

UNIVERSIDAD NACIONAL DE CAJAMARCA

FACULTAD DE INGENIERÍA

ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA DE SISTEMAS



“Ejercicios”

Asignatura:

Programación Aplicada

Estudiante:

Caruajulca Tiglla Alex Eli

Cajamarca – Perú

2024

Este primer ejercicio está basado en el ejercicio anterior donde se incluye un método llamado `tbDniBusca_TextChanged` el cual autocompleta al seleccionar

```
public partial class CRUD_Entrenador : Window
{
    private SqlConnection con;

    public CRUD_Entrenador()
    {
        InitializeComponent();
        cargarDGrid();
    }
    public void limpiar()
    {
        tbApellido.Text = tbNombre.Text = tbEdad.Text = string.Empty;
        tbFleco.Text = tbDniBusca.Text = string.Empty;
    }
    public void conectar()
    {
        string conex = "Data Source=.;Initial Catalog=Federacion;Integrated
Security=True";
        con = new SqlConnection(conex);
    }
    public void cargarDGrid()
    {
        conectar();
        SqlCommand cmd = new SqlCommand("SELECT * FROM Entrenador", con);
        DataTable dt = new DataTable();
        con.Open();
        SqlDataReader dr = cmd.ExecuteReader();
        dt.Load(dr); //tengo la informacion en mi aplicación
        con.Close();
        dataGridView1.ItemsSource = dt.DefaultView;
    }

    private void btnInsertar_Click(object sender, RoutedEventArgs e)
    {
        try
        {
            conectar();
            //string inserta = "INSERT INTO Alumno VALUES
            (@apellido,@nombre,@dni,@edad," +
            //      "@peso, @fecha)";
            SqlCommand cmd = new SqlCommand("INSERT INTO Entrenador VALUES
            (@Apellidos,@Nombre,@Edad," +
            //      "@idFederacion)", con);
            cmd.Parameters.AddWithValue("@Apellidos", tbApellido.Text);
            cmd.Parameters.AddWithValue("@Nombre", tbNombre.Text);
            cmd.Parameters.AddWithValue("@Edad", tbEdad.Text);
            cmd.Parameters.AddWithValue("@idFederacion", tbFleco.Text);

            con.Open();
            cmd.ExecuteNonQuery();
            con.Close();
            cargarDGrid();
            MessageBox.Show("Información registrada", "Seguridad");
            limpiar();
        }
        catch (SqlException ex)
        {
            MessageBox.Show(ex.Message, "Seguridad SQL");
        }
    }

    private void btnActualizar_Click(object sender, RoutedEventArgs e)
```

```

{
    string actualizar = "UfIUPDATE Entrenador " +
        "SET Apellidos = '" + tbApelli.Text + "'", " +
        "Nombre = '" + tbNomb.Text + "'", " +
        "Edad = '" + tbEdad.Text + "'", " +
        "idFederacion = '" + tbfleso.Text + "'" +
        "WHERE id = '" + tbDniBusca.Text;

    try
    {
        conectar();
        SqlCommand cambio = new SqlCommand(actualizar, con);
        con.Open();
        cambio.ExecuteNonQuery();
        con.Close();
        cargarDGrid();
        MessageBox.Show("Actulización exitosa", "Seguridad");
        limpiar();
    }
    catch (SqlException ex)
    {
        MessageBox.Show(ex.Message, "Seguridad SQL");
        MessageBox.Show(actualizar);
    }
}

private void btEliminar_Click(object sender, RoutedEventArgs e)
{
    try
    {
        conectar();
        SqlCommand cmd = new SqlCommand("DELETE FROM Entrenador WHERE id
=" + tbDniBusca.Text, con);
        con.Open();
        cmd.ExecuteNonQuery();
        con.Close();
        cargarDGrid();
        limpiar();
        MessageBox.Show("Entrenador eliminado", "Seguridad");
    }
    catch (SqlException ex)
    {
        MessageBox.Show("Error al eliminar : " + ex.Message, "Seguridad
SQL");
    }
}

private void btLimpiar_Click(object sender, RoutedEventArgs e)
{
    try
    {
        conectar();
        SqlCommand cmd = new SqlCommand("TRUNCATE TABLE Entrenador",
con);
        con.Open();
        cmd.ExecuteNonQuery();
        con.Close();
        cargarDGrid();
        limpiar();
        MessageBox.Show("Todos los entrenadores han sido eliminados",

```

```

"Seguridad");
    }

    catch (SqlException ex)
    {
        MessageBox.Show("Error al eliminar : " + ex.Message, "Seguridad
SQL");
    }
}

private void Window_Loaded(object sender, RoutedEventArgs e)
{
    cargarDGrid();
}

private void dataGrid_SelectionChanged(object sender,
SelectionChangedEventArgs e)
{
    try
    {
        limpiar();
        if (dataGrid.SelectedItem != null)
        {
            DataRowView row = (DataRowView)dataGrid.SelectedItem;
            tbDniBusca.Text = row[property: "id"].ToString();
            tbApelli.Text = row["Apellidos"].ToString();
            tbNomb.Text = row["Nombre"].ToString();
            tbEdad.Text = row["Edad"].ToString();
            tbfleso.Text = row["idFederacion"].ToString();
        }
    }
    catch { }
}

private void tbDniBusca_TextChanged(object sender, TextChangedEventArgs
e)
{
    string consulta = "SELECT * FROM Entrenador WHERE id = @ID";
    SqlCommand cmd = new SqlCommand(consulta, con);
    cmd.Parameters.AddWithValue("@ID", tbDniBusca.Text);

    try
    {
        con.Open();
        SqlDataReader reader = cmd.ExecuteReader();

        if (reader.Read())
        {
            tbApelli.Text = reader["Apellidos"].ToString();
            tbNomb.Text = reader["Nombre"].ToString();
            tbEdad.Text = reader["Edad"].ToString();
            tbfleso.Text = reader["idFederacion"].ToString();
        }
        else
        {
        }

        reader.Close();
    }
    catch (SqlException ex)
    {
        MessageBox.Show("Error al obtener información: " + ex.Message,

```

```

"Seguridad SQL");
    }
    finally
    {
        con.Close();
    }
}
}

```

Ejercicio 2

```

<Grid>
    <DataGrid x:Name="dgPedidos" d:ItemsSource="{Binding}" Margin="0,116,0,-
116"/>
    <DatePicker x:Name="datePickerInicio" HorizontalAlignment="Left"
Margin="99,26,0,0" VerticalAlignment="Top"/>
    <DatePicker x:Name="datePickerFin" HorizontalAlignment="Left"
Margin="516,26,0,0" VerticalAlignment="Top"/>
    <Button x:Name="btn" Content="Buscar" HorizontalAlignment="Left"
Margin="371,82,0,0" VerticalAlignment="Top"/>
</Grid>

```

```

private void btnBuscar_Click(object sender, RoutedEventArgs e)
{
    // Paso 2: Validar que las fechas de inicio y fin estén seleccionadas
    if (datePickerInicio.SelectedDate == null || datePickerFin.SelectedDate ==
null)
    {
        MessageBox.Show("Por favor, seleccione fechas de inicio y fin.");
        return;
    }

    DateTime fechaInicio = datePickerInicio.SelectedDate.Value;
    DateTime fechaFin = datePickerFin.SelectedDate.Value;

    try
    {
        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            connection.Open();
            SqlCommand command = new SqlCommand("SELECT * FROM Orders WHERE
OrderDate BETWEEN @FechaInicio AND @FechaFin", connection);
            command.Parameters.AddWithValue("@FechaInicio", fechaInicio);
            command.Parameters.AddWithValue("@FechaFin", fechaFin);

            SqlDataAdapter adapter = new SqlDataAdapter(command);
            DataTable dataTable = new DataTable();
            adapter.Fill(dataTable);

            dgPedidos.ItemsSource = dataTable.DefaultView;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error al consultar la base de datos: " + ex.Message);
    }
}

```