

Universidad Tecnológica Nacional Facultad Regional Avellaneda



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

Materia: Laboratorio de Programación II

Apellido:		Fecha:	04/10/2018
Nombre:		Docente ⁽²⁾ :	F. Dávila/H. Dillon
División:	2°C	Nota ⁽²⁾ :	
Legajo:		Firma ⁽²⁾ :	
Instancia ⁽¹⁾ :	<div style="display: flex; justify-content: space-around;"> PP X RPP </div>	<div style="display: flex; justify-content: space-around;"> SP RSP </div>	<div style="display: flex; justify-content: space-around;"> FIN </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 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.

TIEMPO MÁXIMO PARA RESOLVER EL EXAMEN 90 MINUTOS.

1. Crear una solución con el nombre en el siguiente formato: [APELLIDO].[NOMBRE]
2. Dentro crear 3 proyectos: *Entidades* (Class Library), *VistaConsola* (Console) y *VistaForm* (WindowsForms).
3. Dentro del **Program**, en el **Main** de *VistaConsola*, colocar el siguiente código para probar las entidades:

```
// Genero un equipo nuevo
Equipo equipo= new Equipo("UTN");
// Genero un Director Técnico y lo asigno al equipo
DirectorTecnico dt = new DirectorTecnico("Roberto", "Gonzalez", 60, 8332112, 10);
DirectorTecnico dtInvalido = new DirectorTecnico("Angel", "Ramirez", 90, 6332112, 1);

equipo.DirectorTecnico = dtInvalido;
// Genero Jugadores
Jugador j1 = new Jugador("Juan", "López", 28, 33222456, 70.5f, 1.82f, Posicion.Arquero);
Jugador j2 = new Jugador("José", "Martínez", 28, 33222456, 72f, 1.70f, Posicion.Defensor);
Jugador j3 = new Jugador("Pedro", "Gonzalez", 31, 31333444, 73.5f, 1.90f,
Posicion.Delantero);
Jugador j4 = new Jugador("Matías", "Rodríguez", 32, 30443332, 75.5f, 1.75f,
Posicion.Central);
Jugador j5 = new Jugador("Jugador5", "Inválido", 26, 36555443, 99.5f, 1.73f,
Posicion.Delantero);
Jugador j6 = new Jugador("Jugador6", "Inválido", 50, 37456678, 86.5f, 1.94f,
Posicion.Delantero);
Jugador j7 = new Jugador("Martín", "Perez", 26, 36555443, 66.5f, 1.73f, Posicion.Defensor);
```

```

Jugador j8 = new Jugador("Diego", "Roa", 25, 37456678, 86.5f, 1.94f, Posicion.Delantero);
// Agrego los jugadores al equipo
equipo = equipo + j1;
equipo = equipo + j2;
equipo = equipo + j3;

Console.WriteLine("*****EQUIPO INVÁLIDO*****");
// Imprimo los datos del equipo
Console.WriteLine((string)equipo);

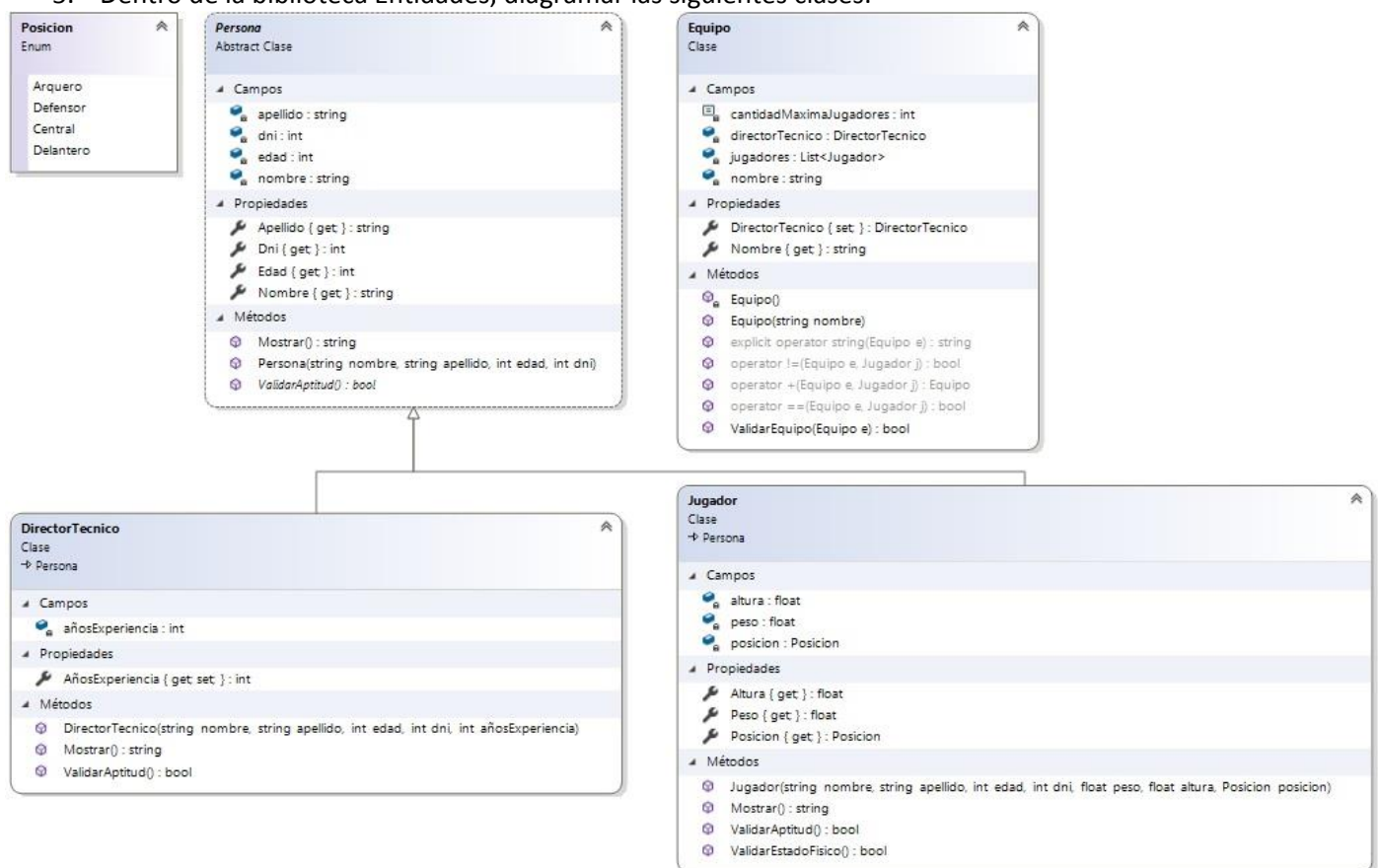
if(Equipo.ValidarEquipo(equipo))
Console.WriteLine("El equipo es válido");
else
Console.WriteLine("El equipo no es válido");

equipo.DirectorTecnico = dt;
equipo = equipo + j4;
equipo = equipo + j5;
equipo = equipo + j6;
equipo = equipo + j7;
equipo = equipo + j8;

Console.WriteLine("*****EQUIPO VÁLIDO*****");
// Imprimo los datos del equipo
Console.WriteLine((string)equipo);
if(Equipo.ValidarEquipo(equipo))
Console.WriteLine("El equipo es válido");
else
Console.WriteLine("El equipo no es válido");
Console.ReadKey();

```

4. El ejercicio consiste en armar un equipo para un deporte “X”.
5. Dentro de la biblioteca Entidades, diagramar las siguientes clases:



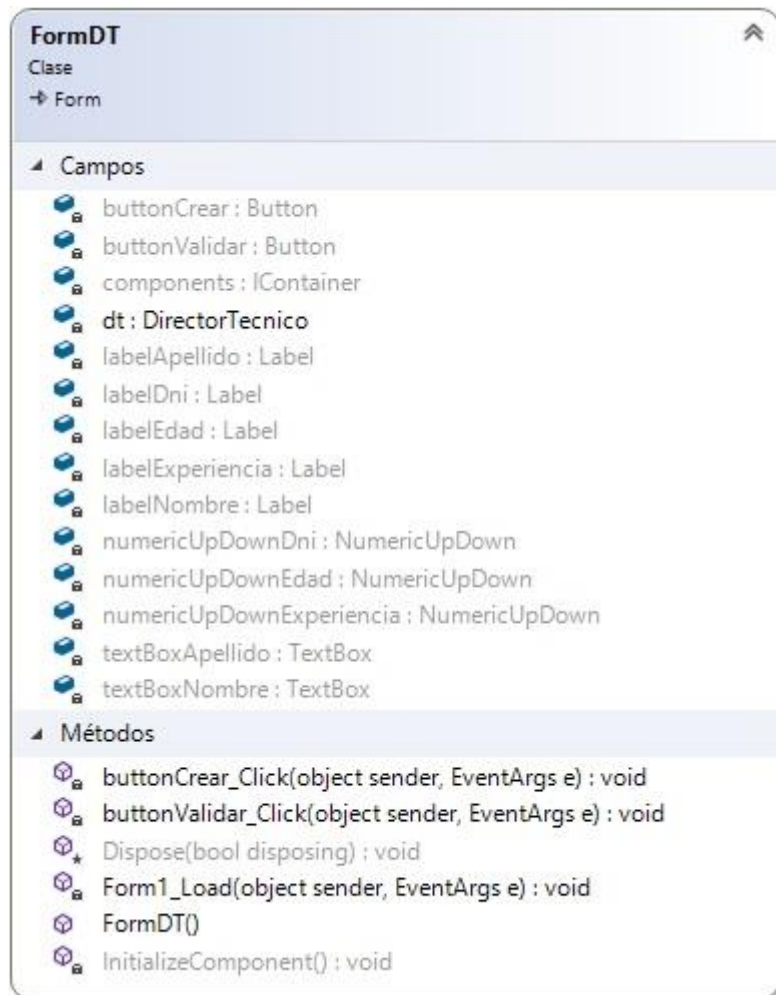
6. Clase Persona:
 - a. *Mostrar* tendrá implementación en la clase Persona y podrá ser sobreescrita en las clases derivadas. Deberá retornar todos los datos de la persona utilizando StringBuilder para compilar dicha información.

- b. *ValidarAptitud* no tendrá implementación dentro de Persona.
- 7. Clase DirectorTecnico:
 - a. *Mostrar* retornará los datos de DirectorTecnico y su clase base.
 - b. *ValidarAptitud* validará que el Director Técnico sea menor a 65 años y tenga al menos 2 años de experiencia.
- 8. Clase Jugador:
 - a. *Mostrar* retornará los datos de Jugador y su clase base.
 - b. *ValidarEstadoFisico* deberá validar que el índice de masa corporal se encuentre en el rango de 18.5 y 25 **inclusive**. Para calcularlo, utilizar la siguiente fórmula:
IMC = peso / altura^2
 - c. *ValidarAptitud*: Un jugador será apto para ingresar a un equipo siempre que su edad no supere los 40 años y se encuentre en estado físico
- 9. Clase Equipo:
 - a. El **constructor privado** será el único lugar donde se instanciará la lista de jugadores.
 - b. La **constante** cantidadMaximaJugadores tendrá un valor de 6.
 - c. La propiedad DirectorTecnico será utilizada para asignar un director técnico al equipo siempre y cuando cumpla con las condiciones de aptitud.
 - d. El **operador explícito** retornará los datos del equipo, su director técnico (en caso de no tener uno asignado aún, mostrar en su lugar el string "Sin DT asignado") y todos sus jugadores, utilizando StringBuilder para compilar dicha información.
 - e. El **operador ==** entre Equipo y Jugador informará true si esa instancia de Jugador ya se encuentra agregada al equipo.
 - f. El **operador +** entre Equipo y Jugador agregará el jugador al equipo siempre y cuando:
 - i. Este no haya sido agregado aún.
 - ii. No haya sido superada la cantidad de integrantes.
 - iii. Cumpla las condiciones de aptitud.
 - g. *ValidarEquipo*: Método **de clase** que retorna true en caso de que un equipo sea válido, para lo cual debe cumplir con las siguientes condiciones:
 - i. Tener un Director Técnico asignado.
 - ii. Tener al menos un jugador de c/posición.
 - iii. Tener sólo 1 arquero.
 - iv. Cumplir con la cantidad de integrantes completa (la cantidad de jugadores agregados a la lista debe ser igual a la constante "cantidadMaximaJugadores").
- 10. Por último, generar el siguiente **formulario** dentro de *VistaForm*:


The image shows a Windows application window titled "FormDT". Inside the window, there are five input fields arranged vertically, each with a label to its left: "Nombre", "Apellido", "Edad", "DNI", and "Experiencia". The "Edad", "DNI", and "Experiencia" fields are numeric spinners, currently showing the value "0". Below these input fields, there are two buttons: "Crear" on the left and "Validar" on the right. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

Siendo los elementos a utilizar Button, Label, NumericUpDown y TextBox.

11. El formulario tendrá un atributo del tipo DirectorTecnico. Con el boton “Crear” se deberá instanciar dicho atributo con los datos cargados en el formulario y mostrar al usuario un mensaje que diga “Se ha creado el DT!” .
12. Con el botón Validar se deberá mostrar al usuario un mensaje que diga:
 - a. “Aún no se ha creado el DT del formulario” en caso de que el director técnico del formulario aún no se haya instanciado
 - b. “El DT es apto.” si el DT cumple con las validaciones de aptitud.
 - c. “El DT no es apto.” si el DT no cumple con las validaciones de aptitud.



Al finalizar, colocar la carpeta de la Solución completa en un archivo ZIP que deberá tener como nombre Apellido.Nombre.division.zip y dejar este último en el Escritorio de la máquina.

Luego presionar el botón  de la barra superior, colocar un mensaje y apretar **Aceptar**. Finalmente retirarse del aula y aguardar por la corrección.