

1. Clase para la conexión

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data;
using MySql.Data.MySqlClient;

namespace WindowsFormsApp3
{
    class BdComun
    {
        public static MySqlConnection ObtenerConexion()
        {
            MySqlConnection conectar = new
            MySqlConnection("datasource=localhost;port=3306;username=root;password=qwerty;databa
            se=pruebasena;");
            conectar.Open();
            return conectar;
        }
    }
}
```

Se crea una clase que tenga los métodos de consulta

```
class ConjuntoDatos

public static DataTable Consultas(string query)
{
    // retornar datatable para llenar grilla o control que necesite un DataSet o
    DataTable
        DataTable dtRetorno = new DataTable();
        using (MySqlCommand sqlCmd = new MySqlCommand(query,
        BdComun.ObtenerConexion()))
        using (MySqlDataAdapter sqlDA = new MySqlDataAdapter(sqlCmd))
        {
            sqlDA.Fill(dtRetorno);
        }
        return dtRetorno;
    }
}
```

//Llamar Metodos consultas para cargar Grilla

```
DtgUsuario.DataSource = ConjuntoDatos.Consultas(("p_usuario"));
```

Se utiliza un procedimiento almacenado creado en MySql

.....

Ejemplo de metodo que permite Guardar o Actualizar varios registros de una tabla

```
public static int Consultas4(string query, int i, string n, string a, string d,
string t)
{

    int retorno;
    DataTable dtRetorno = new DataTable();
```

```

        MySqlCommand sqlCmd = new MySqlCommand();
        using (sqlCmd)
        {
            sqlCmd.Connection = BdComun.ObtenerConexion();
            sqlCmd.CommandType = CommandType.StoredProcedure;
            sqlCmd.CommandText = query;
            sqlCmd.Parameters.AddWithValue("id_u", i);
            sqlCmd.Parameters.AddWithValue("no_u", n);
            sqlCmd.Parameters.AddWithValue("ap_u", a);
            sqlCmd.Parameters.AddWithValue("di_u", d);
            sqlCmd.Parameters.AddWithValue("te_u", t);
            retorno = sqlCmd.ExecuteNonQuery();
        }

        return retorno;
    }
}

```

Este método retorna un entero que nos permite determinar si se hizo o no la transacción a la base de datos.

se llama de la siguiente manera, para el siguiente ejemplo en un Boton de Insertar

```

private void BtInsertar_Click(object sender, EventArgs e)
{
    i = Convert.ToInt32(TxtId.Text);
    n = TxtNombre.Text.ToUpper();
    a = TxtApellido.Text.ToUpper();
    d = TxtDireccion.Text.ToUpper();
    t = TxtTelefono.Text.ToUpper();
    int resultado = ConjuntoDatos.Consultas4("InsertUsuario", i, n, a, d,
t);

    if (resultado > 0)
    {
        MessageBox.Show("Usuario Guardado Con Exito!!", "Guardado",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
    {
        MessageBox.Show("No se pudo guardar el Usuario", "Fallo!!",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}
}

```

// esta línea llena un dataGrid usando un método llamado consultas que recibe un procedure

```
DtgUsuario.DataSource = ConjuntoDatos.Consultas(("p_usuario"));
```

.....
 Metodo que retorna un entero si se realiza una transacción de un procedimiento que recibe un parámetro

```

public static int Consultas3(string query, int i)
{
    //DataSet dsRetorno = new DataSet();
    int retorno;
}

```

```

        DataTable dtRetorno = new DataTable();
        MySqlCommand sqlCmd = new MySqlCommand();
        using (sqlCmd)
        {
            sqlCmd.Connection = BdComun.ObtenerConexion();
            sqlCmd.CommandType = CommandType.StoredProcedure;
            sqlCmd.CommandText = query;
            sqlCmd.Parameters.AddWithValue("id_u", i);
            retorno = sqlCmd.ExecuteNonQuery();
        }
        using (MySqlDataAdapter sqlDA = new MySqlDataAdapter(sqlCmd))
        {
            sqlDA.Fill(dtRetorno);
        }
        return retorno;
    }
}

```

.....

Metodo que recibe un string como nombre de procedure y un string como nombre de campo varchar o texto no numerico que permitira ejecutar una transacción

```

public static DataTable Consultas2(string query,string t)
{
    DataTable dtRetorno = new DataTable();
    MySqlCommand sqlCmd = new MySqlCommand();
    using (sqlCmd)
    {
        sqlCmd.Connection = BdComun.ObtenerConexion();
        sqlCmd.CommandType = CommandType.StoredProcedure;
        sqlCmd.CommandText = query;
        sqlCmd.Parameters.AddWithValue("no_u", t);
    }
    using (MySqlDataAdapter sqlDA = new MySqlDataAdapter(sqlCmd))
    {
        sqlDA.Fill(dtRetorno);
    }
    if (dtRetorno.Rows.Count == 0)
    {
        MessageBox.Show("No se encontro el registro", "Buscar",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    return dtRetorno;
}

```

Llena para el siguiente ejemplo una grilla que recibe un parámetro de busqueda
t = TxtNombre.Text;
DtgUsuario.DataSource = ConjuntoDatos.Consultas2("UsuarioBuscar", t);

Tambien se puede ejecutar este método para cargar una grilla teniendo en cuenta lo escrito en un textbox con un procedimiento almacenado que tenga la sentencia:

```

CREATE DEFINER=`root`@`localhost` PROCEDURE `SelectLike`(in no_u varchar(45))
BEGIN
SELECT * FROM pruebasena.usuarios WHERE nombre LIKE concat('%',no_u,'%') ;
END

```

```

private void TxtNombre_TextChanged(object sender, EventArgs e)
{
    t = TxtNombre.Text;
}

```

```
        DtgUsuario.DataSource = ConjuntoDatos.Consultas2("SelectLike", t);  
    }  
}
```

Cargar los trxtbox con los datos seleccionados de una grilla

```
private void DtgUsuario_CellContentClick(object sender, DataGridViewCellEventArgs e)  
{  
    TxtId.Text = DtgUsuario.CurrentRow.Cells[0].Value.ToString();  
    TxtNombre.Text= DtgUsuario.CurrentRow.Cells[1].Value.ToString();  
    TxtApellido.Text= DtgUsuario.CurrentRow.Cells[2].Value.ToString();  
    TxtDireccion.Text= DtgUsuario.CurrentRow.Cells[3].Value.ToString();  
    TxtTelefono.Text= DtgUsuario.CurrentRow.Cells[4].Value.ToString();  
}
```