# Resumen del Programa de Gestión Académica

Este programa en C# permite gestionar un sistema académico utilizando una base de datos MySQL. Las funcionalidades incluidas son:

1. **Agregar Estudiante**: Permite ingresar un nuevo estudiante a la base de datos con datos como nombre, apellido, cédula, edad y nivel de educación.
2. **Listar Estudiantes**: Muestra una lista de todos los estudiantes almacenados en la base de datos.
3. **Buscar Estudiante**: Permite buscar a un estudiante específico usando su cédula.
4. **Modificar Estudiante**: Permite actualizar los datos de un estudiante existente.
5. **Eliminar Estudiante por Nombre y Apellido**: Elimina un estudiante de la base de datos basándose en su nombre y apellido.

El programa utiliza una cadena de conexión para interactuar con la base de datos MySQL y maneja errores para asegurar la robustez del sistema.

## Paso 1: Configurar el Entorno

1. **Instalar Visual Studio:**
   - Descarga e instala Visual Studio Community (o cualquier versión que prefieras).
2. **Instalar MySQL:**
   - Asegúrate de tener MySQL instalado en tu computadora.
   - Puedes usar XAMPP o instalar MySQL directamente.
3. **Crear la Base de Datos:**
   - Abre phpMyAdmin.
   - Crea una base de datos llamada sistema_gestion_educativa;.
   - Ejecuta el siguiente comando para crear la tabla `estudiantes`:

```
CREATE DATABASE sistema_gestion_educativa;

USE sistema_gestion_educativa;


CREATE TABLE estudiantes (
    codigo INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100),
    apellido VARCHAR(100),
    cedula VARCHAR(20),
    edad INT,
    nivel_educacion VARCHAR(50)
);
```

## Paso 2: Crear el Proyecto en Visual Studio

1. **Nuevo Proyecto:**
    - Abre Visual Studio y selecciona "Crear un nuevo proyecto".
    - Elige "Aplicación de consola (.NET Core)" y dale un nombre (por ejemplo, `GestionEstudiantes`).
2. **Agregar la Dependencia MySQL:**
    - Abre el "Administrador de paquetes NuGet".
    - Busca `MySql.Data` y selecciona instalarlo.

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;




//***************** Paso 1: Configuración
Inicial*************************************//

//*****************Importar las bibliotecas
necesarias:*****************************//

using MySql.Data.MySqlClient;




namespace lionenala

{

   internal class Program

   {



      //*********** Definir el espacio de nombres y la clase principal: *********//
```

```
        static string connectionString =
"server=localhost;database=migracion;user=nuevo_usuario;AllowUserVariables=True;";




        static void Main(string[] args)
        {


            //************ Paso 2: Configuración de la Consola ********************//


            Console.Title = "Universidad del Istmo de Panamá";
            Console.BackgroundColor = ConsoleColor.Black;
            Console.Clear();
            Console.ForegroundColor = ConsoleColor.White;
            Console.WriteLine("Universidad del Istmo de Panamá\n");
            Console.ResetColor();



            //********** Probar la conexión a la base de datos ******************//
            TestConnection();


            //********* Paso 3: Menú Principal*************************************//
```

```
//********* Implementar un bucle para el menú**********************//

try
{
    while (true)
    {
        Console.WriteLine("\n\nPlataforma de Gestión Académica");
        Console.WriteLine("1. Agregar Estudiante");
        Console.WriteLine("2. Listar Estudiantes");
        Console.WriteLine("3. Buscar Estudiante");
        Console.WriteLine("4. Modificar Estudiante");
        Console.WriteLine("5. Eliminar Estudiante por Nombre y Apellido");
        Console.WriteLine("6. Salir");
        Console.Write("Seleccione una opción: ");
        var option = Console.ReadLine();



        //************ Paso 4: Manejo de Opciones ***************//
        //************ Implementar el manejo de opciones********//

        switch (option)
        {
```

```
            case "1":
                AgregarEstudiante();
                break;
            case "2":
                ListarEstudiantes();
                break;
            case "3":
                BuscarEstudiante();
                break;
            case "4":
                ModificarEstudiante();
                break;
            case "5":
                EliminarEstudiantePorNombreApellido();
                break;
            case "6":
                return;
            default:
                Console.WriteLine("Opción no válida. Intente de nuevo.");
                break;
        }
    }
}
```

```
        catch (Exception ex)
        {
            Console.WriteLine($"Error: {ex.Message}");
        }
    }

    //********** Paso 5: Métodos para la Gestión de Estudiantes **********//
    //********** Método para probar la conexión:**********//

    static void TestConnection()
    {
        using (var connection = new MySqlConnection(connectionString))
        {
            try
            {
                connection.Open();
                Console.WriteLine("Conexión exitosa!\n");
            }
            catch (MySqlException ex)
            {
                Console.WriteLine($"Error de conexión: {ex.Message}\n");
            }
```

```
        }
    }




    //********* Método para agregar un estudiante:**********************************//


    static void AgregarEstudiante()
    {
        try
        {
            Console.Write("Nombre: ");
            string nombre = Console.ReadLine();
            Console.Write("Apellido: ");
            string apellido = Console.ReadLine();
            Console.Write("Cédula: ");
            string cedula = Console.ReadLine();
            Console.Write("Edad: ");

            if (!int.TryParse(Console.ReadLine(), out int edad))
            {
                Console.WriteLine("Por favor, ingrese una edad válida.");
                return;
```

```
        }

        Console.Write("Nivel de Educación: ");
        string nivelEducacion = Console.ReadLine();

        using (var connection = new MySqlConnection(connectionString))
        {
            connection.Open();
            using (var command = new MySqlCommand("INSERT INTO estudiantes (nombre, apellido, cedula, edad, nivel_educacion) VALUES (@nombre, @apellido, @cedula, @edad, @nivelEducacion)", connection))
            {
                command.Parameters.AddWithValue("@nombre", nombre);
                command.Parameters.AddWithValue("@apellido", apellido);
                command.Parameters.AddWithValue("@cedula", cedula);
                command.Parameters.AddWithValue("@edad", edad);
                command.Parameters.AddWithValue("@nivelEducacion", nivelEducacion);
                command.ExecuteNonQuery();
            }
        }

        Console.WriteLine("\nEstudiante agregado con éxito.\n");
        MostrarTablaEstudiantes();
    }
```

```
        catch (MySqlException ex)
        {
            Console.WriteLine($"Error de MySQL: {ex.Message}\n");
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Error: {ex.Message}\n");
        }
    }



    //************Método para mostrar la tabla de estudiantes:*****************//

    static void MostrarTablaEstudiantes()
    {
Console.WriteLine("_____
_____");
        Console.WriteLine(" | ID | Nombre          | Apellido        | Cédula          | Edad | Nivel de Educación      |");
Console.WriteLine("_____
_____");

        using (var connection = new MySqlConnection(connectionString))
```

```
        {
            connection.Open();
            using (var command = new MySqlCommand("SELECT * FROM estudiantes", connection))
            {
                using (var reader = command.ExecuteReader())
                {
                    while (reader.Read())
                    {
                        Console.WriteLine($" | {reader["codigo"],-3} | {reader["nombre"],-18} | {reader["apellido"],-18} | {reader["cedula"],-15} | {reader["edad"],-4} | {reader["nivel_educacion"],-22} |");
                    }
                }
            }
        }

Console.WriteLine("_____
_____");
    }

    //************ Método para listar estudiantes:**********************//
```

```
        static void ListarEstudiantes()
        {
            try
            {
                using (var connection = new MySqlConnection(connectionString))
                {
                    connection.Open();
                    using (var command = new MySqlCommand("SELECT * FROM estudiantes", connection))
                    {

Console.WriteLine("\n_____
_____
_____");

                        Console.WriteLine(" | ID | Nombre          | Apellido        | Cédula          | Edad | Nivel de Educación      |");

Console.WriteLine("_____
_____
_____");


                        using (var reader = command.ExecuteReader())
                        {
                            while (reader.Read())
                            {
                                Console.WriteLine($" | {reader["codigo"],-3} | {reader["nombre"],-18} | {reader["apellido"],-18} | {reader["cedula"],-15} | {reader["edad"],-4} | {reader["nivel_educacion"],-22} |");
```

```
                    }


Console.WriteLine("_____
_____
_____");
                }
            }
        }
    }
    catch (MySqlException ex)
    {
        Console.WriteLine($"Error de MySQL: {ex.Message}\n");
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error: {ex.Message}\n");
    }
}




//**************Método para buscar un estudiante:**********************

static void BuscarEstudiante()
```

```
    {
        try
        {
            Console.Write("Consulta la cédula del estudiante: ");
            string cedula = Console.ReadLine();

            if (string.IsNullOrWhiteSpace(cedula))
            {
                Console.WriteLine("La cédula no puede estar vacía.");
                return;
            }

            using (var connection = new MySqlConnection(connectionString))
            {
                connection.Open();
                using (var command = new MySqlCommand("SELECT * FROM estudiantes
WHERE cedula = @cedula", connection))
                {
                    command.Parameters.AddWithValue("@cedula", cedula);

                    using (var reader = command.ExecuteReader())
                    {

Console.WriteLine("\n_____
_____");
```

```
                    Console.WriteLine(" | ID | Nombre          | Apellido        | Cédula
| Edad | Nivel de Educación      |");

Console.WriteLine("_____
_____");



                    if (reader.Read())

                    {

                        Console.WriteLine($" | {reader["codigo"],-3} | {reader["nombre"],-18}
| {reader["apellido"],-18} | {reader["cedula"],-15} | {reader["edad"],-4} |
{reader["nivel_educacion"],-22} |");

Console.WriteLine("_____
_____");

                    }

                    else

                    {

                        Console.WriteLine("Estudiante no encontrado.\n");

                    }

                }

            }

        }

        catch (MySqlException ex)

        {

            Console.WriteLine($"Error de MySQL: {ex.Message}\n");

        }
```

```
        catch (Exception ex)
        {
            Console.WriteLine($"Error: {ex.Message}\n");
        }
    }


    //*************** Método para modificar un estudiante:****************//

    static void ModificarEstudiante()
    {
        try
        {
            Console.Write("Consulta la cédula del estudiante a modificar: ");
            string cedula = Console.ReadLine();

            using (var connection = new MySqlConnection(connectionString))
            {
                connection.Open();
                using (var command = new MySqlCommand("SELECT * FROM estudiantes
WHERE cedula = @cedula", connection))
                {
                    command.Parameters.AddWithValue("@cedula", cedula);
```

```
                using (var reader = command.ExecuteReader())
                {
                    if (reader.Read())
                    {
                        Console.Write("Nuevo Nombre (dejar vacío para no modificar): ");
                        string nuevoNombre = Console.ReadLine();
                        Console.Write("Nuevo Apellido (dejar vacío para no modificar): ");
                        string nuevoApellido = Console.ReadLine();
                        Console.Write("Nueva Edad (dejar vacío para no modificar): ");
                        string nuevaEdadInput = Console.ReadLine();
                        int? nuevaEdad = string.IsNullOrEmpty(nuevaEdadInput) ? (int?)null :
int.Parse(nuevaEdadInput);
                        Console.Write("Nuevo Nivel de Educación (dejar vacío para no
modificar): ");
                        string nuevoNivelEducacion = Console.ReadLine();

                        reader.Close();

                        using (var updateCommand = new MySqlCommand("UPDATE
estudiantes SET " +
                            "nombre = IFNULL(NULLIF(@nuevoNombre, ''), nombre), " +
                            "apellido = IFNULL(NULLIF(@nuevoApellido, ''), apellido), " +
                            "edad = IFNULL(@nuevaEdad, edad), " +
                            "nivel_educacion = IFNULL(NULLIF(@nuevoNivelEducacion, ''),
nivel_educacion) " +
```

```
                    "WHERE cedula = @cedula", connection))
                {
                    updateCommand.Parameters.AddWithValue("@nuevoNombre",
nuevoNombre);
                    updateCommand.Parameters.AddWithValue("@nuevoApellido",
nuevoApellido);
                    updateCommand.Parameters.AddWithValue("@nuevaEdad",
nuevaEdad);

updateCommand.Parameters.AddWithValue("@nuevoNivelEducacion",
nuevoNivelEducacion);
                    updateCommand.Parameters.AddWithValue("@cedula", cedula);


                    int rowsAffected = updateCommand.ExecuteNonQuery();
                    if (rowsAffected > 0)
                    {
                        Console.WriteLine("\nInformación del estudiante actualizada con
éxito.\n");
                    }
                    else
                    {
                        Console.WriteLine("No se realizaron cambios.\n");
                    }
                }

                // Mostrar la tabla actualizada
```

```
                    MostrarTablaEstudiantes();
                }
                else
                {
                    Console.WriteLine("Estudiante no encontrado.\n");
                }
            }
        }
    }
    catch (MySqlException ex)
    {
        Console.WriteLine($"Error de MySQL: {ex.Message}\n");
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error: {ex.Message}\n");
    }
}



//**********Método para eliminar un estudiante:******************//
```

```
static void EliminarEstudiantePorNombreApellido()
{
    try
    {
        Console.Write("Ingrese el nombre del estudiante a eliminar: ");
        string nombre = Console.ReadLine();
        Console.Write("Ingrese el apellido del estudiante a eliminar: ");
        string apellido = Console.ReadLine();

        using (var connection = new MySqlConnection(connectionString))
        {
            connection.Open();
            using (var command = new MySqlCommand("DELETE FROM estudiantes
WHERE nombre = @nombre AND apellido = @apellido", connection))
            {
                command.Parameters.AddWithValue("@nombre", nombre);
                command.Parameters.AddWithValue("@apellido", apellido);

                int rowsAffected = command.ExecuteNonQuery();
                if (rowsAffected > 0)
                {
                    Console.WriteLine("\nEstudiante eliminado con éxito.\n");
                }
```

```
                else
                {
                    Console.WriteLine("Estudiante no encontrado.\n");
                }
            }
        }
    }
    catch (MySqlException ex)
    {
        Console.WriteLine($"Error de MySQL: {ex.Message}\n");
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error: {ex.Message}\n");
    }
  }
}
```

## Paso : Ejecutar y Probar

1. **Ejecuta el Proyecto:**
    - Presiona `F5` para compilar y ejecutar tu aplicación.
    - Prueba todas las opciones del menú para asegurarte de que funcionan correctamente.

**[Paso 1: Seleccionar Usuarios Existentes]**

- **Explicación:**
    - "El primer paso es ver quiénes son los usuarios actuales en nuestra base de datos MySQL. Para esto, utilizamos el siguiente comando:"

**Mostrar Comando:**

```
SELECT User, Host FROM mysql.user;
```

**Demostración:**

- Ejecuta el comando en tu terminal o herramienta de gestión de base de datos y muestra la lista de usuarios.

**[Paso 2: Eliminar un Usuario]**

- **Explicación:**
    - "Ahora, si queremos eliminar un usuario específico, usamos el comando `DROP USER`. Por ejemplo, si queremos eliminar al usuario 'nuevo_usuario' que se conecta desde 'localhost', lo hacemos así:"

**Mostrar Comando:**

```
DROP USER 'nuevo_usuario'@'localhost';
```

**Demostración:**

- Ejecuta el comando y muestra cómo el usuario es eliminado de la lista.

**[Paso 3: Crear un Nuevo Usuario]**

- **Explicación:**
  - "Vamos a crear un nuevo usuario llamado 'nuevo_usuario' que se conectará desde 'localhost'. Para esto, usamos el comando `CREATE USER`:"

- **Mostrar Comando:**

```
CREATE USER 'nuevo_usuario'@'localhost' IDENTIFIED BY '';
```

**Demostración:**

- Ejecuta el comando y explica la opción de dejar la contraseña vacía, mencionando la importancia de usar contraseñas seguras.

**[Paso 4: Conceder Privilegios al Usuario]**

- **Explicación:**
  - "Después de crear el usuario, necesitamos darle permisos. Vamos a conceder todos los privilegios sobre la base de datos sistema_gestion_educativa; al usuario 'nuevo_usuario':"

**Mostrar Comando:**

```
GRANT ALL PRIVILEGES ON bomburbuja.* TO 'nuevo_usuario'@'localhost';
```

**Demostración:**

- Ejecuta el comando y muestra cómo el usuario ahora tiene permisos sobre la base de datos `sistema_gestion_educativa`.

**Paso 5: Actualizar Privilegios]**

- **Explicación:**
  - "Finalmente, para asegurarnos de que todos los cambios en los privilegios se apliquen, usamos el comando `FLUSH PRIVILEGES`:"

- **Mostrar Comando:**

```
FLUSH PRIVILEGES;
```

**Demostración:**

- Ejecuta el comando y explica su importancia para recargar los privilegios.

**[Conclusión]**

- **Resumen y Despedida:**
  - "Y eso es todo. Hoy hemos aprendido a gestionar usuarios y permisos en MySQL. Espero que esta guía les haya sido útil. No olviden suscribirse al canal, darle like al video y activar la campanita para más tutoriales como este. ¡Hasta la próxima!"