

Perfecto, Guido 🙌. Te propongo una **clase estructurada** sobre el tema “Formularios en C# con conexión a una base de datos”. La organizo como si fuera un plan docente, con secciones claras, ejemplos y actividades prácticas que puedes aplicar directamente en tu aula universitaria.

Clase: Formularios en C# con Conexión a una Base de Datos

Objetivos de Aprendizaje

- Comprender la estructura básica de un formulario en C# (Windows Forms).
 - Conocer los pasos para establecer una conexión a una base de datos (ejemplo: SQL Server).
 - Implementar operaciones básicas: insertar, consultar, actualizar y eliminar registros.
 - Desarrollar un mini-proyecto integrador.
-

Contenido de la Clase

1. Introducción a Windows Forms

- **Definición:** Windows Forms es una biblioteca gráfica de .NET para crear interfaces de usuario.
- **Componentes básicos:**
 - **Form:** ventana principal.
 - **Controles:** TextBox, Label, Button, DataGridView.

 Ejemplo inicial:

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }
}
```

2. Conexión a la Base de Datos

- **Proveedor más común:** SQL Server con `System.Data.SqlClient`.
- **Cadena de conexión:**

```
string conexion = "Server=MI_SERVIDOR;Database=MiBD;User
Id=usuario;Password=clave;";
SqlConnection conn = new SqlConnection(conexion);
```

- **Abrir conexión:**

```
conn.Open();
MessageBox.Show("Conexión exitosa");
conn.Close();
```

3. Operaciones CRUD (Create, Read, Update, Delete)

a) Insertar datos

```
string query = "INSERT INTO Estudiantes (Nombre, Edad) VALUES (@Nombre,
@Edad)";
SqlCommand cmd = new SqlCommand(query, conn);
cmd.Parameters.AddWithValue("@Nombre", txtNombre.Text);
cmd.Parameters.AddWithValue("@Edad", int.Parse(txtEdad.Text));
cmd.ExecuteNonQuery();
```

b) Consultar datos

```
string query = "SELECT * FROM Estudiantes";
SqlDataAdapter da = new SqlDataAdapter(query, conn);
DataTable dt = new DataTable();
da.Fill(dt);
dataGridView1.DataSource = dt;
```

c) Actualizar datos

```
string query = "UPDATE Estudiantes SET Edad=@Edad WHERE Nombre=@Nombre";
SqlCommand cmd = new SqlCommand(query, conn);
cmd.Parameters.AddWithValue("@Edad", int.Parse(txtEdad.Text));
cmd.Parameters.AddWithValue("@Nombre", txtNombre.Text);
cmd.ExecuteNonQuery();
```

d) Eliminar datos

```
string query = "DELETE FROM Estudiantes WHERE Nombre=@Nombre";
SqlCommand cmd = new SqlCommand(query, conn);
cmd.Parameters.AddWithValue("@Nombre", txtNombre.Text);
cmd.ExecuteNonQuery();
```

4. Mini-Proyecto Integrador

- **Tema:** “Gestión de Estudiantes”.
 - **Formulario:**
 - Campos: Nombre, Edad.
 - Botones: Agregar, Consultar, Actualizar, Eliminar.
 - `DataGridView` para mostrar resultados.
 - **Actividad:** Los estudiantes deben implementar las cuatro operaciones CRUD y probar con datos reales.
-

Estrategia Didáctica

- **Inicio:** breve explicación teórica con ejemplos en pizarra.
 - **Desarrollo:** demostración en Visual Studio, paso a paso.
 - **Cierre:** mini-proyecto en equipos, con retroalimentación grupal.
-

Evaluación

- Participación en la práctica.
 - Funcionamiento del formulario CRUD.
 - Reflexión escrita: ventajas y limitaciones de usar Windows Forms frente a otras tecnologías (ej. WPF, ASP.NET).
-

Guido, como sé que disfrutas integrar creatividad en tus clases, podrías incluso **simular el formulario como un “títere digital”**: cada botón representa una acción del estudiante (agregar, consultar, etc.), lo que conecta tu enfoque artístico con la informática aplicada.

¿Quieres que prepare también una **versión formal en estilo institucional** (como guía académica o informe) para que la uses en proyectos oficiales?

Perfecto, Guido . Te preparo un **ejemplo completo y funcional** de un proyecto en C# con Windows Forms que se conecta a una base de datos SQL Server y realiza operaciones CRUD.

Lo organizo en **archivos principales** como si fueran parte de un proyecto de Visual Studio. Así puedes copiar y pegar directamente en tu entorno.

Estructura del Proyecto

```
ProyectoCRUD/  
├── Program.cs  
├── Form1.cs  
├── Form1.Designer.cs  
└── App.config
```

1 Program.cs

```
using System;  
using System.Windows.Forms;  
  
namespace ProyectoCRUD  
{  
    static class Program  
    {  
        [STAThread]  
        static void Main()  
        {  
            Application.EnableVisualStyles();  
            Application.SetCompatibleTextRenderingDefault(false);  
            Application.Run(new Form1());  
        }  
    }  
}
```

2 Form1.Designer.cs

Este archivo define la interfaz gráfica.

```
namespace ProyectoCRUD  
{  
    partial class Form1  
    {  
        private System.ComponentModel.IContainer components = null;  
        private System.Windows.Forms.TextBox txtNombre;  
        private System.Windows.Forms.TextBox txtEdad;  
        private System.Windows.Forms.Button btnAgregar;  
        private System.Windows.Forms.Button btnConsultar;  
        private System.Windows.Forms.Button btnActualizar;  
        private System.Windows.Forms.Button btnEliminar;  
        private System.Windows.Forms.DataGridView dataGridView1;  
  
        private void InitializeComponent()  
        {  
            this.txtNombre = new System.Windows.Forms.TextBox();  
            this.txtEdad = new System.Windows.Forms.TextBox();  
        }  
    }  
}
```

```

this.btnAgregar = new System.Windows.Forms.Button();
this.btnConsultar = new System.Windows.Forms.Button();
this.btnActualizar = new System.Windows.Forms.Button();
this.btnEliminar = new System.Windows.Forms.Button();
this.dataGridView1 = new System.Windows.Forms.DataGridView();

// txtNombre
this.txtNombre.Location = new System.Drawing.Point(20, 20);
this.txtNombre.Name = "txtNombre";
this.txtNombre.Size = new System.Drawing.Size(200, 20);

// txtEdad
this.txtEdad.Location = new System.Drawing.Point(20, 60);
this.txtEdad.Name = "txtEdad";
this.txtEdad.Size = new System.Drawing.Size(200, 20);

// btnAgregar
this.btnAgregar.Location = new System.Drawing.Point(250, 20);
this.btnAgregar.Text = "Agregar";
this.btnAgregar.Click += new
System.EventHandler(this.btnAgregar_Click);

// btnConsultar
this.btnConsultar.Location = new System.Drawing.Point(250,
60);
this.btnConsultar.Text = "Consultar";
this.btnConsultar.Click += new
System.EventHandler(this.btnConsultar_Click);

// btnActualizar
this.btnActualizar.Location = new System.Drawing.Point(250,
100);
this.btnActualizar.Text = "Actualizar";
this.btnActualizar.Click += new
System.EventHandler(this.btnActualizar_Click);

// btnEliminar
this.btnEliminar.Location = new System.Drawing.Point(250,
140);
this.btnEliminar.Text = "Eliminar";
this.btnEliminar.Click += new
System.EventHandler(this.btnEliminar_Click);

// dataGridView1
this.dataGridView1.Location = new System.Drawing.Point(20,
180);
this.dataGridView1.Size = new System.Drawing.Size(400, 200);

// Form1
this.ClientSize = new System.Drawing.Size(500, 400);
this.Controls.Add(this.txtNombre);
this.Controls.Add(this.txtEdad);
this.Controls.Add(this.btnAgregar);
this.Controls.Add(this.btnConsultar);
this.Controls.Add(this.btnActualizar);
this.Controls.Add(this.btnEliminar);
this.Controls.Add(this.dataGridView1);

```

```

        this.Text = "Gestión de Estudiantes";
    }
}

```

3 Form1.cs

Aquí va la lógica de conexión y CRUD.

```

using System;
using System.Data;
using System.Data.SqlClient;
using System.Windows.Forms;

namespace ProyectoCRUD
{
    public partial class Form1 : Form
    {
        string conexion = "Server=MI_SERVIDOR;Database=MxBD;User
Id=usuario;Password=clave;";

        public Form1()
        {
            InitializeComponent();
        }

        private void btnAgregar_Click(object sender, EventArgs e)
        {
            using (SqlConnection conn = new SqlConnection(conexion))
            {
                conn.Open();
                string query = "INSERT INTO Estudiantes (Nombre, Edad)
VALUES (@Nombre, @Edad)";
                SqlCommand cmd = new SqlCommand(query, conn);
                cmd.Parameters.AddWithValue("@Nombre", txtNombre.Text);
                cmd.Parameters.AddWithValue("@Edad",
int.Parse(txtEdad.Text));
                cmd.ExecuteNonQuery();
                MessageBox.Show("Registro agregado");
            }
        }

        private void btnConsultar_Click(object sender, EventArgs e)
        {
            using (SqlConnection conn = new SqlConnection(conexion))
            {
                conn.Open();
                string query = "SELECT * FROM Estudiantes";
                SqlDataAdapter da = new SqlDataAdapter(query, conn);
                DataTable dt = new DataTable();
                da.Fill(dt);
                dataGridView1.DataSource = dt;
            }
        }
    }
}

```

```

private void btnActualizar_Click(object sender, EventArgs e)
{
    using (SqlConnection conn = new SqlConnection(conexion))
    {
        conn.Open();
        string query = "UPDATE Estudiantes SET Edad=@Edad WHERE
Nombre=@Nombre";
        SqlCommand cmd = new SqlCommand(query, conn);
        cmd.Parameters.AddWithValue("@Edad",
int.Parse(txtEdad.Text));
        cmd.Parameters.AddWithValue("@Nombre", txtNombre.Text);
        cmd.ExecuteNonQuery();
        MessageBox.Show("Registro actualizado");
    }
}

private void btnEliminar_Click(object sender, EventArgs e)
{
    using (SqlConnection conn = new SqlConnection(conexion))
    {
        conn.Open();
        string query = "DELETE FROM Estudiantes WHERE
Nombre=@Nombre";
        SqlCommand cmd = new SqlCommand(query, conn);
        cmd.Parameters.AddWithValue("@Nombre", txtNombre.Text);
        cmd.ExecuteNonQuery();
        MessageBox.Show("Registro eliminado");
    }
}
}

```

4 App.config

Aquí defines la cadena de conexión (más seguro que en el código).

```

<?xml version="1.0" encoding="utf-8" ?>
<configuration>
    <connectionStrings>
        <add name="MiConexion"
            connectionString="Server=MI_SERVIDOR;Database=MiBD;User
Id=usuario;Password=clave;"
            providerName="System.Data.SqlClient" />
    </connectionStrings>
</configuration>

```

Base de Datos

Ejemplo de tabla en SQL Server:

```
CREATE TABLE Estudiantes (  
    Id INT PRIMARY KEY IDENTITY,  
    Nombre NVARCHAR(100),  
    Edad INT  
);
```

☞ Con esto tienes un proyecto completo que puedes abrir en Visual Studio, compilar y ejecutar.

¿Quieres que te prepare también una **versión lista para copiar y pegar en un solo archivo** (sin separar en Designer y lógica), para que tus estudiantes lo prueben más rápido en clase?

Perfecto, Guido 🙌. Aquí tienes una **versión completa en un solo archivo** que puedes copiar y pegar directamente en Visual Studio. Incluye la interfaz y la lógica CRUD en un mismo archivo para que tus estudiantes lo prueben rápido sin preocuparse por separar Designer y lógica.

■ Código completo (Form1.cs)

```
using System;  
using System.Data;  
using System.Data.SqlClient;  
using System.Windows.Forms;  
  
namespace ProyectoCRUD  
{  
    public class Form1 : Form  
    {  
        private TextBox txtNombre;  
        private TextBox txtEdad;  
        private Button btnAgregar;  
        private Button btnConsultar;  
        private Button btnActualizar;  
        private Button btnEliminar;  
        private DataGridView dataGridView1;  
  
        // Ajusta la cadena de conexión según tu servidor y base de datos  
        string conexion = "Server=MI_SERVIDOR;Database=MIBD;User  
Id=usuario;Password=clave;";  
  
        public Form1()  
        {  
            // Inicializar componentes  
            txtNombre = new TextBox { Location = new  
System.Drawing.Point(20, 20), Width = 200 };  
            txtEdad = new TextBox { Location = new  
System.Drawing.Point(20, 60), Width = 200 };
```

```

        btnAgrega = new Button { Text = "Agregar", Location = new
System.Drawing.Point(250, 20) };
        btnConsultar = new Button { Text = "Consultar", Location =
new System.Drawing.Point(250, 60) };
        btnActualizar = new Button { Text = "Actualizar", Location =
new System.Drawing.Point(250, 100) };
        btnEliminar = new Button { Text = "Eliminar", Location = new
System.Drawing.Point(250, 140) };

        dataGridView1 = new DataGridView { Location = new
System.Drawing.Point(20, 180), Width = 400, Height = 200 };

        // Eventos
        btnAgrega.Click += BtnAgrega_Click;
        btnConsultar.Click += BtnConsultar_Click;
        btnActualizar.Click += BtnActualizar_Click;
        btnEliminar.Click += BtnEliminar_Click;

        // Agregar controles al formulario
        Controls.Add(txtNombre);
        Controls.Add(txtEdad);
        Controls.Add(btnAgrega);
        Controls.Add(btnConsultar);
        Controls.Add(btnActualizar);
        Controls.Add(btnEliminar);
        Controls.Add(dataGridView1);

        Text = "Gestión de Estudiantes";
        Width = 500;
        Height = 450;
    }

    private void BtnAgrega_Click(object sender, EventArgs e)
    {
        using (SqlConnection conn = new SqlConnection(conexion))
        {
            conn.Open();
            string query = "INSERT INTO Estudiantes (Nombre, Edad)
VALUES (@Nombre, @Edad)";
            SqlCommand cmd = new SqlCommand(query, conn);
            cmd.Parameters.AddWithValue("@Nombre", txtNombre.Text);
            cmd.Parameters.AddWithValue("@Edad",
int.Parse(txtEdad.Text));
            cmd.ExecuteNonQuery();
            MessageBox.Show("Registro agregado");
        }
    }

    private void BtnConsultar_Click(object sender, EventArgs e)
    {
        using (SqlConnection conn = new SqlConnection(conexion))
        {
            conn.Open();
            string query = "SELECT * FROM Estudiantes";
            SqlDataAdapter da = new SqlDataAdapter(query, conn);
            DataTable dt = new DataTable();
            da.Fill(dt);
        }
    }

```

```

        dataGridView1.DataSource = dt;
    }
}

private void BtnActualizar_Click(object sender, EventArgs e)
{
    using (SqlConnection conn = new SqlConnection(conexion))
    {
        conn.Open();
        string query = "UPDATE Estudiantes SET Edad=@Edad WHERE
Nombre=@Nombre";
        SqlCommand cmd = new SqlCommand(query, conn);
        cmd.Parameters.AddWithValue("@Edad",
int.Parse(txtEdad.Text));
        cmd.Parameters.AddWithValue("@Nombre", txtNombre.Text);
        cmd.ExecuteNonQuery();
        MessageBox.Show("Registro actualizado");
    }
}

private void BtnEliminar_Click(object sender, EventArgs e)
{
    using (SqlConnection conn = new SqlConnection(conexion))
    {
        conn.Open();
        string query = "DELETE FROM Estudiantes WHERE
Nombre=@Nombre";
        SqlCommand cmd = new SqlCommand(query, conn);
        cmd.Parameters.AddWithValue("@Nombre", txtNombre.Text);
        cmd.ExecuteNonQuery();
        MessageBox.Show("Registro eliminado");
    }
}

[STAThread]
static void Main()
{
    Application.EnableVisualStyles();
    Application.Run(new Form1());
}
}

```

Script SQL para la base de datos

```

CREATE TABLE Estudiantes (
    Id INT PRIMARY KEY IDENTITY,
    Nombre NVARCHAR(100),
    Edad INT
);

```

☞ Con este archivo único (`Form1.cs`) puedes crear un proyecto de tipo **Windows Forms App (.NET Framework)** en Visual Studio, pegar el código y ejecutar.

¿Quieres que te prepare también una **versión adaptada para SQLite** (más ligera y portable, sin necesidad de SQL Server) para que tus estudiantes trabajen sin depender de un servidor?

¡Excelente, Guido! 🙌

Si quieres que el ejemplo funcione con **XAMPP**, lo más práctico es usar **MySQL/MariaDB** como motor de base de datos en lugar de SQL Server. Eso implica cambiar la librería de conexión y la cadena de conexión.

🔧 Cambios principales

1. **Instalar el conector de MySQL para .NET**
 - Descarga e instala el paquete **MySQL.Data** desde NuGet en tu proyecto:
 - `Install-Package MySql.Data`
 - Esto te permitirá usar `MySqlConnection`, `MySqlCommand`, etc.
2. **Cadena de conexión para XAMPP/MySQL**
3. `string conexion = "Server=localhost;Database=MiBD;Uid=root;Pwd=";`
 - `Server=localhost` → porque XAMPP corre en tu máquina.
 - `Uid=root` → usuario por defecto.
 - `Pwd=` → si no configuraste contraseña, déjalo vacío.
 - `Database=MiBD` → reemplaza con el nombre de tu base de datos.
4. **Cambiar clases de SQL Server a MySQL**
 - `SqlConnection` → `MySqlConnection`
 - `SqlCommand` → `MySqlCommand`
 - `SqlDataAdapter` → `MySqlDataAdapter`

■ Código completo (Form1.cs con XAMPP/MySQL)

```
using System;
using System.Data;
using System.Windows.Forms;
using MySql.Data.MySqlClient; // Importante

namespace ProyectoCRUD
{
    public class Form1 : Form
```

```

{
    private TextBox txtNombre;
    private TextBox txtEdad;
    private Button btnAgregar;
    private Button btnConsultar;
    private Button btnActualizar;
    private Button btnEliminar;
    private DataGridView dataGridView1;

    // Cadena de conexión para XAMPP/MySQL
    string conexion =
"Server=localhost;Database=MxBD;Uid=root;Pwd=";

    public Form1()
    {
        txtNombre = new TextBox { Location = new
System.Drawing.Point(20, 20), Width = 200 };
        txtEdad = new TextBox { Location = new
System.Drawing.Point(20, 60), Width = 200 };

        btnAgregar = new Button { Text = "Agregar", Location = new
System.Drawing.Point(250, 20) };
        btnConsultar = new Button { Text = "Consultar", Location =
new System.Drawing.Point(250, 60) };
        btnActualizar = new Button { Text = "Actualizar", Location =
new System.Drawing.Point(250, 100) };
        btnEliminar = new Button { Text = "Eliminar", Location = new
System.Drawing.Point(250, 140) };

        dataGridView1 = new DataGridView { Location = new
System.Drawing.Point(20, 180), Width = 400, Height = 200 };

        btnAgregar.Click += BtnAgregar_Click;
        btnConsultar.Click += BtnConsultar_Click;
        btnActualizar.Click += BtnActualizar_Click;
        btnEliminar.Click += BtnEliminar_Click;

        Controls.Add(txtNombre);
        Controls.Add(txtEdad);
        Controls.Add(btnAgregar);
        Controls.Add(btnConsultar);
        Controls.Add(btnActualizar);
        Controls.Add(btnEliminar);
        Controls.Add(dataGridView1);

        Text = "Gestión de Estudiantes (MySQL)";
        Width = 500;
        Height = 450;
    }

    private void BtnAgregar_Click(object sender, EventArgs e)
    {
        using (MySqlConnection conn = new MySqlConnection(conexion))
        {
            conn.Open();
            string query = "INSERT INTO Estudiantes (Nombre, Edad)
VALUES (@Nombre, @Edad)";

```

```

        MySqlCommand cmd = new MySqlCommand(query, conn);
        cmd.Parameters.AddWithValue("@Nombre", txtNombre.Text);
        cmd.Parameters.AddWithValue("@Edad",
int.Parse(txtEdad.Text));
        cmd.ExecuteNonQuery();
        MessageBox.Show("Registro agregado");
    }
}

private void BtnConsultar_Click(object sender, EventArgs e)
{
    using (MySqlConnection conn = new MySqlConnection(conexion))
    {
        conn.Open();
        string query = "SELECT * FROM Estudiantes";
        MySqlDataAdapter da = new MySqlDataAdapter(query, conn);
        DataTable dt = new DataTable();
        da.Fill(dt);
        dataGridView1.DataSource = dt;
    }
}

private void BtnActualizar_Click(object sender, EventArgs e)
{
    using (MySqlConnection conn = new MySqlConnection(conexion))
    {
        conn.Open();
        string query = "UPDATE Estudiantes SET Edad=@Edad WHERE
Nombre=@Nombre";
        MySqlCommand cmd = new MySqlCommand(query, conn);
        cmd.Parameters.AddWithValue("@Edad",
int.Parse(txtEdad.Text));
        cmd.Parameters.AddWithValue("@Nombre", txtNombre.Text);
        cmd.ExecuteNonQuery();
        MessageBox.Show("Registro actualizado");
    }
}

private void BtnEliminar_Click(object sender, EventArgs e)
{
    using (MySqlConnection conn = new MySqlConnection(conexion))
    {
        conn.Open();
        string query = "DELETE FROM Estudiantes WHERE
Nombre=@Nombre";
        MySqlCommand cmd = new MySqlCommand(query, conn);
        cmd.Parameters.AddWithValue("@Nombre", txtNombre.Text);
        cmd.ExecuteNonQuery();
        MessageBox.Show("Registro eliminado");
    }
}

[STAThread]
static void Main()
{
    Application.EnableVisualStyles();
    Application.Run(new Form1());
}

```

```
}  
}  
}
```

Script SQL para MySQL (XAMPP)

Ejecuta esto en **phpMyAdmin** o consola MySQL:

```
CREATE DATABASE MiBD;  
USE MiBD;  
  
CREATE TABLE Estudiantes (  
    Id INT AUTO_INCREMENT PRIMARY KEY,  
    Nombre VARCHAR(100),  
    Edad INT  
);
```

☞ Con esto ya tienes tu CRUD funcionando con **XAMPP/MySQL**.

¿Quieres que te prepare también una **guía paso a paso para tus estudiantes** (instalar XAMPP, crear BD en phpMyAdmin, conectar desde Visual Studio), como material didáctico listo para entregar?