

Universidad Tecnológica Nacional
Facultad Regional Avellaneda



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

Materia: Laboratorio de Programación II

Apellido:		Fecha:	12/12/2019
Nombre:		Docente ⁽²⁾ :	
División:	2º	Nota ⁽²⁾ :	
Legajo:		Firma ⁽²⁾ :	
Instancia ⁽¹⁾ :	<div style="display: flex; justify-content: space-around;"> PP RPP X SP RSP FIN </div>		

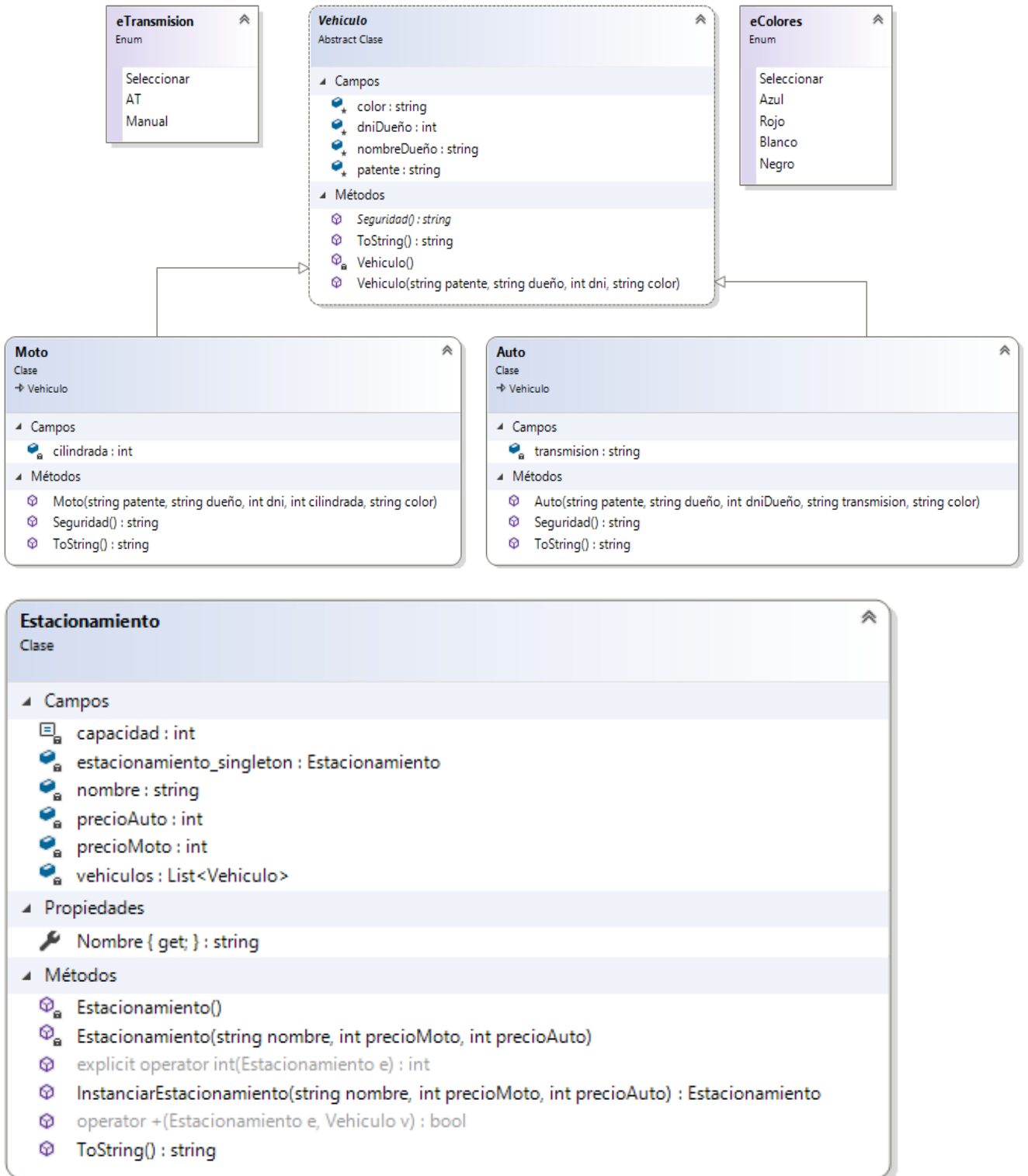
(1) Las instancias validas son: 1º Parcial (**PP**), Recuperatorio 1º Parcial (**RPP**), 2º Parcial (**SP**), Recuperatorio 2º Parcial (**RSP**), Final (**FIN**). Marque con una cruz.

(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.**
- Colocar sus datos personales en el nombre del proyecto principal, colocando:
Apellido.Nombre.Departamento. Ej: Pérez.Juan.2D. No se corregirán proyectos que no sea identificable su autor.
- **TODAS** las clases deberán ir en una Biblioteca de Clases llamada Entidades.
- 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.
- Colocar el nombre del docente con el cual cursó.

TIEMPO MÁXIMO PARA RESOLVER EL EXAMEN 90 MINUTOS.



1. Crear un proyecto del tipo Biblioteca de Clases y colocar el siguiente esquema de clases:

1. Clase **Vehiculo**:

- Tendrá un constructor por defecto privado en el cual los atributos de la clase serán inicializados con valores por defecto (utilizar string.Empty para los strings). El otro constructor será publico y deberá llamar al constructor privado para setear valores por defecto.
- Tendrá un método abstracto Seguridad que deberá ser implementado en las clases hijas.
- Sobrecargar el método ToString para que retorne un string que contenga la siguiente información: Nombre del cliente, DNI del cliente, tipo de vehículo (auto/moto), color y patente. Utilizar StringBuilder.

2. Clase **Auto**:

- Tendrá un constructor público que cargará el atributo transmisión. Este constructor deberá llamar al constructor de Vehículo para cargar los restantes.
- Sobreescibirá el método ToString llamando al mismo de la clase Vehiculo,agregando:
 - Tipo de transmisión
 - Llamará al método Seguridad declarado en esta misma clase. Utilizar StringBuilder.
- Método Seguridad. Implementará el método Seguridad el cual retornará un string con el siguiente mensaje: *"Estimado cliente: Recuerde utilizar el cinturón de seguridad, encender las luces bajas al entrar y salir del garage"*.
En caso de tener transmisión Manual, deberá agregar la siguiente línea: *Dentro del garage, solo puede circular en primera marcha*.

4. Clase **Moto**:

- Tendrá un constructor público que cargará el atributo Cilindrada. Este constructor deberá llamar al constructor de Vehículo para cargar los restantes.
- Sobrescribirá el método ToString invocando al mismo de la clase Vehículo, agregando:
 - Cilindrada
 - Llamará al método Seguridad declarado en esta misma clase. Utilizar StringBuilder.
- Método Seguridad. Implementará el método Seguridad el cual retornará un string con el siguiente mensaje: "Estimado cliente: Recuerde utilizar casco y mantener las luces encendidas al circular por el garage"

5. Clase **Estacionamiento**:

- Tendrá una variable de tipo constante con valor 5.
- Tendrá un atributo estático de tipo Estacionamiento llamado estacionamiento_singleton.
- Tendrá un constructor privado por defecto donde se instanciará la lista de Vehículos.
- Tendrá un segundo constructor privado el cual recibirá como parámetros el nombre del estacionamiento, precio de la moto por hora, precio del auto por hora. Llamará al constructor privado para instanciar la lista.
- Método estático InstanciarEstacionamiento: Será público, estático, de tipo Estacionamiento que recibirá por parámetro: nombre del estacionamiento, precio de la moto por hora, precio del auto por hora.

En caso de que el atributo estacionamiento_singleton sea nulo, este será instanciado utilizando el constructor correspondiente para setearle el nombre, precio moto y precio auto y deberá retornarlo

En caso de que estacionamiento_singleton no sea nulo (es decir, ya haya sido instanciado alguna vez), deberá simplemente retornarlo.

- Conversión explícita: Deberá convertirse explícitamente un estacionamiento en un valor entero que representará la cantidad de autos actuales del estacionamiento.
- Sobrecarga operador +: Deberá recibir un estacionamiento y un vehículo. Deberá agregar el vehículo al estacionamiento siempre y cuando haya capacidad libre. Retornará true si pudo agregarlo, o false si no pudo. Utilizar la conversión explícita solicitada en el punto anterior para obtener la cantidad de vehículos actuales del estacionamiento.
- Sobreescribir el método ToString para mostrar la información del estacionamiento. A su vez deberá recorrer la lista de vehículos y mostrar la información de ellos. Utilizar String Builder.

2. Crear los siguientes formularios:

Clase FrmAlta:

The image shows a Windows-style dialog box titled "Alta". It contains two main sections: "Datos dueño" and "Datos vehiculo". The "Datos dueño" section has two text input fields labeled "Nombre" and "DNI". The "Datos vehiculo" section has several controls: a text input for "Patente Vehiculo", a dropdown menu for "Color", radio buttons for "Tipo vehiculo" (with "Auto" and "Moto" options), a dropdown menu for "Transmision", and a text input for "Cilindrada (CC)". At the bottom of the dialog are two buttons: "Aceptar" and "Cancelar".

- Tendrá un atributo estático de tipo FrmAlta llamado frmAlta_instancia.
- Método estático CrearInstancia: Será público, estático, de tipo FrmAlta que recibirá por parámetro un estacionamiento.
- En caso de que el atributo frmAlta_Instance sea nulo, este será instanciado utilizando el constructor correspondiente y retornará la instancia.
- En caso de que frmAlta_Instance no sea nulo (es decir, ya haya sido instanciado alguna vez), deberá simplemente retornarlo.
- En el método Load
 - Deberá llenar el combobox de los colores con los enumerados asignándole a la

propiedad DataSource del combobox el siguiente código:

```
Enum.GetValues(typeof(eColores));
```

- deberá llenar el combobox de las transmisiones con los enumerados asignándole a la propiedad DataSource del combobox el siguiente código:

```
Enum.GetValues(typeof(eTransmision));
```

- El botón Aceptar deberá tomar los datos y validarlos. Si algún dato no es correcto, deberá pedir que el usuario revise los datos. En caso de que sean válidos, instanciar un Auto o una moto según corresponda debiendo utilizar la sobrecarga del operador + de la clase estacionamiento para agregar un vehículo. En caso de poder ser agregado, mostrar en un messagebox que el vehículo fue agregado satisfactoriamente. En caso de que no, notificar que no pudo ser agregado.

Nota: Antes de salir, setear la variable FrmAlta_Instancia en null.

- El botón cancelar simplemente seteara FrmAlta_Instancia en null y cerrará el form.
- El evento FormClosing debe setear FrmAlta_Instancia en null.
- El evento CheckedChange del radiobutton Auto deberá ocultar cilindrada (label y textbox), y mostrar el label de transmisión y el combobox de transmisión. El de Moto, lo opuesto.

Clase **FrmPrincipal**:

- Setearlo como contenedor MDI en el constructor (IsMdiContainer).
- En el método Load instanciar un estacionamiento. Utilizar el método estático InstanciarEstacionamiento de la clase Estacionamiento.
- Agregar un MenoStrip el cual contendrá 2 ítems:
 - Nuevo cliente
 - Instanciará un formulario de tipo Alta utilizando el método estático “Crear instancia” de la clase Alta.
 - El formulario Alta deberá tener como padre al formulario **actual**.
 - Lanzará el formulario de tipo Alta.
 - Listar información
 - Deberá mediante un MessageBox, mostrar la información total del Estacionamiento. Deberá pasar como parámetro la información del estacionamiento (utilizar sobreescritura del método ToString) y el nombre del estacionamiento accediendo al valor de tu atributo nombre.