

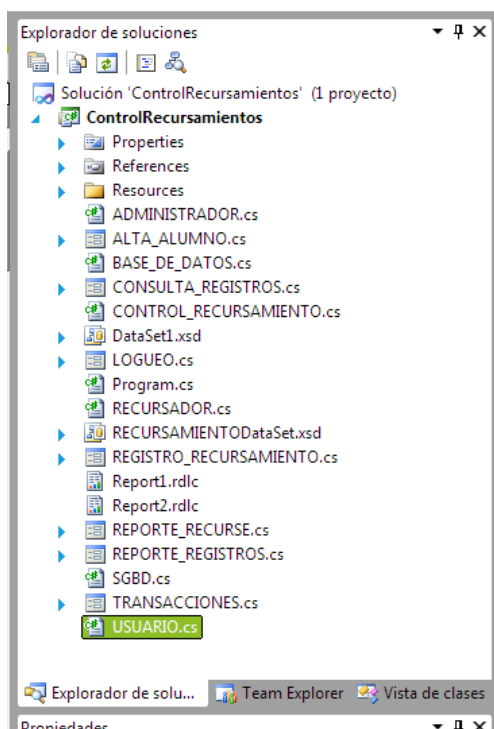
MANUAL TÉCNICO CONTROL RECURSAMIENTO

Es importante saber que el programa se compone de 7 clases y que es lo más relevante que tiene cada una de ellas

clase	Código de importancia	especificaciones
ADMINISTRADOR	<pre>using System.Data.SqlClient; namespace ControlRekursamientos { class ADMINISTRADOR:USUARIO { public bool insertar(string sql) { Base.cn.Open(); comando = new SqlCommand(sql, Base.cn); comando.ExecuteNonQuery(); Base.cn.Close(); } } }</pre>	<p>System.Data.SqlClient; es una librería que se agrega para tener acceso a SqlCommand, además podemos ver que ADMINISTRADOR hereda de la Clase USUARIO.</p> <p>Insertar es un método que retorna un valor booleano por el cual se mandara una cadena que contiene un comando de sql que ejecutar para agregar la boleta y el contraseña de un alumno no registrado</p>
BASE DE DATOS	<pre>public string cadena = @"Data Source=(local);Initial Catalog=RECURSAMIENTO;Integrated Security=True";</pre>	<p>RECURSAMIENTO es la base de datos, cadena es la cadena por el cual acercamos a esta</p>
CONTROL_RECURSMIENTO	<pre>public SGBD Base;</pre>	<p>SGBD es el tipo de dato que se requiere para formar el objeto base cual tendremos acceso a las transacciones por realizar</p>
Program	<pre>Application.Run(new LOGUEO());</pre>	<p>LOGUEO es el formulario con el que inicializara al ejecutar, esta clase no se puede eliminar ya que es la que crea por default Visual Studio</p>
RECURSADOR	<pre>class RECURSADOR: USUARIO { public string nombre; public string ape_pat; public string ape_mat; public int id_mat; public int id_grupo; public int id_prof; public string Nombre { get { return this.nombre; } set { this.nombre = value; } } }</pre>	<p>Clase RECURSADOR que hereda de USUARIO, las variables de id se insertan en la tabla ALUMNO de la base para que haga la inscripción, get y set respectivos setter y getter</p>
SGBD	<pre>public DataSet ds = new DataSet(); public SqlDataAdapter da; public SqlCommand comando; public SqlCommandBuilder cmb; public SGBD() { conectar(); }</pre>	<p>ds, da, comando y cmb son clases exclusivas para el uso de sql para tener acceso a ellas se requiere la librería</p> <p>using System.Data.SqlClient;</p> <p>SGBD es un constructor que inicializa con el método conectar el cual tiene la conexión a la</p>

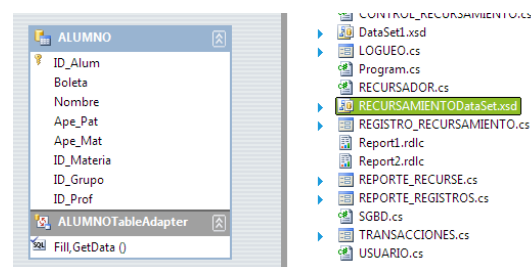
	<pre> public void conectar() { Base.cn = new SqlConnection(Base.cadena) </pre>	base
USUARIO	<pre> public string Id { get { return this.id; } set { this.id = value; } } public string Contraseña { get { return this.contraseña; } set { this.contraseña = value; } } </pre>	ID y contraseña son obtenidas al ingresar el usuario en el formulario logueo y las hereda a administrador y recursador

EXPLORADOR DE SOLUCIONES

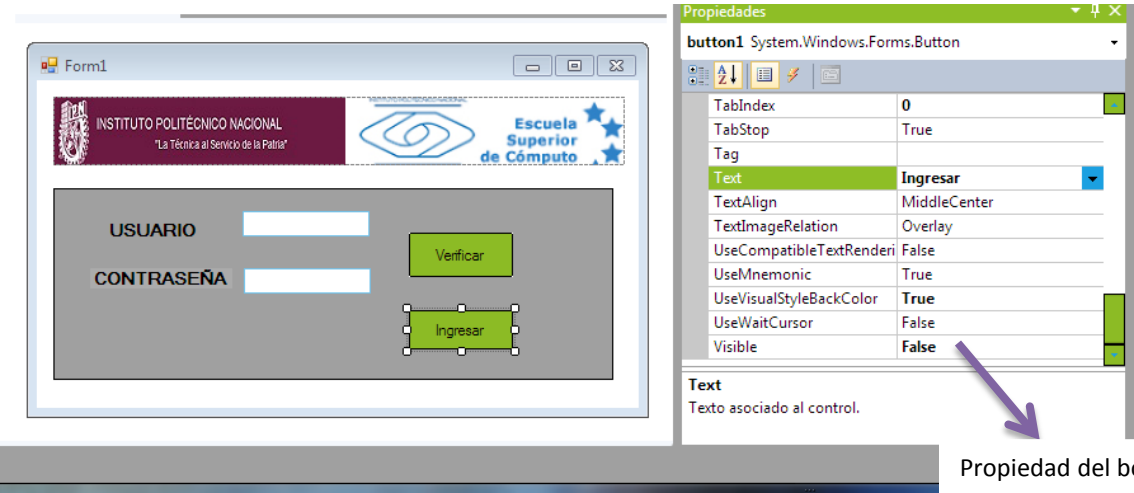


En la parte derecha de Visual Studio encontramos todas las clases usadas, los formularios con su respectivo nombre y cabe destacar RECURSAMIENTO.DataSet , Report1 y Report2.

RECURSAMIENTO.DataSet es utilizado al agregar dando click derecho en ControlRecursamientos-> Agregar Conjunto de datos y ahí agregamos el dataset con un store procedimiento que a su vez en el formulario REPORTE_RECURSE en el cual se agrega ese reporte para poderlo visualizar



ACCESO



Propiedad del botón ingresar para que no sea visible al iniciar el login

Al momento de hacer el login puede haber 2 posibilidades que el programa detecte el id de un recursador o el del administrador,

Formulario LOGUEO	
<pre>if (usuario.logueoRecursoador() == true) { MessageBox.Show("BIENVENIDO"); REGISTRO_RECURSAMIENTO registro = new REGISTRO_RECURSAMIENTO(); registro.BOLETA = textBox1.Text; registro.Show(); this.Hide(); } else if (usuario.loguepAdmin() == true) { MessageBox.Show("BIENVENIDO"); TRANSACCIONES transaccion = new TRANSACCIONES(); transaccion.Show(); this.Hide(); }</pre>	<p>Del objeto usuario se usa el metido logueo recursador de donde esta la función booleana</p> <p>registro.BOLETA = textBox1.Text; registro.Show(); línea que pasa el parametron dde id para que en el forulario REGISTRO_RECURSAMIENTO lo use para trabajar con la base de datos</p> <p>reanssion en el administrador es u objeto que llevara al formulario para que visualice que dese hacer</p>

REGISTRO_RECUSAMIENTO

button3 System.Windows.Forms.Button	
TextImageRelation	Overlay
UseCompatibleTextRenderi	False
UseMnemonic	True
UseVisualStyleBackColor	True
UseWaitCursor	False
Visible	False

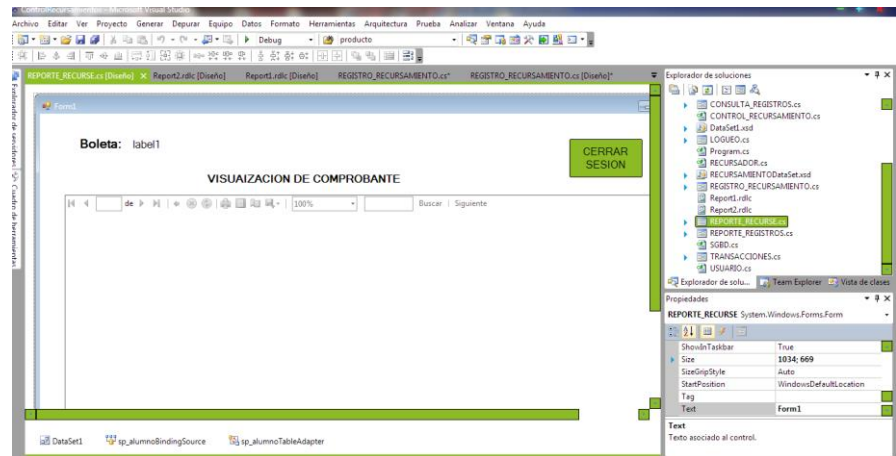
En las propiedades del botón NUVA MATERIA Y COMPROBANTE la visibilidad esta desactivada

Formulario REGISTRO_RECUSAMIENTO	
<pre>this.comboBox1.DataSource = transaccion.consultar2("MATERIA"); this.comboBox1.DisplayMember = "Descripcion"; this.comboBox1.ValueMember = "ID_Materia"; this.comboBox1.Refresh(); label6.Text = BOLETA; id = transaccion.ObtenerIdAlumno()+1; label12.Text = id.ToString(); string sqlcad = "SELECT * from ALUMNO where Boleta='" + BOLETA + "'"; SqlCommand caro = new SqlCommand(sqlcad, recursador.Base.cn); recursador.Base.cn.Open(); SqlDataReader asignar = caro.ExecuteReader(); if (asignar.Read() == true) { textBox1.Text = asignar["NOMBRE"].ToString(); textBox2.Text = asignar["Ape_Pat"].ToString(); textBox3.Text = asignar["Ape_Mat"].ToString(); textBox1.Enabled = false; textBox2.Enabled = false; textBox3.Enabled = false; } recursador.Base.cn.Close();</pre>	<p>Es importante ver que dentro de este formulario hay un evento LOAD el cual se encargara que al inicializar el formulario cara datos de las materias en el combobox, además que el sistema hace una transacción para verificar si el alumno está registrado en la base de datos pueda ingresar una materi a recusar pero ya no ingresar nuevos datos , por ello la propiedad Enabled desactivada</p> <p>Como se ve con asignar [" "].ToString rellenamos los datos del alumno correspondiente si existe su registro previo</p>

GENERACIÓN DE REPORTES



Report1 Y Report 2 son utilizados para cargar el reporte dentro del formulario REPORTE_RECURSE o REPORTE_REISTROS



FORMULARIO REPORTE RECURSE

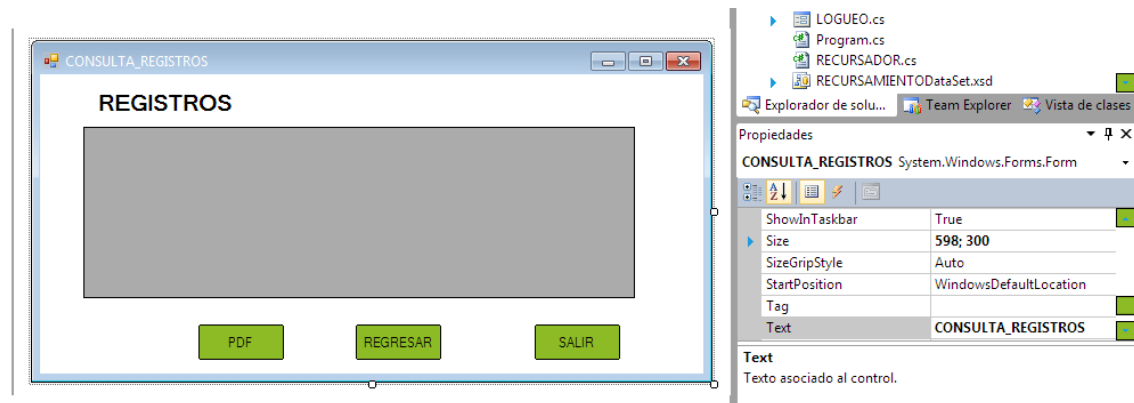
```
public string BOLETA
{
    get { return _boleta; }
    set { _boleta = value; }
}

label1.Text = BOLETA;

this.sp_alumnoTableAdapter.Fill(this.DataSet1.sp_alumno,
int.Parse(label1.Text));
this.reportViewer1.RefreshReport();
```

Es de suma importancia resaltar que exclusivamente en el formulario REPORTE RECURSE para enerar el reporte iternamente hay un store procedure (sp_alumno) presentdo en la seccion explorador de soluciones con la cual al pasr el parámetro de la bolet generara el PDF importabte resaltar que `int.Parse(label1.Text)` es el labeel donde se uestra la boleta y al ser una cadena lo pasa a entero como un parámetro para cargar los datos respecto a ese alumno

CONSULTA REGISTROS



FORMULRIO CONSULTA_REGISTROS	
<pre>registro.visualizar("SELECT Boleta, Nombre, Ape_Pat, Ape_Mat, ID_Materia from ALUMNO", "ALUMNO"); this.dataGridView1.DataSource = registro.ds.Tables["ALUMNO"]; this.dataGridView1.Refresh();</pre>	Aquí es importante resaltar que dentro del código de este formulario hay un evento por default llamado LOAD en el cual vamos a utilizar el método load del SGBD el cual cargara los datos de la sentencia de sql SELECT Boleta, Nombre, Ape_Pat, Ape_Mat, ID_Materia from ALUMNO de la base RECURSAMIENTO

OBSERVACIONES FINALES

Para finalizar este manual es importante resaltar que se está utilizando la base de datos RECURSAMIENTO y tomar en cuenta que las tabla ALUMNO únicamente ingresa llaves las cuales son recuperadas el momento en el recursador selecciona la materia en el comboBox del formulario REGISTRO_RECURSAMIENTO en el cual por medio del método de SGBD ObtenerIDAlumno realiza una sentencia para recuperar el ide de la materia y aparte de ese el ID_prof e ID_Grupo sino el registro no se realizará dado que estos funcionan como llaves foráneas.

Cualquier duda respecto a este manual comunicarse al correo wy_darkmagician@hotmail.com

Anexo Base de datos para ver las tablas utilizadas en el programa

BASE DE DATOS RECURSAMIENTO

```
CREATE TABLE REGISTRO(  
  UserBoleta NUMERIC(10),  
  Contraseña NVARCHAR(15),  
  CONSTRAINT REGISTRO_UserBoleta_pk PRIMARY KEY (UserBoleta))  
  
CREATE TABLE REGISTRO_ADMIN(  
  Usuario NVARCHAR(10),  
  Contraseña NVARCHAR(15),  
  CONSTRAINT REGISTRO_ADMIN_Usuario_pk PRIMARY KEY (Usuario))  
  
CREATE TABLE UNIDADES(  
  Materia VARCHAR(35),  
  Semestre VARCHAR(40),  
  CONSTRAINT MATERIA_Materia_pk PRIMARY KEY (Materia))  
  
CREATE TABLE MATERIA(  
  ID_Materia INT,  
  Descripcion VARCHAR(35),  
  CONSTRAINT MATERIA_ID_Materia_pk PRIMARY KEY (ID_Materia),  
  CONSTRAINT MATERIA_Descripcion_fk Foreign key (Descripcion) References  
  UNIDADES(Materia))  
  
CREATE TABLE GRUPO(  
  ID_Grupo INT,  
  ID_Materia INT,  
  Grupo NVARCHAR(5),  
  Salon NUMERIC(5),  
  Horario VARCHAR(10),  
  Ocupabilidad NUMERIC(2),  
  Fech_Inic DATE,  
  Fech_Fin DATE,  
  CONSTRAINT GRUPO_ID_Grupo_pk PRIMARY KEY (ID_Grupo),  
  CONSTRAINT GRUPO_ID_Materia_fk Foreign key (ID_Materia) References  
  MATERIA(ID_Materia))  
  
CREATE TABLE PROFESOR(  
  ID_Prof INT,  
  Nombre VARCHAR(50),  
  Ape_Pat VARCHAR(50),  
  Ape_Mat VARCHAR(50),  
  ID_Materia INT,  
  ID_Grupo INT,  
  CONSTRAINT PROFESOR_ID_Prof_pk PRIMARY KEY (ID_Prof),  
  CONSTRAINT PROFESOR_ID_Materia_fk Foreign key (ID_Materia) References  
  MATERIA(ID_Materia),  
  CONSTRAINT PROFESOR_ID_Grupo_fk Foreign key (ID_Grupo) References  
  GRUPO(ID_Grupo))  
  
CREATE TABLE ALUMNO(  
  ID_Alum INT,  
  Boleta NUMERIC(10),  
  Nombre VARCHAR(50),  
  Ape_Pat VARCHAR(50),  
  Ape_Mat VARCHAR(50),  
  Carrera VARCHAR(15),  
  Grupo_Actual NVARCHAR(5),  
  ID_Materia INT,  
  ID_Grupo INT,  
  ID_Prof INT,
```

```
CONSTRAINT ALUMNO_ID_Alum_pk PRIMARY KEY (ID_Alum),
CONSTRAINT ALUMNO_Boleta_fk Foreign key (Boleta) References
REGISTRO (UserBoleta),
CONSTRAINT ALUMNO_ID_Materia_fk Foreign key (ID_Materia) References
MATERIA (ID_Materia),
CONSTRAINT ALUMNO_ID_Grupo_fk Foreign key (ID_Grupo) References
GRUPO (ID_Grupo),
CONSTRAINT ALUMNO_ID_Prof_fk Foreign key (ID_Prof) References
PROFESOR (ID_Prof))
```