqlConnection();
            objCmdBD = new SqlCommand();
            dapGenerico = new SqlDataAdapter();
            strVrUnico = "";
            objParametro = new SqlParameter();
            strError = "";
        }
        #endregion
        #region "Propiedades"
        public SqlDataReader Reader
        {
            get { return objReader; }
        }
        public DataSet DataSet_Retornado
        {
            get { return dts; }
        }
        public string Error
        {
            set { strError = value; }
            get { return strError; }
        }
        public string ValorUnico
        {
            get { return strVrUnico; }
        }
        #endregion
        #region "Metodos Privados"

        private bool AbrirConexion()
        {

            try
            {
                strCadenaCnx = "Data Source=DESKTOP-UGSFCI7\\SQLEXPRESS;Initial
Catalog=DB_API_DATOS;Integrated Security=True";
                objCnnBD.ConnectionString = strCadenaCnx;
                objCnnBD.Open();
                blnBDAbierta = true;
                return true;
```

```
        }
        catch (Exception ex)
        {
            blnBDAbierta = false;
            strError = "Error al abrir la conexion -" + ex.Message;
            return false;
        }

    }
    #endregion
    #region "Metodos Publicos"
    public bool Consultar(string SentenciaSQL, bool blnCon_Parametros)
    {
        if (SentenciaSQL == "")
        {
            strError = "Error en instrucción SQL";
            return false;
        }
        if (blnBDAbierta == false)
        {
            if (AbrirConexion() == false)
            {
                return false;
            }
        }
        objCmdBD.Connection = objCnnBD;
        if (blnCon_Parametros)
            objCmdBD.CommandType = CommandType.StoredProcedure;
        else
            objCmdBD.CommandType = CommandType.Text;
        objCmdBD.CommandText = SentenciaSQL;
        try
        {
            objReader = objCmdBD.ExecuteReader();
            return true;
        }
        catch (Exception ex)
        {
            strError = "Falla en ejecutar comando -" + ex.Message;
            return false;
        }
    }
    public bool EjecutarSentencia(string SentenciaSQL, bool blnCon_Parametros)
    {
        if (SentenciaSQL == "")
        {
            strError = "No se ha definido la sentencia a ejecutar ";
            return false;
        }
        if (blnBDAbierta == false)
        {
            if (AbrirConexion() == false)
            {
                return false;
            }
        }
        objCmdBD.Connection = objCnnBD;
        if (blnCon_Parametros)
            objCmdBD.CommandType = CommandType.StoredProcedure;
        else
            objCmdBD.CommandType = CommandType.Text;
        objCmdBD.CommandText = SentenciaSQL;
        try
        {
            objCmdBD.ExecuteNonQuery();
            return true;
        }
        catch (Exception ex)
        {
```

```
                strError = "Error al ejecutar la instrucción -" + ex.Message;
                return false;
            }
        }
        public bool ConsultarValorUnico(string SentenciaSQL, bool blnCon_Parametros)
        {
            if (SentenciaSQL == "")
            {
                strError = "No se ha definido la sentencia a ejecutar ";
                return false;
            }
            if (blnBDAbierta == false)
            {
                if (AbrirConexion() == false)
                {
                    return false;
                }
            }
            objCmdBD.Connection = objCnnBD;
            if (blnCon_Parametros)
                objCmdBD.CommandType = CommandType.StoredProcedure;
            else
                objCmdBD.CommandType = CommandType.Text;
            objCmdBD.CommandText = SentenciaSQL;
            try
            {
                strVrUnico = Convert.ToString(objCmdBD.ExecuteScalar());
                return true;
            }
            catch (Exception ex)
            {
                strError = "Error al ejecutar instrucción -" + ex.Message;
                return false;
            }
        }
        public void CerrarConexion()
        {
            try
            {
                objCmdBD = null; //Destruye objeto Command
            }
            catch (Exception ex)
            {
                strError = "Falla en cerrar Command -" + ex.Message;
            }
            try
            {
                objCnnBD.Close(); //Cierra la conexión
                objCnnBD = null; //Destruye en la emmoria
            }
            catch (Exception ex)
            {
                strError = "Falla en cerrar conexión -" + ex.Message;
            }
        }
        public bool LlenarDataSet(string NombreTabla, string SentenciaSQL, bool
blnCon_Parametros)
        {
            // con este booleano se sabe si hay conexion abierta o no
            if (blnBDAbierta == false)
            {
                if (AbrirConexion() == false)
                {
                    return false;
                }
            }
            objCmdBD.Connection = objCnnBD;
            if (blnCon_Parametros)
                objCmdBD.CommandType = CommandType.StoredProcedure;
```

```
            else
                objCmdBD.CommandType = CommandType.Text;
            objCmdBD.CommandText = SentenciaSQL;
            try
            {
                dts = new DataSet();
                dapGenerico.SelectCommand = objCmdBD;
                dapGenerico.Fill(dts, NombreTabla);
                return true;
            }
            catch (Exception ex)
            {
                strError = ex.Message;
                return false;
            }
        }
        public bool AgregarParametro(ParameterDirection Direccion, string Nombre_En_SP,
        SqlDbType TipoDato, Int16 Tamaño, object Valor)
        {
            try
            {
                objParametro.Direction = Direccion;
                objParametro.ParameterName = Nombre_En_SP;
                objParametro.SqlDbType = TipoDato;
                objParametro.Size = Tamaño;
                objParametro.Value = Valor;
                objCmdBD.Parameters.Add(objParametro);
                objParametro = new SqlParameter();
                return (true);
            }
            catch (Exception ex)
            {
                strError = ex.Message;
                return (false);
            }
        }
        #endregion
    }
}
```