Cargo esto desde SQL Server Management:

use CRUD\_CODE\_RJ

create table Productos

(

Id int Identity (1,1) primary key,

Nombre nvarchar (100),

Descripcion nvarchar (100),

Marca nvarchar (100),

Precio float,

Stock int

)

insert into Productos

values

('Gaseosa', '3 litros', 'marcacola', 300.5,10),

('Chocolate', 'tableta 100 gramos', 'Ibérica', 500,15),

('Arroz', '500 gramos', 'chinito-feliz', 120, 50),

('Harina', '1 kg. Bolsa', 'molinetes', 70,100),

('Azucar', '1 kg. Bolsa', 'Sarampión', 90,80)

Crea una “Solución en blanco” en VS.

Y agrega 3 proyectos: capa de datos, de negocio y de presentación. Agrega las referencias entre ellas.

Crea clase para conexion a BBDD

using System.Data.SqlClient;

using System.Data;

private SqlConnection Conexion = new SqlConnection("Server= DESKTOP-MSJO6R8; database=Productos; Integrated Security:true");

Si el servidor es local se puede colocar (local) y con eso alcanza.

2 formas de sqlConnection:

1\_

private SqlConnection Conexion = new SqlConnection("Server=(local); DataBase=CrudCodeRJ; Integrated Security=true");

2\_ O crear un string y luego utilizarlo:

string connectionString = "Data Source=DESKTOP-MSJO6R8;Initial Catalog=EMPLEADOS\_DB; Integrated Security=true";

using (SqlConnection connection = new SqlConnection(connectionString))

{

}

***Todo para el select:***

Agrega otra clase para hacer un select\* sobre la BBDD.

Luego crea capa de negocio y ahí:

using System.Data;

using System.Data.SqlClient;

using CapaDeDatos;

**Momento 1**

**Capa de datos**

public class CD\_ConexionBBDD

{

private SqlConnection Conexion = new SqlConnection("Server=(local); DataBase=CrudCodeRJ; Integrated Security=true");

public SqlConnection AbrirConexion()

{

if (Conexion.State == ConnectionState.Closed)

{

Conexion.Open();

}

return Conexion;

}

public SqlConnection CerrarConexion()

{

if (Conexion.State == ConnectionState.Open)

{

Conexion.Close();

}

return Conexion;

}

}

public class CD\_Productos

{

private CD\_ConexionBBDD conexion = new CD\_ConexionBBDD();

SqlDataReader leer;

DataTable tabla = new DataTable();

SqlCommand comando = new SqlCommand();

public DataTable Mostrar()

{

//Transact SQL

comando.Connection = conexion.AbrirConexion();

comando.CommandText = "Select \* from Productos";

leer = comando.ExecuteReader();

tabla.Load(leer);

conexion.CerrarConexion();

return tabla;

}

}

**Capa de negocio**

public class CN\_Productos

{

private CD\_Productos objetoCD = new CD\_Productos();

public DataTable MostrarProducto()

{

DataTable tabla = new DataTable();

tabla = objetoCD.Mostrar();

return tabla;

}

}

**Capa de presentación**

public partial class Form1 : Form

{

CN\_Productos objetoCN = new CN\_Productos();

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

MostrarProductos();

}

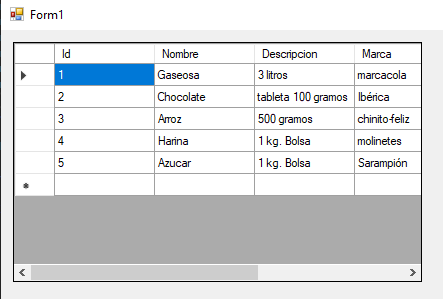
private void MostrarProductos()

{

dataGridView1.DataSource = objetoCN.MostrarProducto();

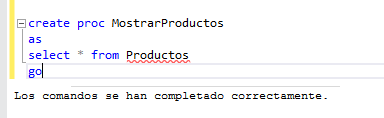
}

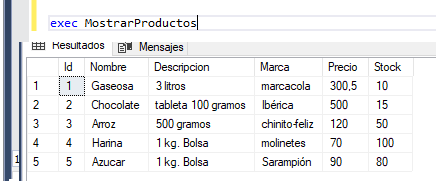
}



A continuación, quiere utilizar un procedimiento almacenado (alternativo a transact sql) para el método mostrar.

Procedimiento almacenado





Así se ejecuta.

public DataTable Mostrar()

{

//Con procedimiento almacenado

comando.Connection = conexion.AbrirConexion();

comando.CommandText = "MostrarProductos";

comando.CommandType = CommandType.StoredProcedure;

leer = comando.ExecuteReader();

tabla.Load(leer);

conexion.CerrarConexion();

return tabla;

}

Se pone esto en la lógica de datos, una vez creado al storedProcedure en el management.

***Todo para el Create:***

Se obtienen 5 strings. Luego hay uno que debe convertirse a double y otro a float. Afirma que eso debe hacerse en la capa de negocio, no en la de presentación.

Atenti (Si usas stored procedure)

Cuando pones comando.GetType=StoredProcedure. Eso queda grabado, después, cuando quieras usar el comando para algo que no sea storedProcedure va a tirar error. Hay que volver a definir el comando.GetType.

Atenti2 (Para que no printee lo mismo 2 veces)

private void MostrarProductos()

{

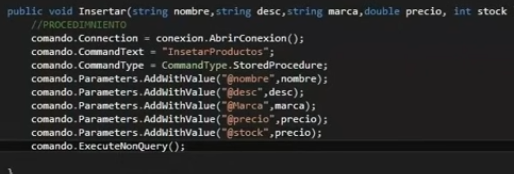
objetoCN = new CN\_Productos();

dataGridView1.DataSource = objetoCN.MostrarProducto();

}

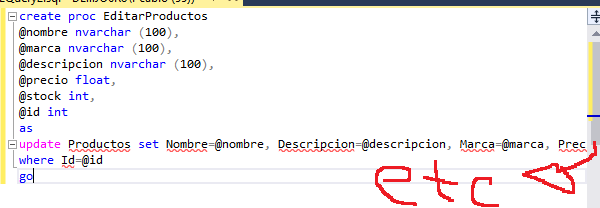
Elige crear una nueva instancia del objeto. No hace ningún clear.



Ese es el procedimiento almacenado para insertar los registros.

Ahí con C#. Nótese que no pone exec.

***Todo para el Update:***



***Todo para el Delete:***

create proc EliminaRegistro

@id int

as

delete from Productos where Id=@id

go

Se ejecuta haciendo esto:

exec EliminaRegistro 8

Si se va a usar varias veces una misma instancia de sqlcommand. Tras usarlo hay que limpiarlo, porque si no al reusarlo para otro storedProcedure podría haber problemas.

ExecuteNonQuery solo es para: insert, delete, créate y update.

ExecuteReader devuelve filas.