

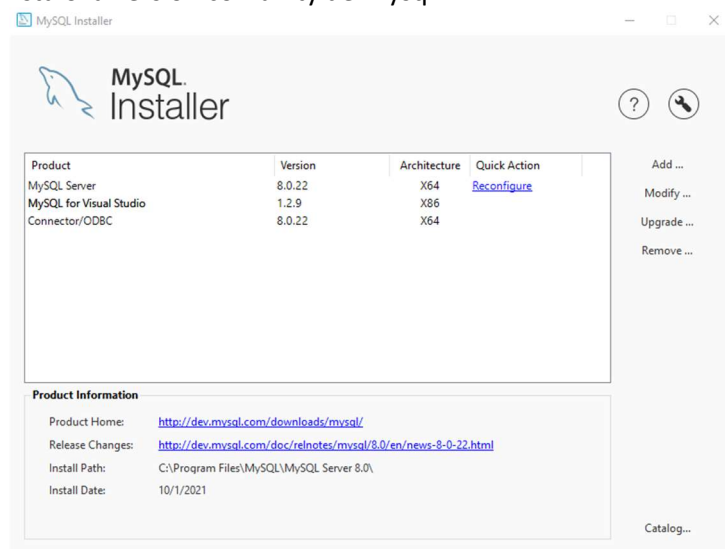
**DEPARTAMENTO:** Ciencias de la Ingeniería  
**CARRERA:** Sistemas de Información  
**CURSO:** Séptimo **PARALELO:** "A"  
**ASIGNATURA:** Plataformas de Desarrollo 1

**PROFESOR:** Mg. Luis Fernando Aguas B.  
**ESTUDIANTE:** Marco Antonio Ayala Lituma  
**DESCRIPCIÓN:** Deber 1-S4

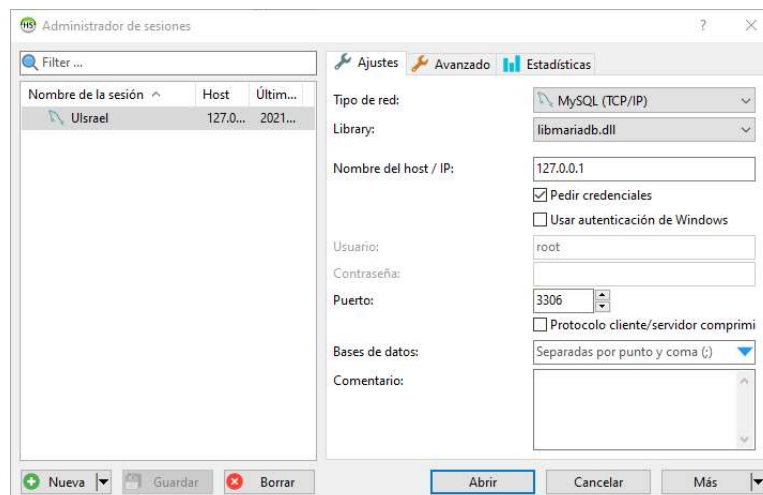
## TEMA: CRUD con MYSQL

### DESARROLLO:

Mediante Visual Studio se realiza la conexión con una base de datos en Mysql  
Previamente se instalo la versión community de Mysql



Ahora mediante un cliente como lo es HeidiSQL vamos a conectarnos a al motor de base de datos



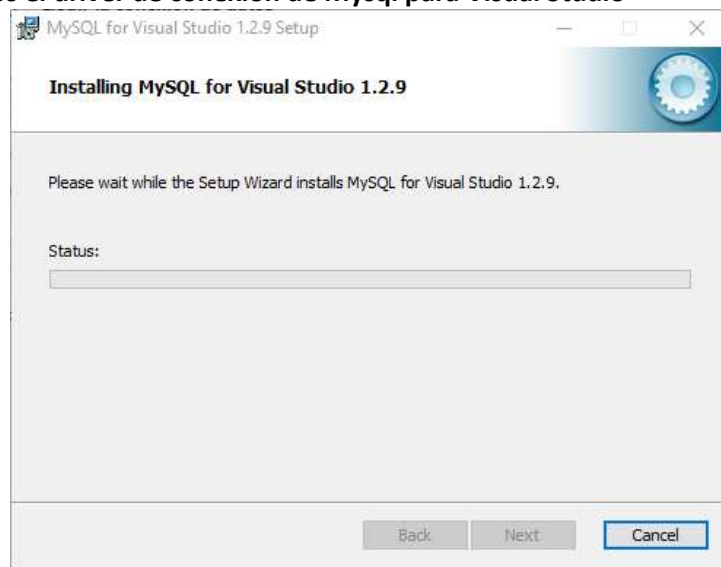
## Ahora creamos la base de datos usiael y también creamos la tabla Alumno

The screenshot shows the 'CREATE TABLE' dialog in SQL Server Enterprise Manager. The table name is 'Alumno'. The primary key is 'Id'. The columns are defined as follows:

#	Nombre	Tipo de datos	Longitud/Co...	Sin signo	Permitir NULL	Rellenar con ceros	Predeterminado	Comentario	Collation
1	Id	INT	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT...		
2	Nombre	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeter...		
3	Apellido	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeter...		
4	Edad	INT	2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeter...		

Buttons at the bottom: Ayuda, Descartar, Guardar.

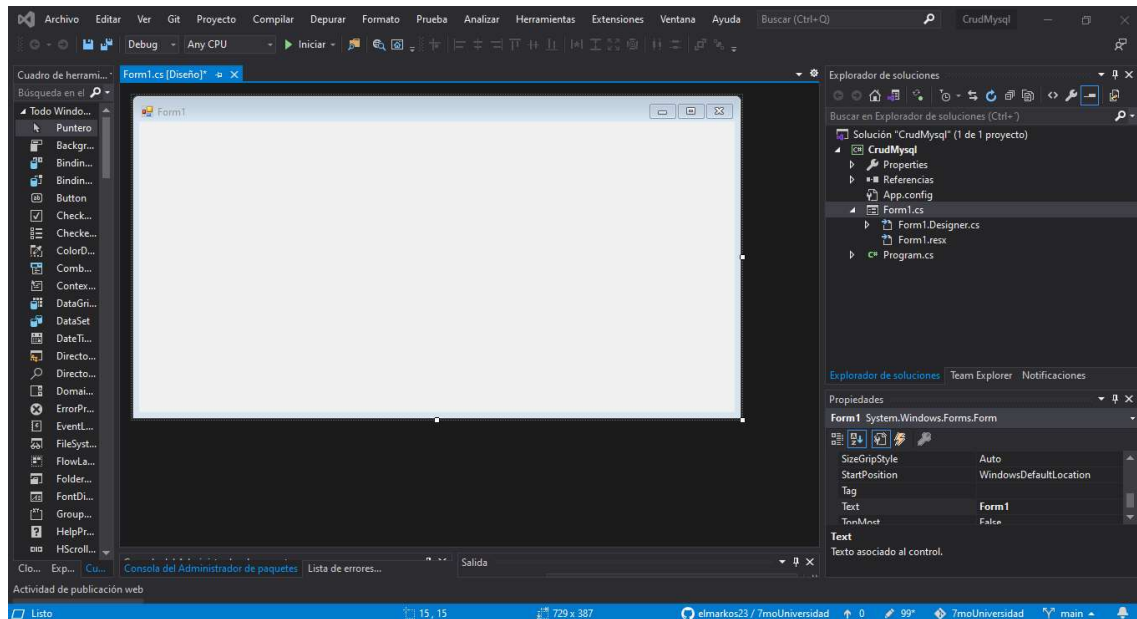
## Ahora Instalamos el driver de conexión de Mysql para Visual Studio



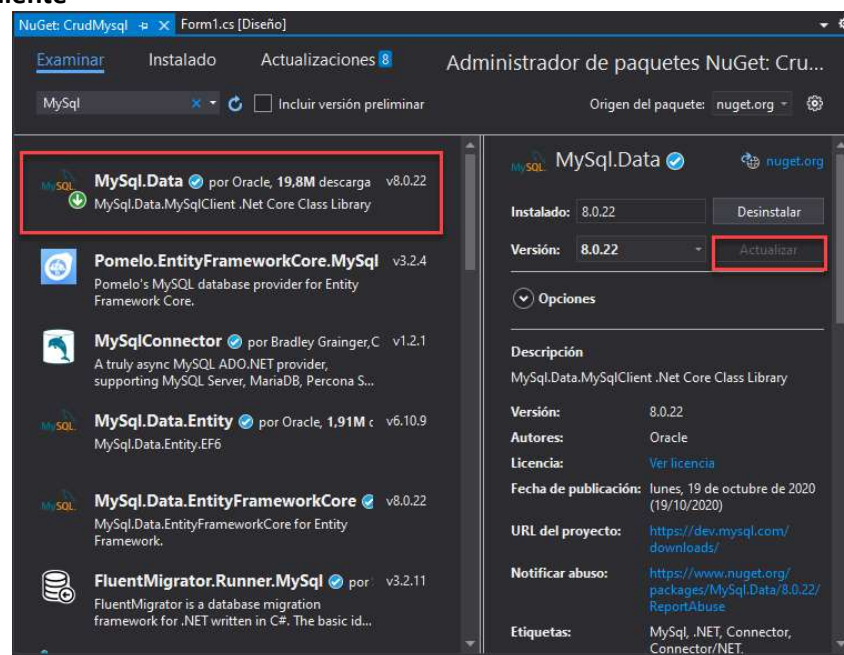
## Aquí procedemos a crear nuestro proyecto, de tipo Windows forms y lo ubicamos en nuestro repositorio común de proyectos

The screenshot shows the 'Configure su nuevo proyecto' dialog in Visual Studio. The project type is 'Aplicación de Windows Forms (.NET Framework)'. The project name is 'CrudMysql'. The location is 'D:\Github\7moUnidad\Plataformas Desarrollo\7.Semana5\'. The solution name is 'CrudMysql'. The checkbox 'Colocar la solución y el proyecto en el mismo directorio' is checked. The framework is '.NET Framework 4.7.1'. Buttons at the bottom: Atrás, Crear.

## Ya con el proyecto creado debería quedarnos de esta manera el explorado de soluciones

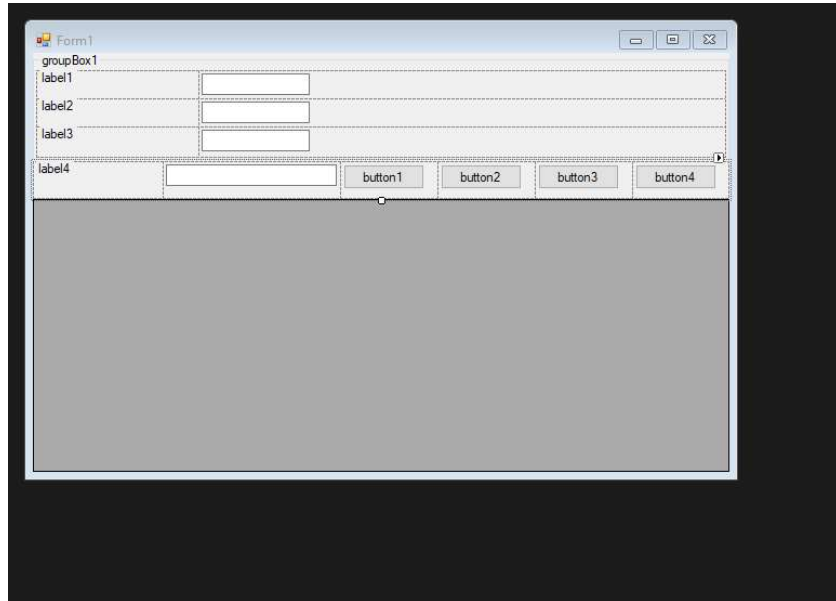


Para tener las ddls de conexión a la base e datos mysql necesitamos instalar mediante Nuget correspondiente

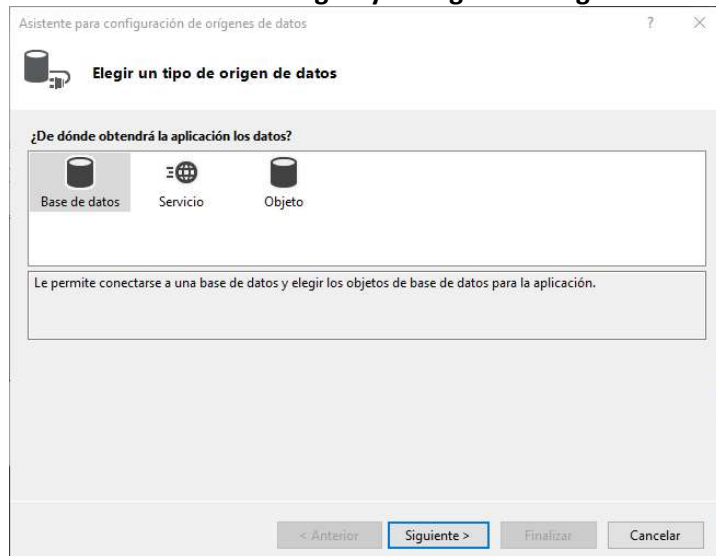


Ahora si diseñamos los controles de esta manera

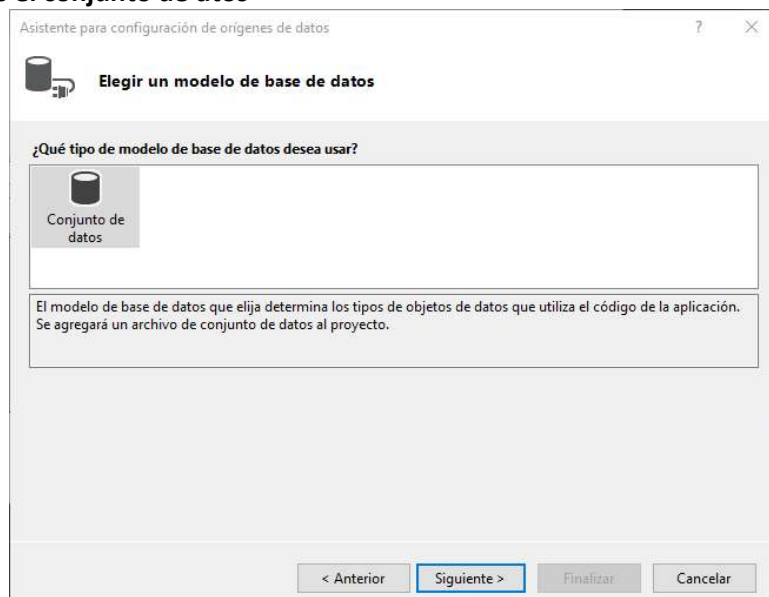




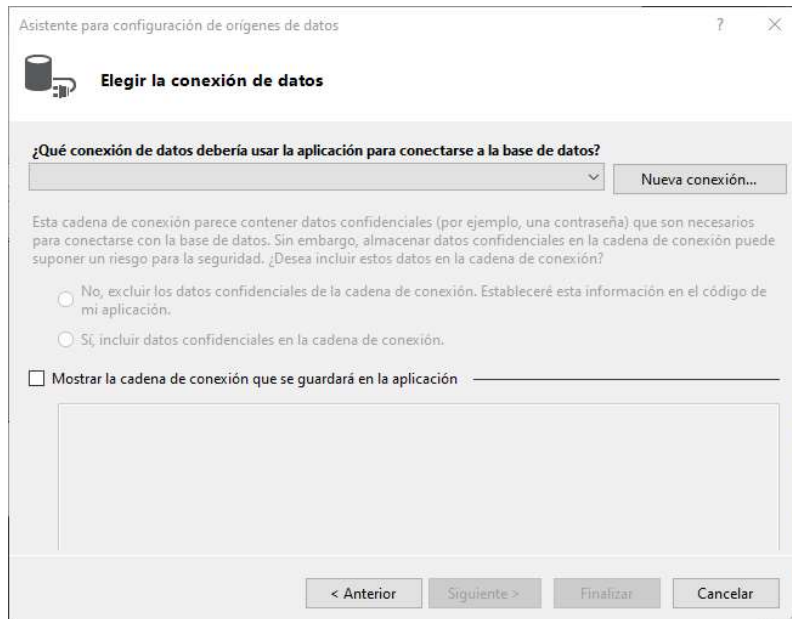
Ahora en esta parte vamos a dar clic en el grid y configurar el orgiente datos



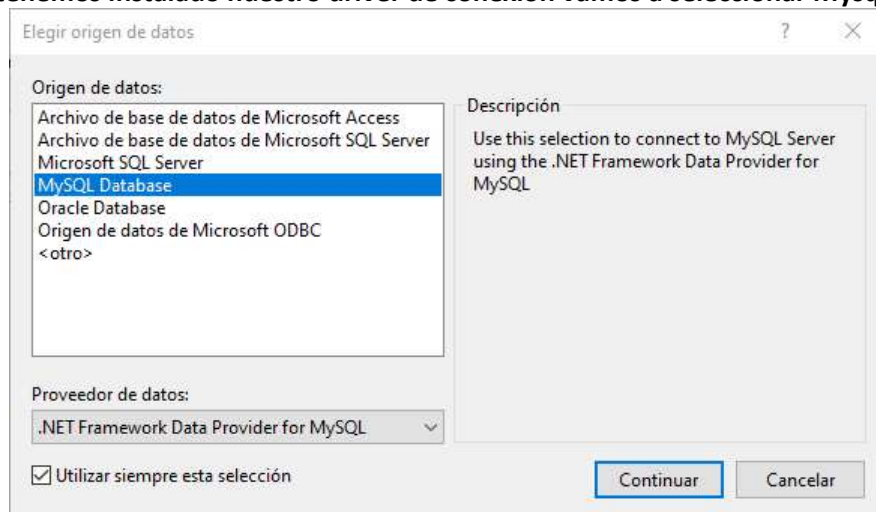
Seleccionamos el conjunto de dtos



En esta opción vamos a seleccionar nueva conexión



Como ya tenemos instalado nuestro driver de conexión vamos a seleccionar MySQL database



Aquí vamos a ingresar las credenciales de acceso al motor de base de datos que previamente ingresamos en la instalación

**Agregar conexión** ? X

Especifique la información para conectarse al origen de datos seleccionado o haga clic en "Cambiar" para elegir otro origen o proveedor de datos.

Origen de datos:

MySQL Database (MySQL Data Provider) Cambiar...

Server name: 127.0.01

User name: root

Password: ●●●●●●

☐ Save my password

Database name: uisrael ▾

Avanzadas...

Probar conexión Aceptar Cancelar

Ya he hecho eso.

Ahora vamos a crear una clase con el nombre Conexión y vamos a ingresar el siguiente código

```

public class Conexion
{
    MySqlConnection conn;

    public void Conectar()
    {
        conn = new MySqlConnection("server=127.0.01;user id=root;password=Maal2308;database=uisrael");
        conn.Open();
    }

    public void Desconectar()
    {
        conn.Close();
    }

    public void EjecutarSql(String consulta)
    {
        MySqlCommand con = new MySqlCommand(consulta, conn);
        int filasAfectadas = con.ExecuteNonQuery();
        if (filasAfectadas > 0)
        {
            MessageBox.Show("Operacion realizada correctamente", "La base de datos a sido modificada", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        else
        {
            MessageBox.Show("No se ha conectado a la base de datos", "Error del sistema", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }

    public void Actualizar(DataGridView dg, String consulta)
    {
        this.Conectar();
        System.Data.DataSet ds = new System.Data.DataSet();
        MySqlDataAdapter da = new MySqlDataAdapter(consulta, conn);
        da.Fill(ds, "alumno");
        dg.DataSource = ds;
        dg.DataMember = "alumno";
        this.Desconectar();
    }
}

```

Ahora en el codigo fuente del formulario que diseñamos vamos a escribir el siguiente codigo  
Vamos a denifir una variables globales que nos ayudaran a manejo de la base de datos y controla eventos del mismo.





```
public Conexion con = new Conexion();
int Id;
Boolean editar;
```

Creamos este método para actualizar el grid con una sentencia sql a la tabla

```
4 referencias | 0 cambios | 0 autores, 0 cambios
public void ActualizarGrid()
{
    con.Actualizar(dataGridView1, "select * from alumno");
}
```

Ahora creamos este método para limpiar los controles de texto

```
2 referencias | 0 cambios | 0 autores, 0 cambios
public void Limpiar()
{
    textBox1.Text = "";
    textBox2.Text = "";
    textBox3.Text = "";
    textBox4.Text = "";
}
```

Ahora creamos el evento load el cual va a cargar y invocar el método antes creado

```
private void Form1_Load(object sender, EventArgs e)
{
    editar = false;
    this.ActualizarGrid();
}
```

Ahora en botón de guardar vamos a escribir este código

```
1 referencia | 0 cambios | 0 autores, 0 cambios
private void button1_Click(object sender, EventArgs e)
{
    if (editar)
    {
        con.Conectar();
        string consulta = "update alumno set Nombre='" + textBox1.Text + "',Apellido='" + textBox2.Text + "', Edad=" + textBox3.Text + " where Id=" + Id + ";";
        con.EjecutarSql(consulta);
        this.ActualizarGrid();
        this.Limpiar();
        editar = false;
    }
    else
    {
        con.Conectar();
        string consulta = "insert into Alumno(Nombre,Apellido,Edad) values ('" + textBox1.Text + "','" + textBox2.Text + "','" + textBox3.Text + "');";
        con.EjecutarSql(consulta);
        this.ActualizarGrid();
        this.Limpiar();
    }
}
```

Ahora el botón de Actualizar escribimos este código

```
1 referencia | 0 cambios | 0 autores, 0 cambios
private void button3_Click(object sender, EventArgs e)
{
    editar = true;
    Id = int.Parse(this.dataGridView1.CurrentRow.Cells[0].Value.ToString());
    textBox1.Text = this.dataGridView1.CurrentRow.Cells[1].Value.ToString();
    textBox2.Text = this.dataGridView1.CurrentRow.Cells[2].Value.ToString();
    textBox3.Text = this.dataGridView1.CurrentRow.Cells[3].Value.ToString();
}
```

Ahora en el texto de textbox4 que es un campo de búsqueda vamos al evento keyup y escribimos el siguiente código

```
1 referencia | 0 cambios | 0 autores, 0 cambios
private void textBox4_KeyUp(object sender, KeyEventArgs e)
{
    con.Actualizar(dataGridView1, "select * from Alumno where Nombre like '" + textBox4.Text + "%'");
}
```

Ahora escribimos el siguiente código para el botón eliminar

```
1 referencia | 0 cambios | 0 autores, 0 cambios
private void button4_Click(object sender, EventArgs e)
{
    Id = int.Parse(this.dataGridView1.CurrentRow.Cells[0].Value.ToString());
    var resultado = MessageBox.Show("desea eliminar el registro", "Confirmación de Eliminar", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if (resultado == DialogResult.Yes)
    {
        con.Conectar();
        con.EjecutarSql("delete from Alumno where Id=" + Id + ";");
        this.ActualizarGrid();
        con.Desconectar();
    }
}
```

Ahora probamos el funcionamiento, al copilar ya tenemos un dato almacenado

The screenshot shows a Windows application window titled 'Form1'. It contains a form with three text boxes labeled 'Nombre', 'Apellido', and 'Edad'. Below these is a search field labeled 'Buscar por Nombre' and three buttons: 'Guardar', 'Actualizar', and 'Eliminar'. At the bottom, there is a data grid with the following columns: 'Id', 'Nombre', 'Apellido', and 'Edad'. The first row contains the values '1', 'Marco', 'Ayala', and '30'. A second row with an asterisk in the 'Id' column is also visible.

	Id	Nombre	Apellido	Edad
▶	1	Marco	Ayala	30
*				

Probamos el botón guardar

This screenshot shows the same application window as before, but with the 'Nombre' field containing 'Carlos', 'Apellido' containing 'Narvaez', and 'Edad' containing '25'. The 'Guardar' button is highlighted with a red box. The data grid remains the same, showing the record for Marco Ayala.

	Id	Nombre	Apellido	Edad
▶	1	Marco	Ayala	30
*				

Resultado





Form1

Datos

Nombre

Apellido

Edad

Buscar por Nombre

	Id	Nombre	Apellido	Edad
▶	1	Marco	Ayala	30
	2	Carlos	Narvaez	25
*				

Ahora probamos el filtrado del texto del buscar nombre

Form1

Datos

Nombre

Apellido

Edad

Buscar por Nombre

	Id	Nombre	Apellido	Edad
▶	2	Carlos	Narvaez	25
*				

Ahora probamos el funcionamiento de editar

Form1

Datos

Nombre

Apellido

Edad

Buscar por Nombre

	Id	Nombre	Apellido	Edad
▶	1	Marco	Ayala	30
	2	Carlos	Narvaez	25
*				

Resultado

Form1

Datos

Nombre

Apellido

Edad

Buscar por Nombre

	Id	Nombre	Apellido	Edad
▶	1	Marco	Ayala Lituma	31
	2	Carlos	Narvaez	25
*				

**Finalmente probamos el método de eliminar**

Form1

Datos

Nombre

Apellido

Edad

Buscar por Nombre

	Id	Nombre	Apellido	Edad
▶	1	Marco	Ayala Lituma	31
	2	Carlos	Narvaez	25
*				

**En el resultado como hemos puesto que si deseamos eliminar.**

Form1

Datos

Nombre

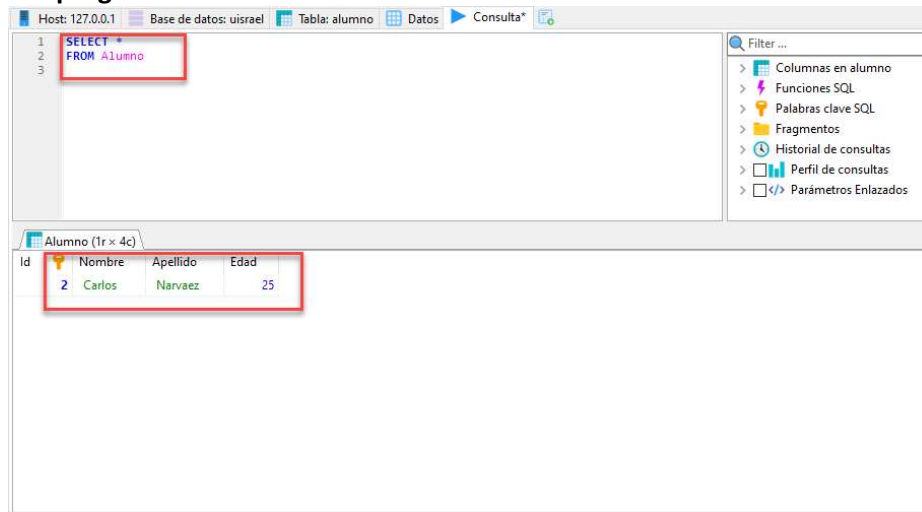
Apellido

Edad

Buscar por Nombre

	Id	Nombre	Apellido	Edad
▶	2	Carlos	Narvaez	25
*				

Si verificamos de forma manual la base de datos también comprobamos lo datos últimos cambios del programa



## BIBLIOGRAFÍA:

CRUD MYSQL - Csharp - Luis Fernando Aguas. Recuperado de <https://www.youtube.com/watch?v=ALrXOQhKPJU>

Instalador de Mysql, Recuperado de <https://dev.mysql.com/>