

Universidad Tecnológica Nacional
Facultad Regional Avellaneda



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

Materia: Laboratorio de Programación II

Apellido:		Fecha:	20-10-2021
Nombre:		Docente ⁽²⁾ :	
División:		Nota ⁽²⁾ :	
Legajo:		Firma ⁽²⁾ :	
Instancia ⁽¹⁾ :	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">PP</div> <div style="text-align: center;"><input type="checkbox"/></div> <div style="text-align: center;">RPP</div> <div style="text-align: center;"><input checked="" type="checkbox"/></div> <div style="text-align: center;">X</div> <div style="text-align: center;">SP</div> <div style="text-align: center;"><input type="checkbox"/></div> <div style="text-align: center;">RSP</div> <div style="text-align: center;"><input type="checkbox"/></div> <div style="text-align: center;">FIN</div> <div style="text-align: center;"><input type="checkbox"/></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.

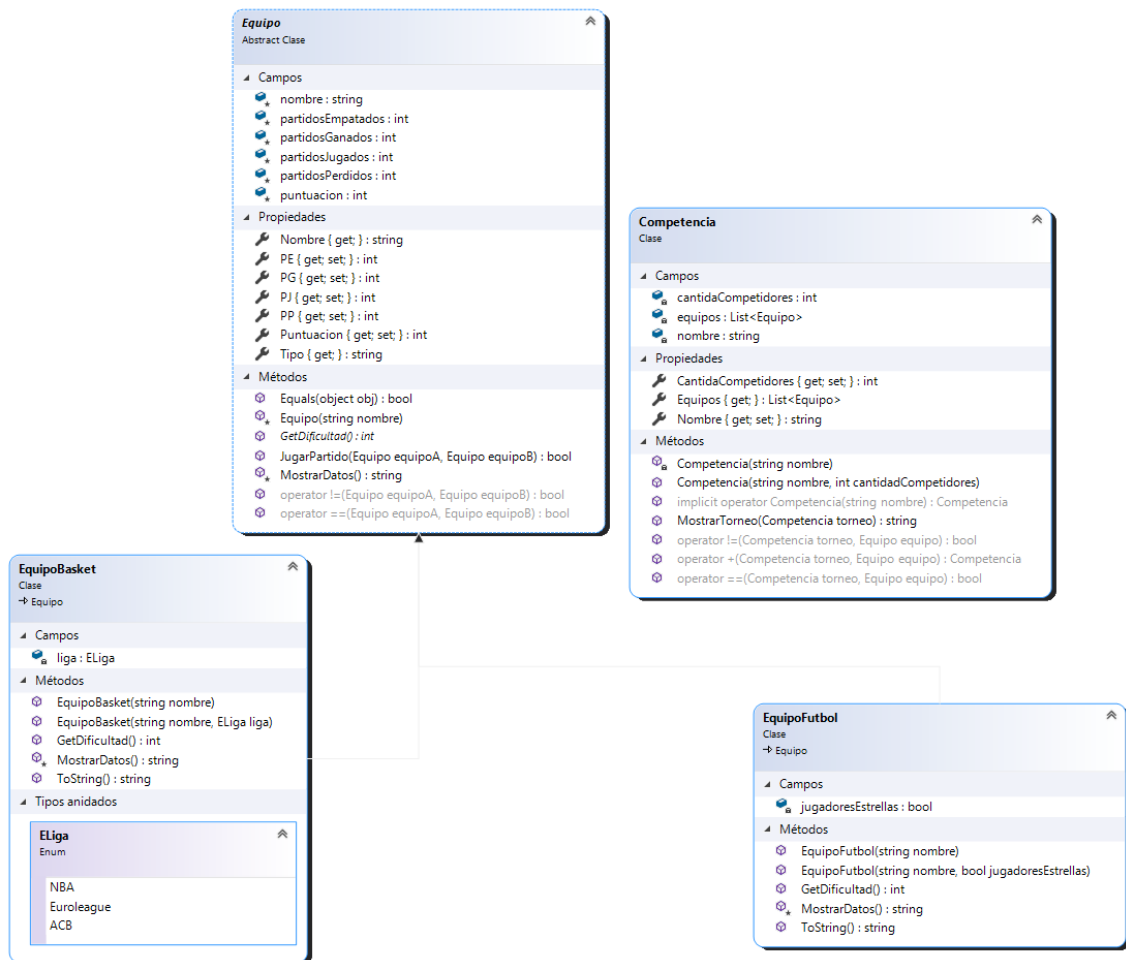
(2) Campos a ser completados por el docente.

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 simule una carrera entre 2 vehículos de F1.
Para ello se debe:

- Crear un proyecto de tipo biblioteca de clases y con el siguiente esquema:



2. Clase Equipo:

- Sera abstracta.
- Todos sus atributos protegidos asi como su constructor son protegidos.
- Las propiedades Nombre y Tipo serán de solo lectura. Tipo retornara la propiedad Name del Type de la instancia.
- Sobrescribir el Equals para comparar los Types de Equipo.
- La sobrecarga del operador == retornara True si los nombres y el Type del equipo son iguales (Reutilizar código).
- MostrarDatos es protegido y solo retornara el nombre del Equipo. Utilizar StringBuilder.
- JugarPartido:
 - Será de clase, recibe 2 equipos, y será el método encargado de simular el juego entre estos.
 - Este método retornara True si se pudo Jugar el partido, el cual, solo se podrá llevar a cabo si ambos Equipos son del mismo Type (Reutilizar Código).
 - Se comparará la dificultad de cada Equipo a través del método GetDificultad para determinar el ganador.
 - Al jugar partido se deberá a ambos Equipos incrementar el valor de **Partidos Jugados**.

- v. El que tenga la mayor dificultad (el ganador) sumara **3 Puntos**, en caso de ser igual (empate), **ambos** sumaran **1 Punto**.

h. GetDificultad deberá implementarse en sus clases derivadas.

3. Clase EquipoFutbol:

- a. Hereda de Equipo.
- b. Poseerá un atributo booleano, para determinar si posee o no "jugadoresEstrellas".
- c. Por defecto un EquipoFutbol no poseerá "jugadoresEstrellas".
- d. GetDificultad retornará un numero Random entre 7 y 10. En caso de poseer "jugadoresEstrellas" este valor se deberá de duplicar.
- e. MostrarDatos agregara "Futbol –"al nombre del objeto.
- f. ToString hará público MostrarDatos.

4. Clase EquipoBasket:

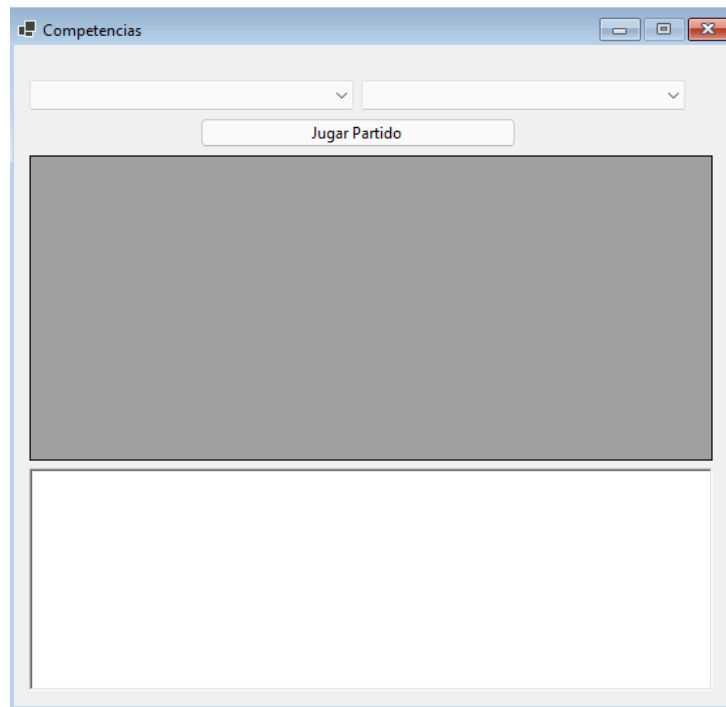
- a. Hereda de Equipo.
- b. Poseerá un Enum con los siguientes tipos: {NBA,Euroleague,ACB}
- c. El atributo Liga será del tipo del enumerado y privado.
- d. Por defecto EquipoBasket su liga será NBA.
- e. GetDificultad retornara un numero Random. Si el equipo es de la liga NBA, su valor será en 8 y 10. EuroLeague entre 5 y 10, ACB entre 1 y 7.
- f. MostrarDatos agregara "Basket –"al nombre del objeto.
- g. ToString hará público MostrarDatos.

5. Clase Competencia:

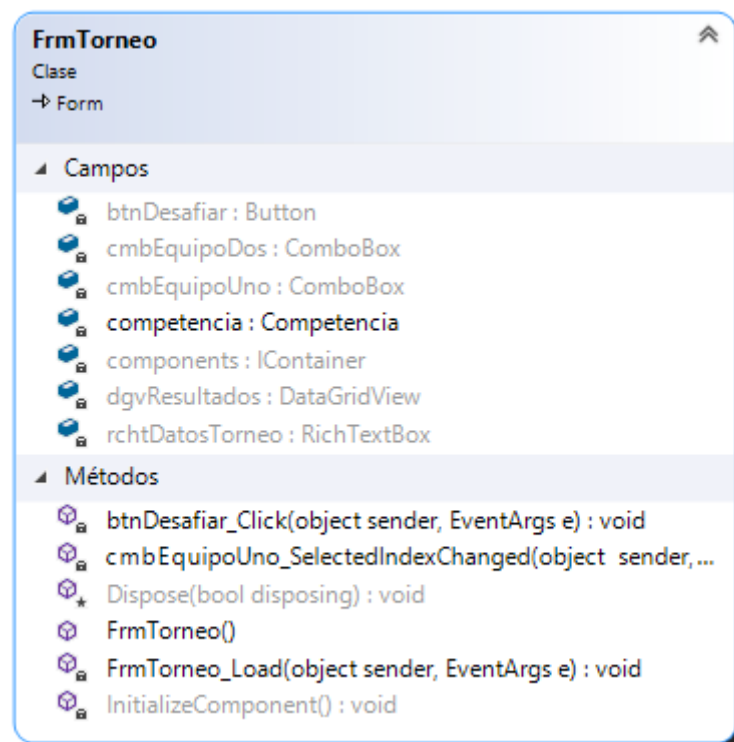
- a. Todos sus atributos son privados. Poseerá una lista de Equipos, un nombre y una cantidad de Competidores.
- b. Su constructor privado asignara por defecto 5 (cinco) a cantidadCompetidores.
- c. Su constructor público:
 - i. Instanciara la lista.
 - ii. Asignara un nombre.
 - iii. Inicializara la cantidad de competidores.
- d. La propiedad Equipos será de solo Lectura.
- e. La conversión Implícita de un string retornara una instancia de una Competencia.
- f. Una Competencia y un Equipo serán iguales si este ya se encuentra de dentro de la competencia. Comparar por nombre (reutilizar código).
- g. Sobrecargar el operador + para agregar un Equipo a la competencia. Solo se podrá agregar si este no se encuentra en esta y la competencia aun dispone de capacidad.
- h. MostrarTorneo será de Clase, recibe una competencia y expondrá los datos de la competencia en conjunto con la lista de competidores. Utilizar StringBuilder.

6. Formulario:

- a. Desarrollar un formulario similar al siguiente:



- b. El mismo debería aparecer Centrado en pantalla, su propiedad `FormBorderStyle` será `FixedSingle` y deberán de establecerse en `False` la propiedad `MaximizeBox`.
 - i.



- c. El mismo poseerá
 - i. 2 ComboBox (La propiedad `DropDownStyle`, deberá establecerse en `DropDownList`).
 - ii. 1 Button.
 - iii. 1 RichText.

- iv. 1 DataGridView.
- v. Respetar los nombres del diagrama.

d. En el Evento Load Colocar el Siguiente codigo:

```
this.competencia = new Competencia("Torneos el Foquito", 10);
Equipo e1 = new EquipoFutbol("Independiente",true);
Equipo e2 = new EquipoFutbol("Boca",true);
Equipo e3 = new EquipoBasket("Spurs");
Equipo e4 = new EquipoBasket("Chicago Bulls");
Equipo e5 = new EquipoFutbol("Racing");

this.competencia += e1;
this.competencia += e1;//pruebo de agregar el mismo equipo
this.competencia += e2;
this.competencia += e3;
this.competencia += e4;
this.competencia += e5;

this.cmbEquipoUno.DataSource = this.competencia.Equipos;

this.rchtDatosTorneo.Text +=
Competencia.MostrarTorneo(this.competencia);
```

e. En el método SelectIndexChanged del cmbEquipoUno, colocar el siguiente código:

```
this.cmbEquipoDos.DataSource =
competencia.Equipos.FindAll(element => element !=
(Equipo)this.cmbEquipoUno.SelectedItem);
```

f. En el evento Click de btnDesafiar colocar el siguiente código:

```
if(Equipo.JugarPartido((Equipo)this.cmbEquipoUno.SelectedItem,
(Equipo)cmbEquipoDos.SelectedItem))
{
    MessageBox.Show("Se Jugo el partido");
    this.dgvResultados.DataSource = null;
    this.dgvResultados.DataSource = this.competencia.Equipos;
}
else
{
    MessageBox.Show("No se puede jugar el partido entre estos 2
equipos");
}
```