

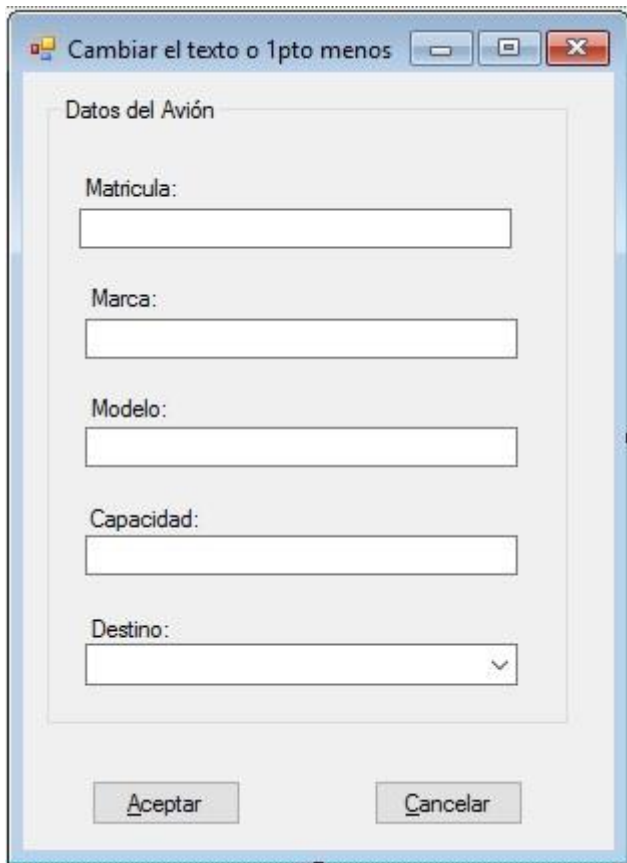
Segundo Parcial Laboratorio de Computación III –Aeropuerto--

- 1- Formularios y Clases:** Realizar una solución llamada con su apellido punto (.) su nombre. Ej: Brancatelli.Diego
A esta solución agregarle un proyecto de tipo **Biblioteca de Clases** llamada **Entidades** en la cual se deberá crear las siguientes clases con sus respectivos constructores:

Avion Class	Vuelo Class
Atributos _matricula int private _marca String private _modelo String private _capacidad int private _codVuelo int private	Atributos _codVuelo int private _destino String private
Propiedades Matricula Int Marca String Modelo String Capacidad Int CodVuelo Int	Propiedades CodVuelo Int Destino String

Además se deberá agregar en la solución un proyecto de tipo **WindowsForm** llamado **Aeropuerto.SegundoParcial.LabIII** con los siguientes formularios:

frmAvion

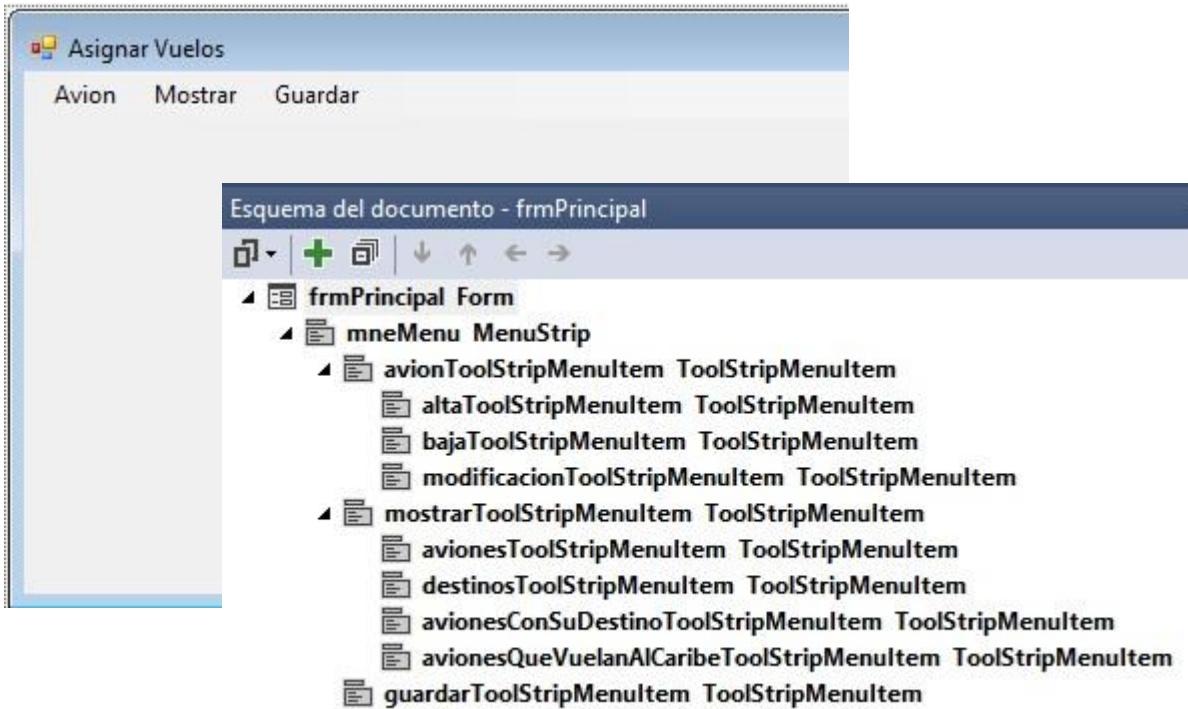


Este formulario nos permitirá realizar las tareas de alta, baja y modificación de los aviones.

En el manejador de evento click del botón **Aceptar**, se creará una instancia de tipo **Avion** que se expondrá en la **propiedad de sólo lectura** de este formulario. Además se colocará en la propiedad **DialogResult** del formulario el valor DialogResult.OK.

En el manejador de evento click del botón **Cancelar** se colocará en la propiedad **DialogResult** del formulario el valor DialogResult.Cancel.

frmPrincipal



frmPrincipal (Atributos)

Dataset _dataSetAviones_Vuelos;
XXXDataAdapter _dataAdapterAviones;

frmPrincipal (Métodos)

CrearDataTableVuelos() As DataTable: Será invocado sólo si el archivo “VuelosDatos.XML” y “VuelosEsquema.XML” NO existen, este método creará el DataTable **dtVuelos** con los siguientes campos: **Código** (entero) y **Destino** (cadena)

Código será auto-incremental, comenzando de 1000 y aumentando de 5 en 5.

Cargar en el DataTable al menos **3 vuelos** con distintos destinos, uno debe ser el Caribe y luego guardar el esquema y los datos en dos archivos distintos (**VuelosDatos.XML** y **VuelosEsquema.XML**)

ConfigurarDataAdapter(): Este método configurará el DataAdapter con su conexión, comandos y parámetros para interactuar con la tabla “**Aviones**” de la base de datos.

Nota: Se pide que la instrucción T-SQL del SelectCommand retorne: Matricula, Marca, Modelo, Capacidad y codigoVuelo (utilizar alias).

TraerDatos(): Carga los datos desde la base de datos en el DataSet (en el DataTable **dtAviones** con un **dataReader**) y los datos del DataTable **dtCursos** desde los archivos .XML o desde el método **CrearDataTableCVuelos()**, según corresponda.

Nota: Se pide que la instrucción T-SQL del Command retorne Matricula, Marca, Modelo, Capacidad y codigoVuelo (utilizar alias).

EstablecerRelacion(): Este método relacionará los DataTables **Aviones** y **Vuelos** que se encuentran dentro del DataSet.

2- Alta de Aviones:

Se instanciará un objeto de tipo frmAvion (**frmAltaAvion**) y se mostrará en forma Modal. Con los atributos que posee el objeto de tipo **Entidades.Avion** en este formulario, se creará una nueva fila del DataTable **dtAvion** que se agregará a la colección **Rows** del mismo.

3- Baja de Avion:

Se pedirá la **matricula** (a través de un **InputBox**) y se buscará en el DataTable el avión con ese número de matrícula. Si se encuentra, se creará un objeto de tipo **Entidades.Avi3n** que se le pasará como parámetro al constructor de **frmAvion**, cuya instancia se llamará **frmBajaAvion**.

En dicha instancia se mostrarán los datos del avión y si se elige Aceptar, el elemento se marcará como borrado. Si no se encuentra el número de Id se deberá informar de lo acontecido.

4- Modificación de Avion:

Se pedirá la **matricula** (a través de un **InputBox**) y se buscará en el DataTable el avión con ese número de Matrícula. Si se encuentra, se creará un objeto de tipo **Entidades.Avion** que se le pasará como parámetro al constructor de **frmAvion**, cuya instancia se llamará **frmModificarAvion**.

En dicha instancia se mostrarán los datos del avión y se permitirán modificar todos a excepción de la matrícula. Si se elige **Aceptar**, se modificarán los datos en la fila que corresponda del DataTable **dtAviones**. Si no se encuentra la matrícula se deberá informar de lo acontecido.

5- Mostrar Aviones y Vuelos:

Al elegir esta opción del menú, se creará una instancia del formulario **frmMostrar** (que usted deberá crear) para mostrar los datos del DataTable que corresponda.

6- Mostrar los datos de los Vuelos junto con la marca y modelo de los aviones que lo realizan (frmMostrar)

Utilizar: **GetParentRow** o **GetChildRows**, según corresponda.

7- Mostar los datos de los Aviones junto con el nombre del Destino, filtrados para los que viajan al Caribe (FrmMostrar). Utilizar: Select y GetParentRow o GetChildRows, según corresponda.