**Ejercicio WPF**

El Instituto Tecnológico Autónomo de México requiere de un sistema de información para para registrar a los aspirantes interesados en los programas de ingeniería. Lo que nos solicitan es:

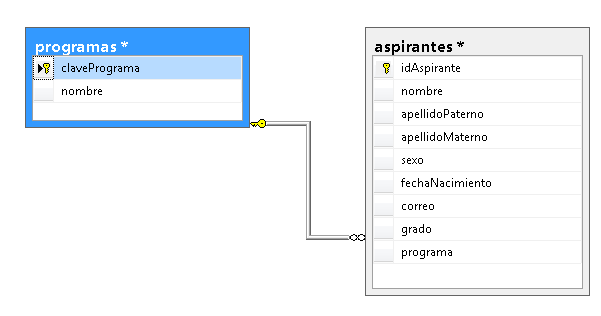
1. una interfaz para el registro de sus datos:

* Nombre completo
* Sexo
* Fecha de nacimiento
* Correo
* Grado (4to, 5to y 6to de preparatoria únicamente)
* Una lista con las ingenierías del ITAM para selección

1. Una base de datos que guarde la información solicitada
2. La aplicación debe poder dar de alta y modificar a los aspirantes.
3. La aplicación debe generar un reporte de todos los aspirantes dados de alta.
4. El reporte se puede consultar por programa elegido.

**Objetivo**: realizar un sistema que le ayude al director de cualquier carrera a llevar el control de sus aspirantes a su programa.

**Paso 0:** crea la base de datos llamada baseApirantes



Script generado:

USE [baseAspirantes]

GO

/\*\*\*\*\*\* Object: Table [dbo].[aspirantes] Script Date: 01/04/2019 10:38:33 p. m. \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[aspirantes](

[idAspirante] [int] NOT NULL,

[nombre] [nvarchar](50) NOT NULL,

[apellidoPaterno] [nvarchar](50) NOT NULL,

[apellidoMaterno] [nvarchar](50) NOT NULL,

[sexo] [nvarchar](20) NOT NULL,

[fechaNacimiento] [date] NOT NULL,

[correo] [nvarchar](50) NOT NULL,

[grado] [int] NOT NULL,

[clavePrograma] [int] NULL,

PRIMARY KEY CLUSTERED

(

[idAspirante] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[programas] Script Date: 01/04/2019 10:38:33 p. m. \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[programas](

[clavePrograma] [int] NOT NULL,

[nombre] [nvarchar](50) NOT NULL,

PRIMARY KEY CLUSTERED

(

[clavePrograma] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

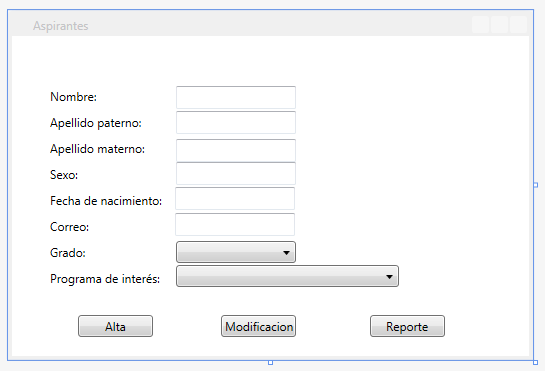
GO

ALTER TABLE [dbo].[aspirantes] WITH CHECK ADD FOREIGN KEY([clavePrograma])

REFERENCES [dbo].[programas] ([clavePrograma])

GO

**Paso 1:** crea la interfaz



<Label Name="lbNombre" Content="Nombre: " HorizontalAlignment="Left" Margin="33,47,0,0" VerticalAlignment="Top"/>

<Label Name="lbApellidoPaterno" Content="Apellido paterno: " HorizontalAlignment="Left" Margin="33,73,0,0" VerticalAlignment="Top"/>

<Label Name="lbApellodoMaterno" Content="Apellido materno: " HorizontalAlignment="Left" Margin="33,99,0,0" VerticalAlignment="Top"/>

<Label Name="lbSexo" Content="Sexo: " HorizontalAlignment="Left" Margin="33,125,0,0" VerticalAlignment="Top"/>

<Label Content="Fecha de nacimiento: " HorizontalAlignment="Left" Margin="33,151,0,0" VerticalAlignment="Top"/>

<Label Content="Correo: " HorizontalAlignment="Left" Margin="33,177,0,0" VerticalAlignment="Top"/>

<Label Content="Grado: " HorizontalAlignment="Left" Margin="33,203,0,0" VerticalAlignment="Top"/>

<Label Content="Programa de interés: " HorizontalAlignment="Left" Margin="33,229,0,0" VerticalAlignment="Top"/>

<TextBox HorizontalAlignment="Left" Height="23" Margin="164,50,0,0" TextWrapping="Wrap" Text="" Name="tbNombre" VerticalAlignment="Top" Width="120"/>

<TextBox HorizontalAlignment="Left" Height="23" Margin="164,75,0,0" TextWrapping="Wrap" Text="" Name="tbApellidoPaterno" VerticalAlignment="Top" Width="120"/>

<TextBox HorizontalAlignment="Left" Height="23" Margin="164,103,0,0" TextWrapping="Wrap" Text="" Name="tbApellidoMaterno" VerticalAlignment="Top" Width="120"/>

<TextBox HorizontalAlignment="Left" Height="23" Margin="164,126,0,0" TextWrapping="Wrap" Text="" Name="tbSexo" VerticalAlignment="Top" Width="120"/>

<TextBox HorizontalAlignment="Left" Height="23" Margin="163,151,0,0" TextWrapping="Wrap" Text="" Name="tbFechaDeNacimiento" VerticalAlignment="Top" Width="120"/>

<TextBox HorizontalAlignment="Left" Height="23" Margin="163,177,0,0" TextWrapping="Wrap" Text="" Name="tbCorreo" VerticalAlignment="Top" Width="120"/>

<ComboBox Name="cbPrograma" HorizontalAlignment="Left" Margin="164,229,0,0" VerticalAlignment="Top" Width="223"/>

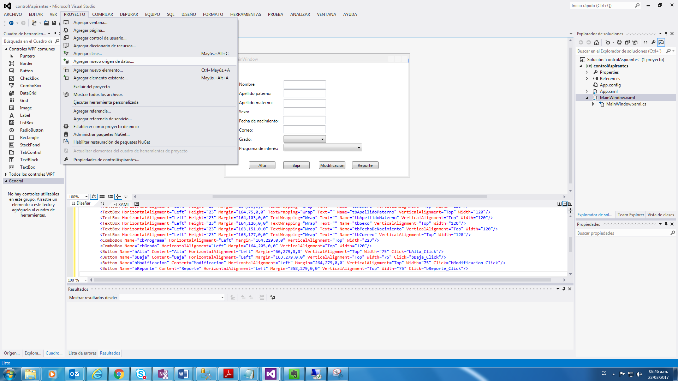
<ComboBox Name="cbGrado" HorizontalAlignment="Left" Margin="164,205,0,0" VerticalAlignment="Top" Width="120"/>

<Button Name="bAlta" Content="Alta" HorizontalAlignment="Left" Margin="66,279,0,0" VerticalAlignment="Top" Width="75" Click="bAlta\_Click" />

<Button Name="bModificacion" Content="Modificacion" HorizontalAlignment="Left" Margin="209,279,0,0" VerticalAlignment="Top" Width="75" Click="bModificacion\_Click" />

<Button Name="bReporte" Content="Reporte" HorizontalAlignment="Left" Margin="358,279,0,0" VerticalAlignment="Top" Width="75" Click="bReporte\_Click"/>

**Paso 2:** crea la conexión a la base de datos creada

 … completar los pasos

**Paso 3:** crea una clase Conexion.cs para realizar la conexión a la base de datos

public static SqlConnection agregarConexion(){

SqlConnection conexion = new SqlConnection("Data Source=CC102-11\SA;Initial Catalog=baseAspirantes;User ID=sa;Password=adminadmin");

conexion.Open();

return conexion;

}

**Paso 4:** en la misma clase crear un método estático para llenar el combo de programa de interés de la interfaz

public static void llenarComboProgramas(ComboBox cb)

{

try {

SqlConnection conexion = agregarConexion();

if (conexion != null) {

SqlCommand cmd = new SqlCommand("select nombre from programas", conexion);

SqlDataReader rd = cmd.ExecuteReader();

while (rd.Read())

cb.Items.Add(rd.GetString(0));

cb.SelectedIndex = 0; //poner el primer elemento en el combo

rd.Close();

conexion.Close();

}

else

MessageBox.Show("No se conenctó");

}catch(Exception ex) {

MessageBox.Show("error" + ex);

}

}

**Paso 5:**en el load de la ventana llenar el combo grado a mano con 4,5,6 como datos y el combo programas con la base y probar que funcione.

private void Window\_Loaded(object sender, RoutedEventArgs e)

{

Conexion.llenarComboProgramas(cbPrograma);

cbGrado.Items.Add(4);

cbGrado.Items.Add(5);

cbGrado.Items.Add(6);

cbGrado.SelectedIndex = 0;

}

**Paso 6:** hacer la funcionalidad del botón alta

Crear una clase Aspirante.cs

public String alta(Aspirante a)

{

String res = "";

int i;

try {

SqlConnection con = Conexion.agregarConexion();

SqlCommand cmd = new SqlCommand("select top(1) idAspirante from aspirantes order by idAspirante des", con);

SqlDataReader rd = cmd.ExecuteReader();

if (rd.Read()) {

i = rd.GetInt32(0) + 1;

rd.Close();

SqlCommand cmd2 = new SqlCommand(String.Format("insert into aspirantes values ({0}, '{1}', '{2}', '{3}', '{4}', '{5}', '{6}', {7}, {8}) ", i, a.nombre, a.apellidoPaterno, a.apellidoMaterno, a.sexo, a.fechaNac, a.correo, a.grado, a.programa), con);

cmd2.ExecuteNonQuery();

}

else {

rd.Close();

SqlCommand cmd2 = new SqlCommand(String.Format("insert into aspirantes values ({0}, '{1}', '{2}', '{3}', '{4}', '{5}', '{6}', {7}, {8}) ", 1, a.nombre, a.apellidoPaterno, a.apellidoMaterno, a.sexo, a.fechaNac, a.correo, a.grado, a.programa), con);

cmd2.ExecuteNonQuery();

}

res = "alata exitosa";

con.Close();

}catch(Exception ex) {

res = "alta no exitosa" + ex.Message;

}

return res;

}

En el evento de clic del botón del Alta agregar el código correspondiente:

private void bAlta\_Click(object sender, RoutedEventArgs e)

{

Aspirante a = new Aspirante(tbNombre.Text, tbApellidoPaterno.Text, tbApellidoMaterno.Text, tbSexo.Text, tbFechaDeNacimiento.Text, tbCorreo.Text, (int)cbGrado.SelectedItem, (int)cbPrograma.SelectedIndex);

MessageBox.Show(a.alta(a));

}

**Paso 7:** hacer la funcionalidad del botón de modificación

public String modifica(int programa, String nombre) {

String res = "";

try {

SqlConnection con = Conexion.agregarConexion();

SqlCommand cmd = new SqlCommand(String.Format("update aspirantes et clavePrograma = {0} where nombre='{1}'", programa, nombre), con);

cmd.ExecuteNonQuery();

res = "modificación exitosa";

} catch(Exception ex) {

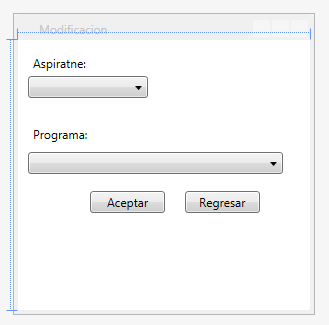
res = "error" + ex;

}

return res;

}

**Paso 8:** Crear la interfaz de modificación



<Label Content="Programa: " HorizontalAlignment="Left" Margin="10,81,0,0" VerticalAlignment="Top"/>

<ComboBox Name="cbProgramas" HorizontalAlignment="Left" Margin="10,112,0,0" VerticalAlignment="Top" Width="255"/>

<Button Name="bAceptar" Content="Aceptar" HorizontalAlignment="Left" Margin="72,151,0,0" VerticalAlignment="Top" Width="75" Click="bAceptar\_Click"/>

<Button Name="bRegresar" Content="Regresar" HorizontalAlignment="Left" Margin="167,151,0,0" VerticalAlignment="Top" Width="75" Click="bRegresar\_Click"/>

<Label Content="Aspiratne:" HorizontalAlignment="Left" Margin="10,10,0,0" VerticalAlignment="Top"/>

<ComboBox HorizontalAlignment="Left" Margin="10,36,0,0" VerticalAlignment="Top" Width="120"/>

**Paso 9:** Llenar los combos

En conexión agregar un llenarComboProgramas2, para llenar el nombre del aspirante

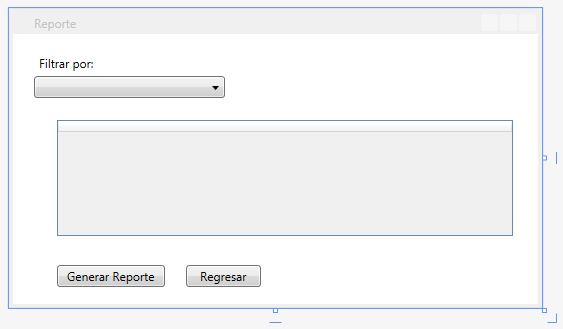
Mandar llamarlos en load de la ventana

**Paso 10:** Hacer la funcionalidad del botón aceptar

**Paso 11:** Hacer la funcionalidad del botón regresar

**Paso 12:** hacer la funcionalidad del botón de reporte

**Paso 13:** Crear la interfaz



<ComboBox Name="cbProgramas" HorizontalAlignment="Left" Margin="21,42,0,0" VerticalAlignment="Top" Width="191"/>

<Label Content="Filtrar por: " HorizontalAlignment="Left" Margin="21,16,0,0" VerticalAlignment="Top"/>

<Button Name="bGenerar" Content="Generar Reporte" HorizontalAlignment="Left" Margin="44,231,0,0" VerticalAlignment="Top" Width="108" Click="bGenerar\_Click"/>

<Button Name="bRegresar" Content="Regresar" HorizontalAlignment="Left" Margin="173,231,0,0" VerticalAlignment="Top" Width="75" Click="bRegresar\_Click"/>

<DataGrid Name="dGReporte" HorizontalAlignment="Left" Margin="44,86,0,0" VerticalAlignment="Top" Height="116" Width="456"/>

**Paso 14:** llenar el combo con los programas

**Paso 15:** hacer la funcionalidad del botón generar reporte

**Paso 16:** hacer la funcionalidad del botón regresar

**Paso 17:** Probar toda la funcionalidad