

Universidad Tecnológica Nacional

Facultad Regional Avellaneda



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

Materia: Laboratorio de Programación II

Apellido:		Fecha:	26-10-2023
Nombre:		Docente ⁽²⁾ :	
División:	2°	Nota ⁽²⁾ :	
Legajo:		Firma ⁽²⁾ :	
Instancia ⁽¹⁾ :	PP	RPP	X
		SP	
		RSP	
		FIN	

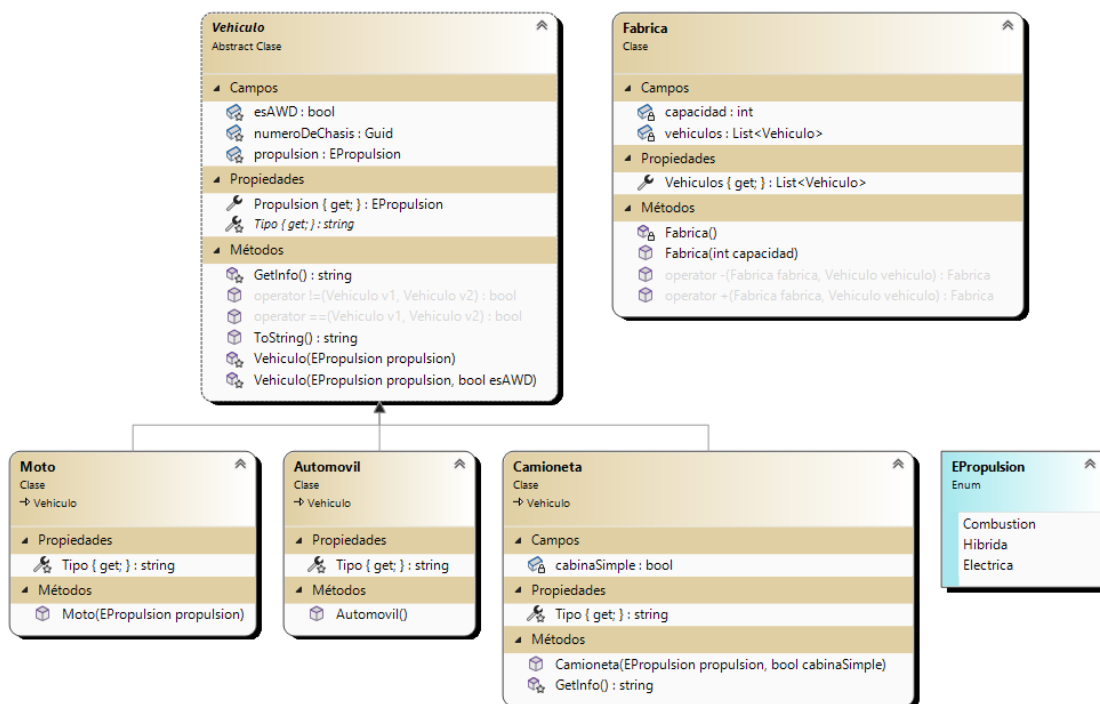
(1) Las instancias validas son: 1^{er} Parcial (PP), Recuperatorio 1^{er} Parcial (RPP), 2^{do} Parcial (SP), Recuperatorio 2^{do} Parcial (RSP), Final (FIN). Marque con una cruz. (2) Campos a ser completados por el docente.

IMPORTANTE:

- **2 (dos) errores en el mismo tema anulan su puntaje.**
- La correcta documentación y reglas de estilo de la cátedra serán evaluadas.
- El proyecto debe ser creado en .Net 5.
- Colocar sus datos personales en el nombre de la carpeta principal y la solución: Apellido.Nombre.Div. Ej: Pérez.Juan.2D. No sé corregirán proyectos que no sea identificable su autor.
- No se corregirán exámenes que no compilen.
- **Reutilizar** tanto código como crean necesario.
- Colocar nombre de la clase (en estáticos), **this** o **base** en todos los casos que corresponda.
- Aplicar los principios de los 4 pilares de la POO.

Vamos a crear la fábrica de vehículos:

1. Para comenzar, crear un proyecto del tipo Biblioteca de Clases y reproducir el siguiente esquema:



Vehiculo

2. La clase será abstracta.
3. Por defecto NO serán AWD (4x4). Reutilizar los constructores de forma que se logre esto.
4. El numero de chasis se generará al construir un vehículo. Utilizar el método estático Guid.NewGuid().
5. La propiedad Propulsión será de sólo lectura, y Tipo se deberá implementar de forma obligatoria en las clases derivadas.
6. GetInfo retornará la información del vehiculo con el siguiente formato: "[Tipo] con propulsión a [Propulsion], [SI/NO] es AWD, numero de chasis [numeroDeChasis]." Utilizar el método Format de String para lograrlo.
7. Operator == retornará TRUE si ambos objetos son del mismo tipo (TYPE) y a su vez poseen el mismo número de chasis.
8. ToString expondrá la informacion del vehículo.

Camioneta

9. Todas las camionetas serán AWD (4x4). Reutilizar constructores para lograrlo.
10. GetInfo informará todos sus datos y agregará si la camioneta es cabina simple o doble: " (todos los datos de la camioneta), con cabina: [Simple (true) o Doble (false)]". Utilizar StringBuilder para lograrlo.
11. La propiedad tipo retornará "Camioneta".

Automovil

12. Todos los automóviles serán de propulsión hibrida. Reutilizar constructores para lograrlo.
13. La propiedad tipo retornará "Automovil".

Moto

14. La propiedad tipo retornará "Moto".

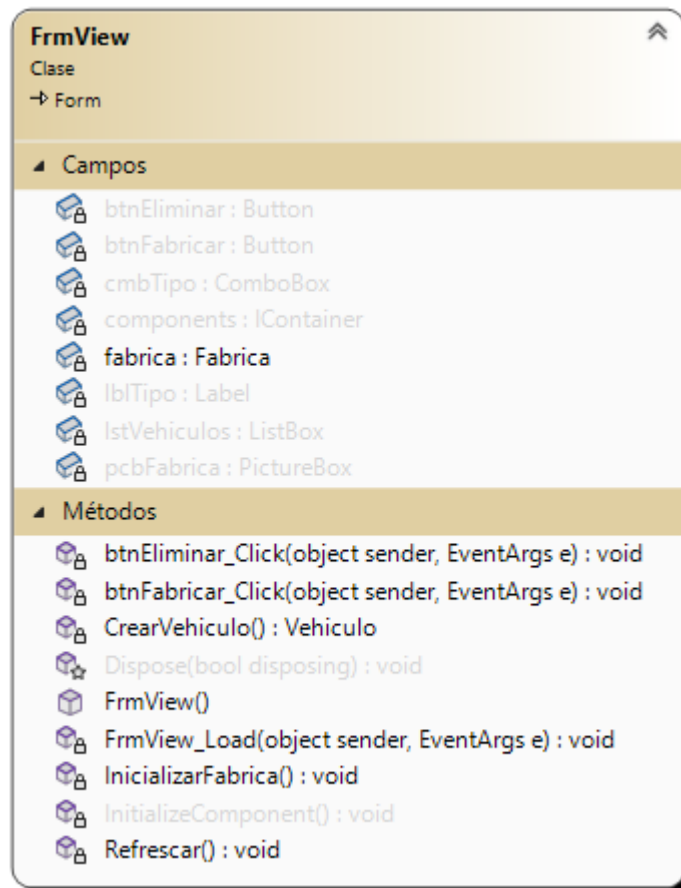
Fábrica

15. Propiedad Vehiculos retornará la lista de vehiculos. Sera de solo lectura.
16. La lista de vehiculos sólo podrá ser instanciada en el constructor privado.
17. El operador + deberá ser capaz de agregar un vehículo a la lista.
 - a. La capacidad se utilizará para dar un límite a la cantidad de vehiculos.
 - b. Si hay lugar, se agregará al nuevo vehiculo y se retornará la fábrica modificada.
18. El operador – deberá ser capaz de sacar a un vehiculo de la lista.
 - a. Se buscará el primer vehiculo del mismo tipo (TYPE) y chasis para ser removido.
 - b. Sólo se quitará la primera aparición de un vehiculo que coincida.
 - c. Se retornará la fábrica modificada.

Formulario

19. Crear en un proyecto aparte el siguiente formulario:

20. El ícono del formulario será favIcon y el PictureBox se utilizará para icons8-vehicles-64.png, ambos archivos presentes en la información entregada junto a este pdf.
21. El ComboBox tendrá el DropDownStyle como DropDownList. Cargar ítems (“Automóvil”, “Camioneta”, “Moto”) en el manejador del evento Form Load.
22. El formulario deberá iniciar centrado en la pantalla, teniendo ancho suficiente como para que se lean los textos. Habilitar la opción “HorizontalScrollBar” en True del lstVehiculos.
23. Respetar título y diseño, así como los nombres de sus atributos como muestra el siguiente diagrama de funcionamiento mínimo requerido:



24. El constructor del formulario, luego de inicializar los componentes, inicializara la fábrica y refrescara la lista de vehículos
25. El atributo `fabrica` deberá ser instanciado en el método `InicializarFabrica`, donde también se deberá agregar a la fábrica una camioneta de propulsión eléctrica y de cabina simple.
26. El método `Refrescar` deberá realizar las siguientes acciones en dicho orden:
 - a. Asignara al `DataSource` del `lstVehiculos` un `null`.
 - b. Asignara al `DataSource` del `lstVehiculos` la lista de vehículos de la fábrica.
27. El método `CrearVehiculo` devolverá una instancia de un nuevo vehículo según sea el tipo seleccionado en el `ComboBox`.
28. El botón `fabricar` realizara:
 - a. Creará un nuevo vehículo.
 - b. Lo agregara a la lista de la fábrica mediante el operador `+` (mas).
 - c. Refrescara la lista
29. El botón `eliminar` deberá:
 - a. Eliminar el vehículo seleccionado de la lista mediante el operador `-` (menos).
 - b. Refrescara la lista.

Si aun tenés tiempo, agregá las opciones para poder crear vehículos según su propulsión, según corresponda.