

Universidad Tecnológica Nacional
Facultad Regional Avellaneda



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

Materia: Laboratorio de Programación II

Apellido:		Fecha:	10-05-2022
Nombre:		Docente ⁽²⁾ :	
División:	2°C	Nota ⁽²⁾ :	
Legajo:		Firma ⁽²⁾ :	
Instancia ⁽¹⁾ :	<div style="display: flex; justify-content: space-around; align-items: center;"> <div>PP</div> <div></div> <div>RPP</div> <div>X</div> <div>SP</div> <div></div> <div>RSP</div> <div></div> <div>FIN</div> <div></div> </div>		

(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.

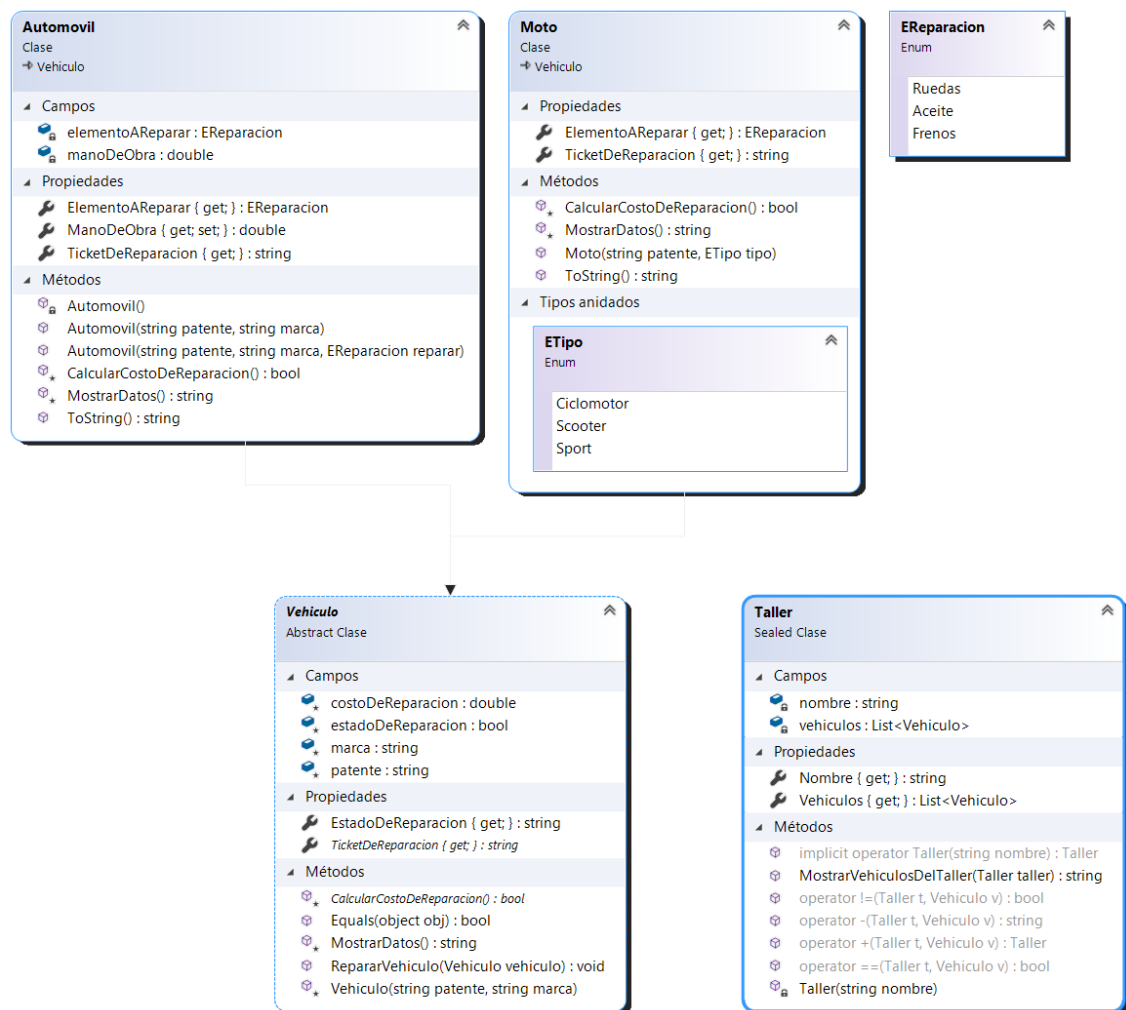
(2) Campos a ser completados por el docente.

IMPORTANTE:

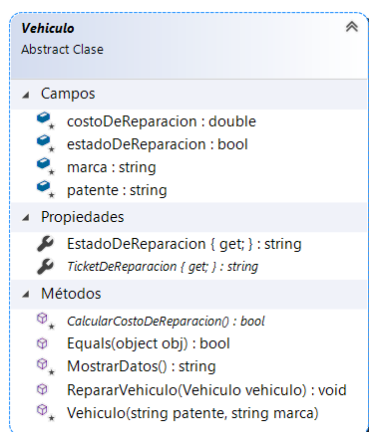
- **2 (dos) errores en el mismo tema anulan su puntaje.**
- La correcta documentación y reglas de estilo de la cátedra serán evaluadas.
- Colocar sus datos personales en el nombre de la carpeta principal y la solución: Apellido.Nombre.Div. Ej: Pérez.Juan.2D. No se corregirán proyectos que no sea identificable 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.

Se desea desarrollar una aplicación que permita controlar el ingreso y egreso de vehículos a un taller de reparaciones. Para ello se debe:

1. Crear un proyecto de tipo biblioteca de clases y colocar el siguiente esquema:

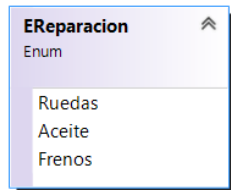


2. Clase Vehículo:

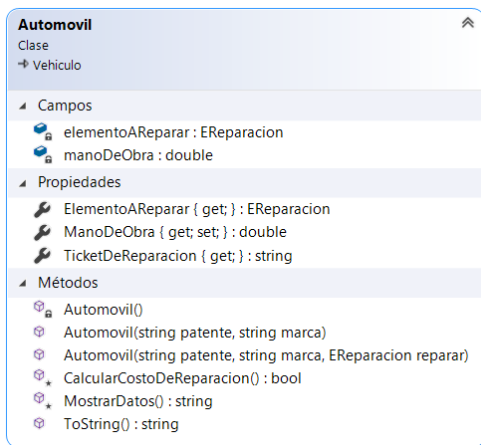


- Sera abstracta. Todos sus atributos, así como su constructor, serán protegidos.
- El atributo `estadoDeReparacion` deberá inicializarse en **false**.
`costoDeReparacion` en 0 (**cero**)
- Sus propiedades serán de solo lectura.
- La propiedad `EstadoDeReparacion` retornara un string. En caso de su estado ser **false** devolverá: **"Reparacion Pendiente"**, de lo contrario **"Reparado"**.

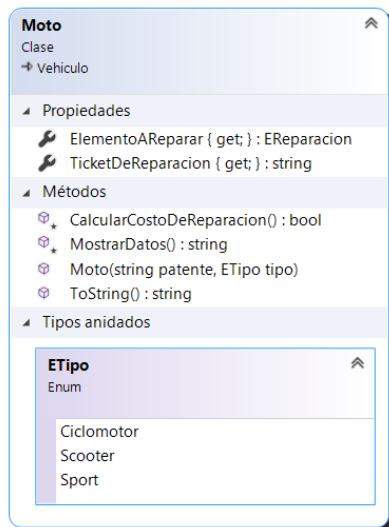
- e. Sobre escribir la funcionalidad del Equals de manera tal que 2 vehículos sean iguales si su patente es igual.
 - f. MostrarDatos será protegido. Expondrá los datos de Patente, Marca y EstadoDeReparacion (Para este último, usar la propiedad). Utilizar StringBuilder.
 - g. RepararVehiculo será de clase, y cambiará el estado de reparación de un vehículo a reparado (**true**).
3. Enumerado EReparacion:
- a. Contendrá los siguientes elementos: {Ruedas=500,Aceite=3000,Frenos=5400}



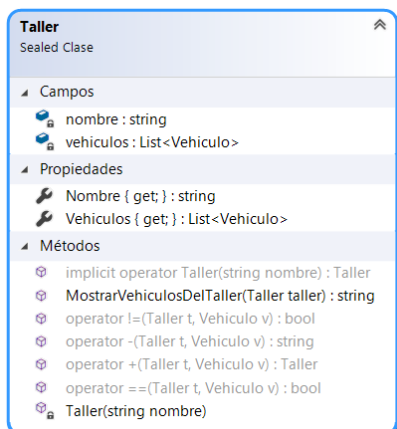
4. Clase Automóvil:



- a. Heredara de vehículo. Todos sus atributos serán privados.
 - b. manoDeObra, será de la clase y su valor se inicializará en su constructor (de clase), siendo el valor inicial: 932.99D.
 - c. Por defecto a un Automóvil se le reparara las ruedas.
 - d. CalcularCostoDeReparacion, calculara el costoDeReparacion y retornara **true** solo si el vehículo ya fue reparado (estadoDeReparacion). El costoDeReparacion se deducirá de sumar la manoDeObra y el valor de elementoAReparar.
 - e. La propiedad TicketDeReparacion retornara un string. Si no se pudo CalcularCostoDeReparacion, devolverá: **"Automovil aun en taller"**. De lo contrario devolverá: **"Ticket Nro {utilizar GetHashCode de la instancia}\nImporte: \${costoDeReparacion}"**.
 - f. MostrarDatos agregara elementoAReparar y el TicketDeReparacion.
 - g. La propiedad ElementoAReparar será de solo lectura.
 - h. La propiedad ManoDeObra en el **Set** solo permitirá cambiar el valor, solo si el nuevo importe es mayor al actual.
 - i. Sobre escribir ToString para exponer los datos del Automóvil.
5. Clase **Moto**:



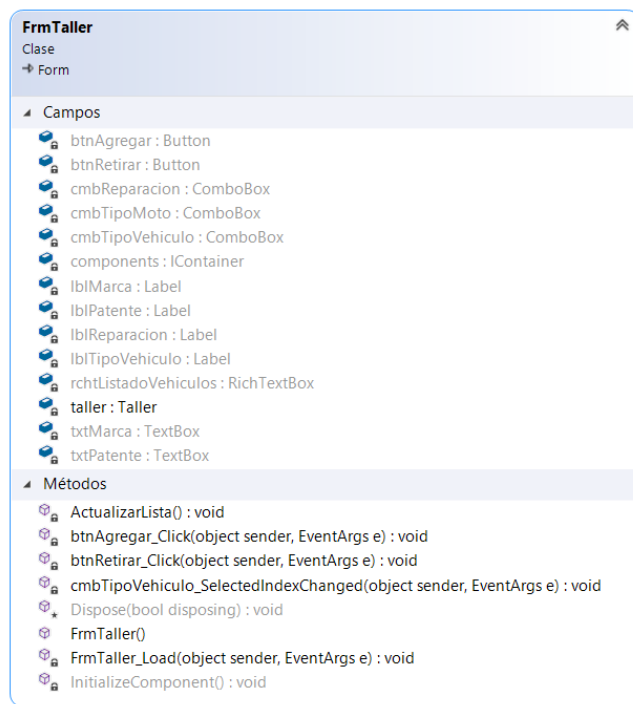
- Heredara de Vehículo.
 - Sus propiedades serán de solo lectura. Por defecto a las motos, solo le reparara el aceite.
 - Tendrá un enumerado ETipo. {Ciclomotor=10, Scooter=20, Sport = 30 }.
 - La marca en Moto será igual al tipo de Moto.
 - CalcularCostoDeReparacion, calculara el costoDeReparacion y retornara **true** solo si el vehículo ya fue reparado (estadoDeReparacion). El costoDeReparacion se deducirá del valor del ElementoAReparar. A este valor se incrementará porcentualmente en base al tipo de moto (+10% Ciclomotor, +20% Scooter y +30% Sport). Para recuperar el tipo de Moto utilizar el siguiente código:
 - Enum.Parse(typeof(ETipo), base.marca);
 - MostrarDatos agregara elementoAReparar y el TicketDeReparacion.
 - La propiedad TicketDeReparacion retornara un string. Si no se pudo CalcularCostoDeReparacion, devolverá: **"Moto aun en taller"**. De lo contrario devolverá: **"Ticket Nro {utilizar GetHashCode de la instancia}\nImporte: \${costoDeReparacion}"**.
 - Sobre escribir ToString para exponer los datos de Moto.
6. Clase **Taller**:



- No se podrá heredar de dicha clase.
- Sus atributos y único constructor serán privados. Este último será el encargado de inicializar los atributos.

- c. Sus propiedades de solo lectura.
 - d. Sobre cargar implícitamente String de modo tal que retorne una Instancia de Taller.
 - e. MostrarVehiculosDelTaller será de clase y retornará un string con los datos de todos los vehículos del taller. Utilizar StringBuilder.
 - f. Un Taller y un Vehículo serán iguales si el vehículo se encuentra en la lista del taller (comparar por patente).
 - g. La sobre carga del operador + permitirá agregar un vehículo a la lista del taller, solo si este aun no fue agregado.
 - h. La sobre carga del operador – retornara un string y será el encargado de retirar un vehículo de la lista. Este solo se podrá retirar si el vehículo se encuentra en el taller, de lo contrario se informara "**Vehículo no encontrado**". Antes de retirarlo se deberá cambiar el estado de reparación del vehículo a reparado, el mensaje a retornar será el TicketDeReparacion del vehículo.
7. Formulario:
- a. Diseñar un formulario como el siguiente que inicie centrado, no permita maximizar, ni modificar su tamaño. Se deberá ajustar las propiedades:
 - i. StartPosition en Center.
 - ii. MaximizeBox en False.
 - iii. FormBorderStyle en FixedSingle.

- b. Poseerá:
 - i. 3 ComboBox. Cambiar su propiedad DropDownStyle a DropDownList.
 - ii. 2 TextBox.
 - iii. 2 Botones, uno para agregar vehiculos y otro para retirar.
 - iv. 1 RichTextBox.
- c. Respetar el nombre de los controles y adicionar los atributos a la clase FrmTaller según el siguiente esquema:



8. En el constructor del FRM instanciar el taller.
9. El evento Load tendrá el siguiente código:

```
this.cmbTipoVehiculo.DataSource = new List<String> {"Automovil", "Moto"};
this.cmbReparacion.DataSource = Enum.GetValues(typeof(EReparacion));
this.cmbTipoMoto.DataSource = Enum.GetValues(typeof(Moto.ETipo));
this.cmbTipoMoto.Visible = false;
```

10. El evento SelectedIndexChanged de cmbTipoVehiculo tendrá el siguiente código:

```
if (this.cmbTipoVehiculo.SelectedItem.ToString() == "Moto")
{
    this.cmbTipoMoto.Visible = true;
    this.txtMarca.Visible = false;
    this.lblReparacion.Visible = false;
    this.cmbReparacion.Visible = false;
    this.cmbTipoMoto.Location = this.txtMarca.Location;
}
else
{
    this.cmbTipoMoto.Visible = false;
    this.txtMarca.Visible = true;
    this.lblReparacion.Visible = true;
    this.cmbReparacion.Visible = true;
}
```

11. Agregar un método privado denominado ActualizarLista y que retorne Void, el cual tendrá el siguiente código:

```
this.rchListadoVehiculos.Text = Taller.MostrarVehiculosDelTaller(this.taller);
```

12. El evento click del boton agregar tendrá el siguiente código:

```
Vehiculo nuevoVehiculo;
if (this.cmbTipoVehiculo.SelectedItem.ToString() == "Moto")
{
    nuevoVehiculo = new Moto(this.txtPatente.Text,
(Moto.ETipo)this.cmbTipoMoto.SelectedItem);
}
else
{
    nuevoVehiculo = new Automovil(this.txtPatente.Text, txtMarca.Text,
(EReparacion)this.cmbReparacion.SelectedItem);
}
this.taller += nuevoVehiculo;
this.ActualizarLista();
```

13. El evento click del boton retirar tendrá el siguiente código:

```
Vehiculo vehiculoARetirar;
if (this.cmbTipoVehiculo.SelectedItem.ToString() == "Moto")
{
    vehiculoARetirar = new Moto(this.txtPatente.Text,
(Moto.ETipo)this.cmbTipoMoto.SelectedItem);
}
else
{
    vehiculoARetirar = new Automovil(this.txtPatente.Text,
txtMarca.Text, (EReparacion)this.cmbReparacion.SelectedItem);
}

MessageBox.Show((this.taller - vehiculoARetirar), "Salida",
MessageBoxButtons.OK, MessageBoxIcon.Information);
this.ActualizarLista();
```

