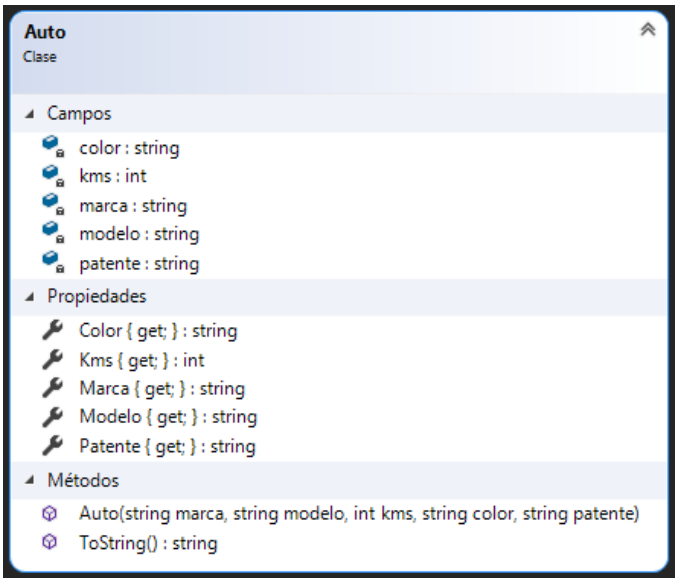
FINAL – LABORATORIO 2 – 2021

IMPORTANTE:

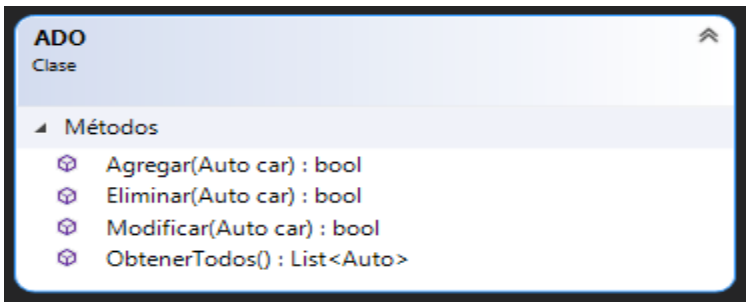
- **Dos (2) errores en el mismo tema anulan su puntaje.**
- La correcta documentación y reglas de estilo de la cátedra serán evaluadas.
- Colocar sus datos personales en el nombre de la carpeta principal y en la solución: ***Apellido.Nombre***. Ej: Pérez.Juan. No sé corregirán proyectos que no sea identificable su autor.
- Los exámenes que no compilen, no aprueban.
- **Reutilizar** tanto código como sea posible.
- Tanto en métodos, atributos y propiedades, colocar nombre de la clase (en estáticos), **this** o **base**.
- Aplicar los principios de los 4 pilares de la POO.
- Tener en cuenta el control de excepciones.
- La entrega será en un archivo comprimido, el cual debe contar con Apellido y Nombre, al igual que la solución. Se entregará al finalizar, mediante un Google Form.
- La duración del final es de 120 minutos.
- Partir de la solución dada.

1. Agregar un proyecto de tipo Bibliotecas de Clases. Nombrarlo **Entidades**. Agregar la clase **Auto**.



Todos los atributos son protegidos.
Cada atributo posee una propiedad pública de sólo lectura.
Sobrescribir el método ToString, para que retorne, en un único renglón, las características del auto, respetando el siguiente formato: "Marca: Torino – Modelo: Coupé – Kms: 150000 – Color: Marrón – Patente: ABC987", siendo *Torino*, *Coupé*, *15000*, *Marrón* y *ABC987*, los valores de una instancia de tipo Auto.

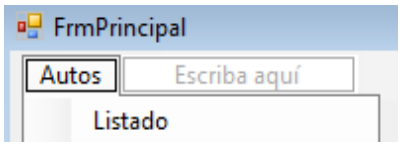
2. Agregar la clase **ADO**, en Entidades. Esta clase será la encargada de realizar un CRUD sobre la base de datos ***concesionaria*** (*).



El método ***ObtenerTodos*** es un método de clase y retorna una lista genérica de tipo Auto, representando a cada uno de los registros de la tabla ***autos*** (**).

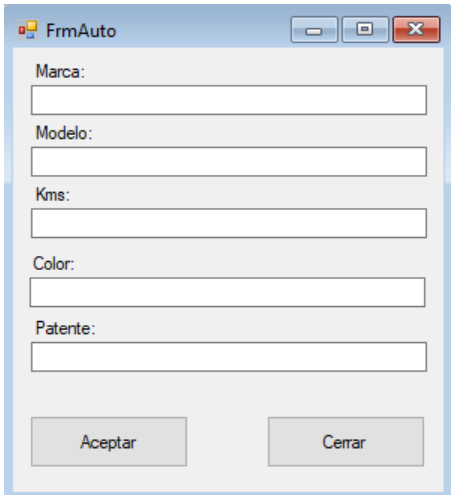
Los métodos Agregar, Eliminar y Modificar son métodos de instancia y retornan un booleano que indica si se pudo o no agregar, eliminar o modificar la base de datos.

Nota: tanto la modificación como la eliminación se harán por patente.



3. FrmPrincipal
   - **Menú -> Autos -> Listado**:
   - Se accede a FrmListado.
   - Contiene un ***DataGridView*** que estará enlazado con una lista genérica de tipo Auto (se carga desde la clase ADO).
   - Posee los botones: Agregar, Modificar y Eliminar.

4. FrmListado -> **Agregar**:
   a. Agrega un nuevo auto a la base de datos utilizando FrmAuto.
   b. FrmAuto expone en una propiedad (pública y de sólo lectura) de tipo Auto (*AutoDelFormulario*).
   c. Interactuar con la clase ADO.
   d. Refrescar listado.

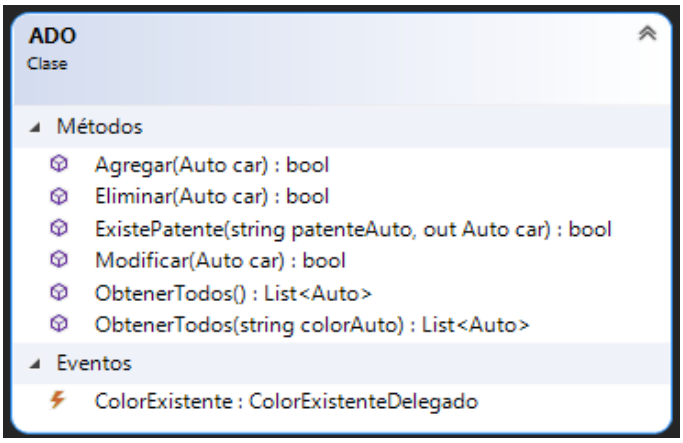


5. FrmListado -> **Modficar**:
   a. Se muestra en FrmAuto TODOS los valores del objeto seleccionado del *DataGridView*.
   b. El valor de la patente no se debe poder modificar (adecuar FrmAuto).
   c. Modifica al auto seleccionado. Interactuar con la clase ADO.
   d. Refrescar listado.

6. FrmListado -> **Eliminar**:
   a. Se muestra en FrmAuto todos los valores del objeto seleccionado del DataGridView.
   b. Elimina el auto seleccionado. Interactuar con la clase ADO.
   c. Refrescar listado.

---

7. Modificar la clase ADO.
   a. Diseñar el delegado *ColorExistenteDelegado*, de acuerdo a las convenciones vistas, para el evento ***ColorExistente***.
   b. Agregar una sobrecarga al método **ObtenerTodos** que permita retornar una lista genérica de tipo Auto, con los autos cuyos colores coincidan con el parámetro recibibo (obtener por color).
   c. Agregar el método de clase **ExistePatente**. Si la patente recibida cómo parámetro existe en la base de datos, se retornará ***true*** y toda la información del auto (en la base de datos) en el parámetro de salida de tipo Auto. Caso contrario, se retornará ***false***.

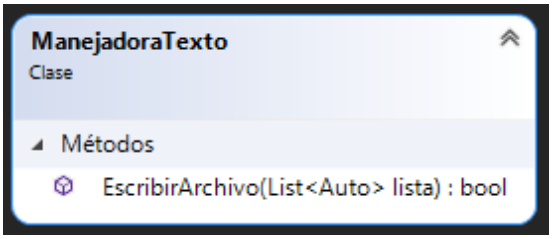


8. Método **Agregar**: Modificar la lógica de este método para que:
    a. Si la patente del auto a ser agregado ya existe en la base de datos, se lance la excepción ***PatenteExisteException***, que recibirá en su construtor la instancia de tipo Auto que está en la base. NO se deberá agregar el nuevo auto.

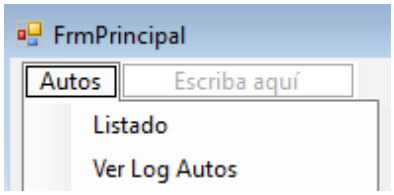
    Capturar la excepción en FrmListado y mostrar (en un MessageBox) la información del auto previamente guardado en la base de datos.

    b. Si el color del auto que se agregó ya existe en la base de datos, se disparará el evento ***ColorExistente***, que recibirá cómo uno de sus parámetros la lista genérica de tipo Auto cuyos colores coincidan con el auto agregado.

    Crear, en FrmListado, el manejador necesario para que una vez capturado el evento, se guarde en un archivo de texto:
    La fecha (con hora, minutos y segundos) y en un nuevo renglón, el color y todos los datos de todos los autos para ese color. Se deben acumular los mensajes.
    Formato:
    09/12/2018 09:12:18
    Color Marrón:
    Marca: Torino – Modelo: Coupé – Kms: 150000 – Color: Marrón – Patente: ABC987
    Marca: Seat – Modelo: PM – Kms: 0 – Color: Marrón – Patente: ABC888
    El archivo se guardará con el nombre 'autos.log' en la carpeta 'Mis documentos' del cliente.
    El manejador de eventos (Manejador_colorExistente) invocará al método (de clase) ***EscribirArchivo****(List<Auto>) (se alojará en la clase* ***ManejadoraTexto*** *en Entidades), que retorna un booleano indicando si se pudo escribir o no.*



9. FrmPrincipal
    a. **Menú -> Autos -> Ver Log Autos**:
    b. Configurar el ***OpenFileDialog*** para poder seleccionar el log de autos.
        ///Título -> 'Abrir archivo de autos'
        ///Directorio por defecto -> Mis documentos
        ///Tipo de archivo (filtro) -> .log
        ///Extensión por defecto -> .log
        ///Nombre de archivo (defecto) -> autos
    c. El contenido del archivo recuperado se mostrará en ***txtAutosLog***.



10. FrmPrincipal **-> Load**
    a. Iniciar un hilo secundario que permita visualizar en ***lstAutos*** los datos de todos los autos que estén en la base de datos.
    El ciclo será continuo y se detendrá cuando se dispare el evento *FormClosed* de FrmPrincipal.
    El listado de autos se refrecará cada 5 segundos.
    b. Desarrollar un método que invoque al método *ADO.ObtenerTodos* y luego mostrar el contenido en el control especificado.