

Profesores:

Neiner Maximiliano  
Scarafilo Germán  
Baus Christian  
Oggioni Ezequiel

**Nota:**

*Esta guía forma parte del trabajo práctico número uno (TP Nro. 1), que será entregado en dos partes (una antes del primer parcial y la otra antes del segundo parcial), las fechas de entrega serán publicadas por el profesor de Laboratorio III a su debido tiempo.*

1. Esta guía será entregada en un CD con una carpeta nombrada con su nombre punto (.) su apellido punto (.) su división. El siguiente ejemplo corresponde a un alumno de 3 C llamado Juan Pérez, la carpeta dentro del CD quedará:

**Juan.Perez.3C.**

Dentro de esta carpeta estarán las soluciones/proyectos correspondientes a cada ejercicio de la guía. Las soluciones/proyectos se nombrarán de la siguiente forma: Aplicación punto (.) ##, Dónde ## será el número del ejercicio. El siguiente ejemplo corresponde al nombre de la solución/proyecto del ejercicio número 3:

**Aplicación.03.**

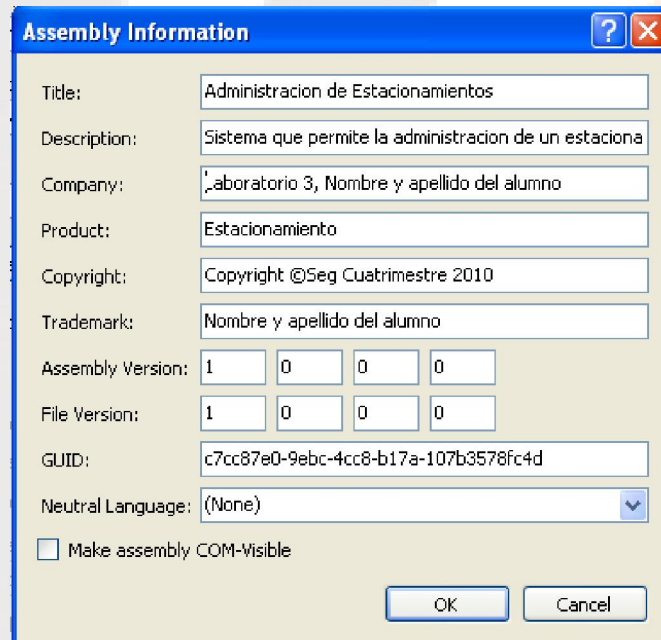
2. Se recomienda utilizar los prefijos para cada elemento dentro de las aplicaciones de tipo Windows Form como las convenciones al nombrar clases, métodos, atributos, etc., ya que se tendrán en cuenta al momento de evaluar este trabajo práctico.
3. Todas las clases intervinientes en los ejercicios de tipo WindowsForm, deben ser creadas en proyectos de tipo Biblioteca de clases.
4. El punto de entrada de todas nuestras aplicaciones debe ser un "Sub Main", que se encontrará en una clase por defecto (Program). Ej.:

```
try{
    FrmPrincipal frmInicio = new FrmPrincipal();
    Application.Run(frmInicio);
}
catch(Exception ex){
    MessageBox.Show("Error: " + ex.Message, "Error desconocido");
}
```

5. Es obligatorio el uso del bloque `try catch finally` en todas las instrucciones que puedan generar excepciones.

## Estacionamiento:

Realizaremos una aplicación que permita administrar un estacionamiento, para esto comenzaremos con la configuración de nuestro ensamblado.



## Aplicación N°9 Menú Principal (Estacionamiento)

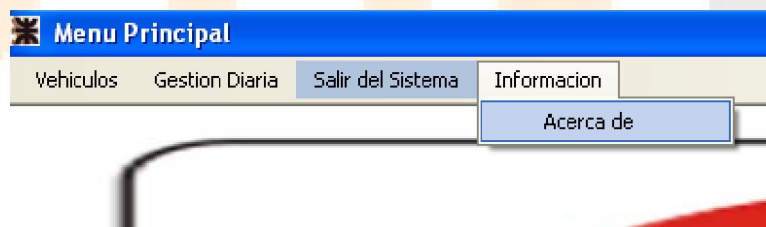
**#Recursos**

**#menú**

**#Formulario MDI**

**#Formulario Owner**

Crear un formulario que utilizaremos como pantalla principal de nuestra aplicación, este formulario será de tipo "MDIContainer", esto será configurado en el **evento Load** del formulario. Este formulario tendrá una lista de objetos de tipo vehículos como atributo propio con sus propiedades

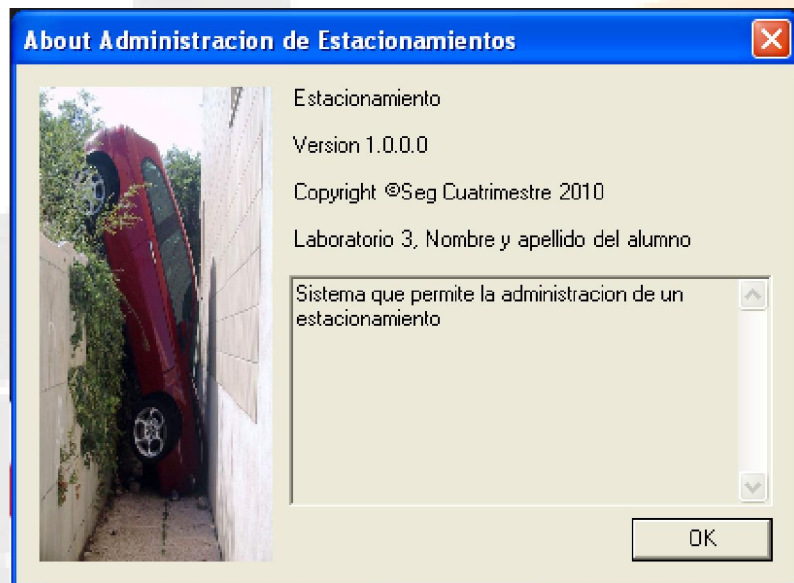


Este formulario tendrá un icono y una imagen de fondo (cargarlos en tiempo de ejecución), estos dos elementos serán cargados a los recursos del sistema antes de utilizarlos.

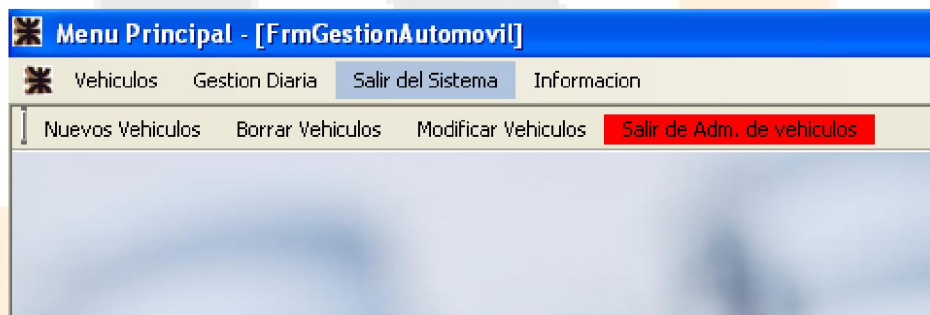
Este formulario tendrá un menú con botones:

# El botón "Acerca de" instanciará un objeto de tipo "AboutBox", cuyo propietario sea el formulario principal (hacerlo en tiempo de ejecución).

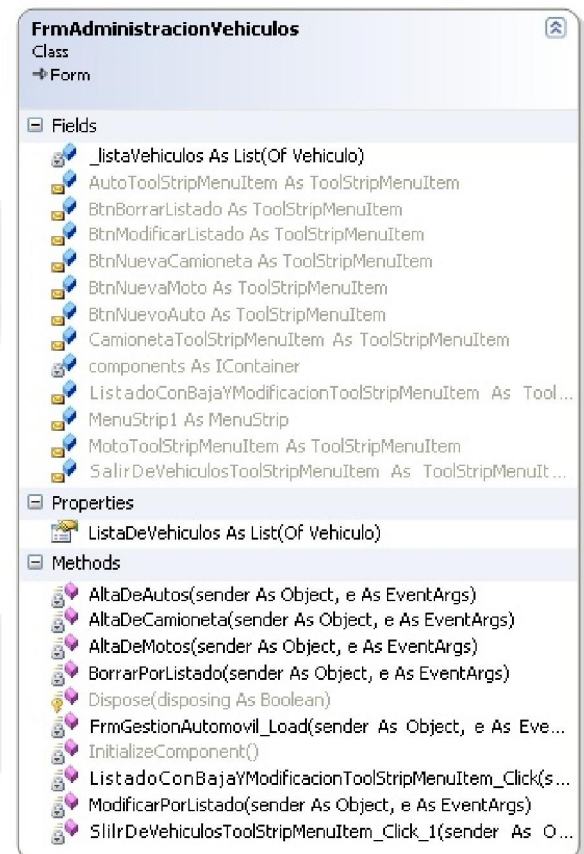
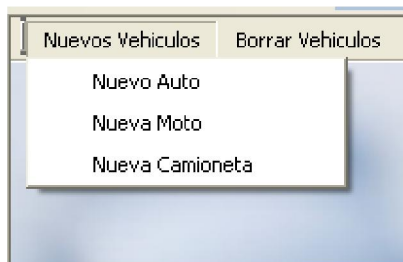
Este formulario muestra la información que configuramos anteriormente de nuestro ensamblado.



# El botón "Vehículos" instanciará un objeto de tipo "FrmGestionAutomovil", Este formulario estará contenido en el formulario principal y tendrá su propio menú como muestra la siguiente imagen.



**FrmGestionAutomovil** tiene el botón del menú "Nuevos vehículos" que nos permitirá acceder a los formularios para hacer las altas de los distintos tipos de vehículos. Este formulario (**FrmGestionAutomovil**) tendrá una lista de objetos de tipo vehículos como atributo propio con sus propiedades



#El botón "Salir de Adm. De vehículos" cerrara el formulario **FrmGestionAutomovil** sin cerrar la aplicación

## **Aplicación N°10 herencia de formularios (Estacionamiento)**

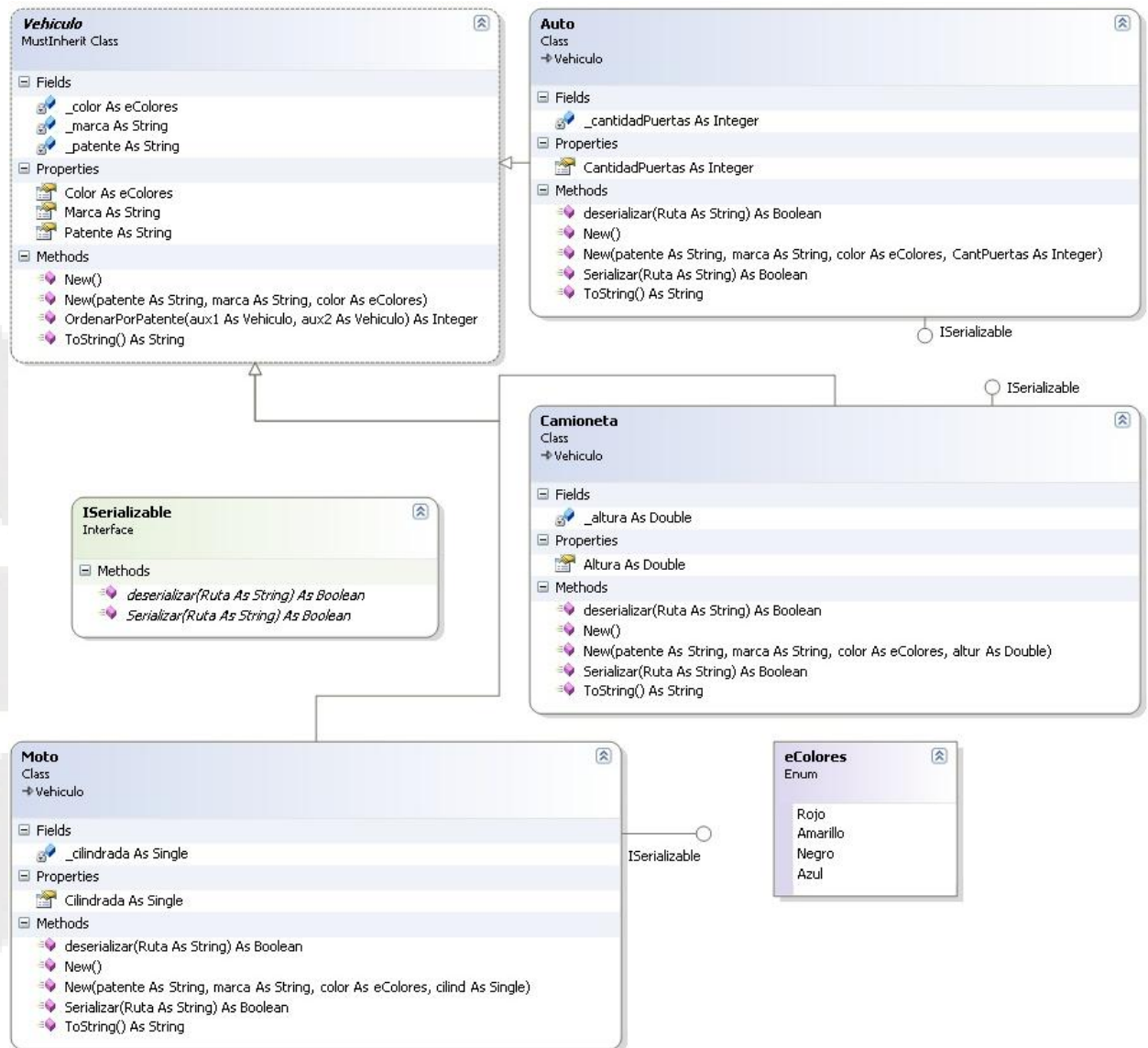
### **#Formulario de dialogo**

### **# Herencia Visual**

Nuestra aplicación deberá crear objetos de las clases Auto, Moto y Camioneta. Estas clases heredan de la clase Vehículo.

Así como tenemos la herencia entre clases, también realizaremos la herencia visual entre los formularios.

El siguientes es el diagrama de las clases que serán creadas en una **Biblioteca de clases** que será un **proyecto separado** al de WindowsForm.



El formulario base será el de vehículo, contendrá:

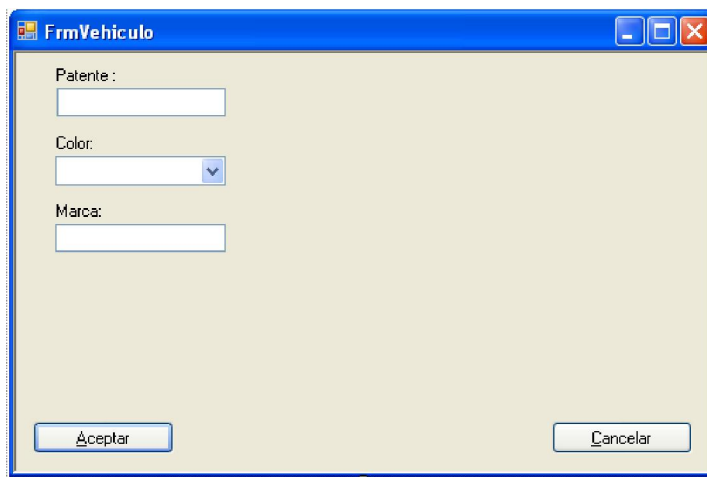
# un ComboBox que se cargara con todos los valores del enumerado eColores en un **método** que se llamará en el evento Load del formulario.

# Dos TextBox

#un botón Cancelar, su manejador asociado será virtual y será el "**DialogResult.Cancel**" del formulario.

#un botón Aceptar, su manejador asociado será virtual y será el "**DialogResult.Ok**" del formulario





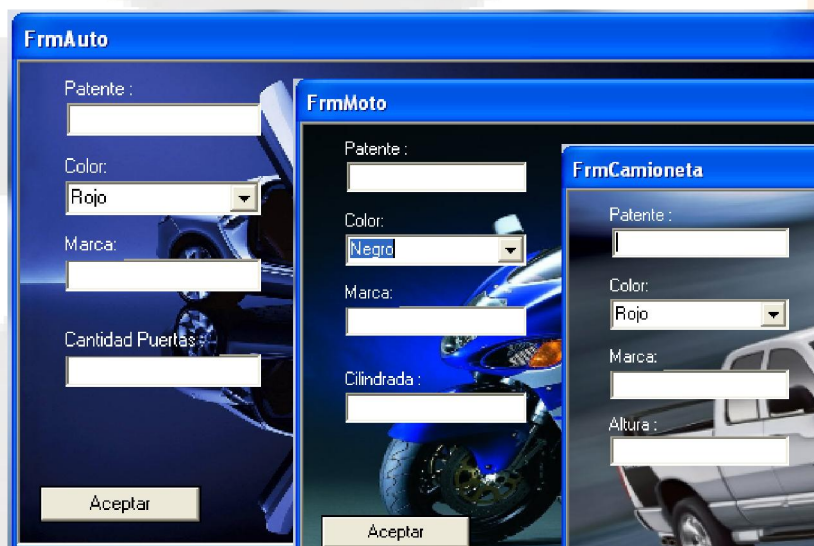
**FrmVehiculo**

Patente :

Color:

Marca:

De este formulario base heredaran los siguientes tres formularios:



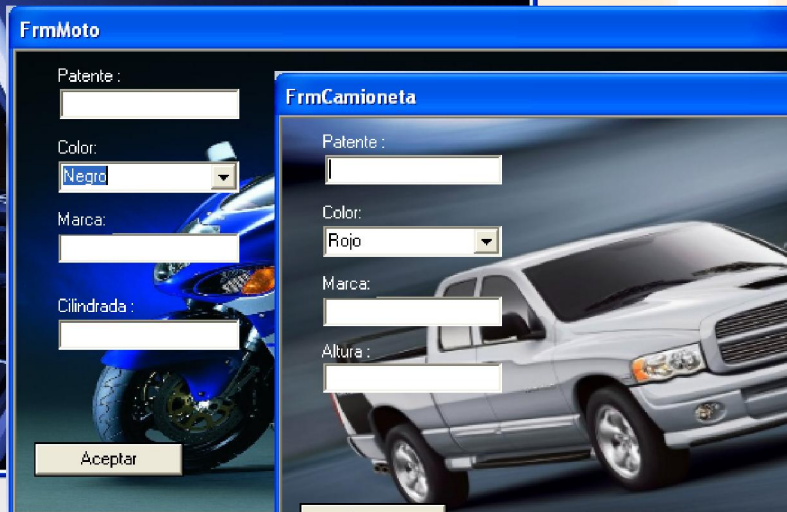
**FrmAuto**

Patente :

Color:

Marca:

Cantidad Puertas:



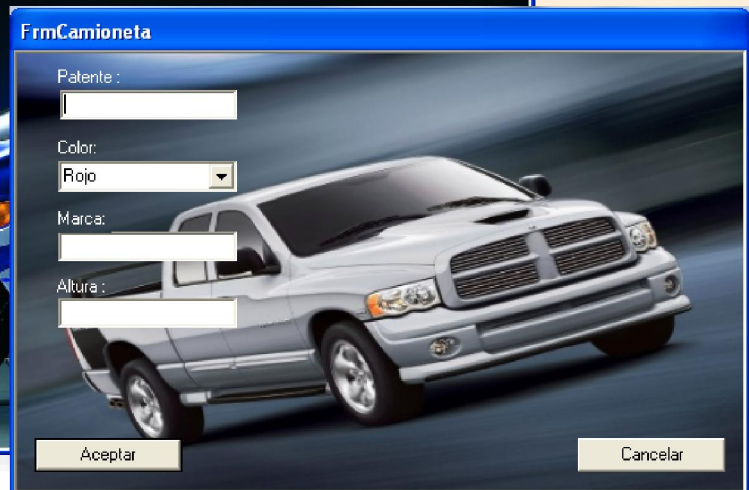
**FrmMoto**

Patente :

Color:

Marca:

Cilindrada :



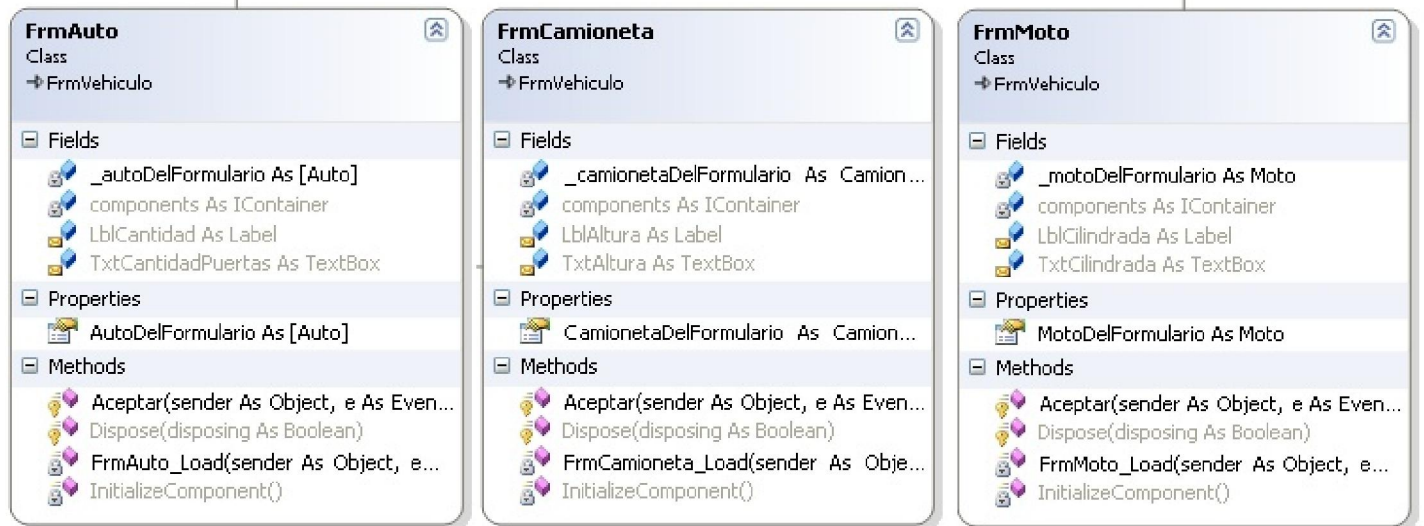
**FrmCamioneta**

Patente :

Color:

Marca:

Altura :



#Cada formulario tendrá un objeto como atributo, con sus propiedades de lectura escritura.  
 FrmCamioneta (objeto camioneta)  
 FrmAuto (objeto Auto)  
 FrmMoto (objeto Moto)

#Se deberá sobrescribir el manejador asociado al botón aceptar, en este método instanciará el objeto que tiene como atributo, pasándole los datos tomados de los controles a su constructor. Y luego indicar que este método y será el **"DialogResult.Ok"** del formulario.

Desde el **FrmGestionAutomovil** el método que muestra el frmAlta para auto será de la siguiente manera:

```
Private Sub AltaDeAutos(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnNuevoAuto.Click
    Dim frmAlta As New FrmAuto
    If frmAlta.ShowDialog() = Windows.Forms.DialogResult.OK Then
        Me.ListaDeVehiculos.Add(frmAlta.AutoDelFormulario)
    End If
End Sub
```

Si el formulario mostrado retorna un **"DialogResult.OK"**, tomaremos el objeto que tiene ese formulario como atributo y lo agregaremos a la lista de vehiculos que el **FrmGestionAutomovil** tiene como atributo. Este mismo procedimiento se deberá realizar para el alta de Moto y Camioneta.