

**Universidad Tecnológica Nacional  
Facultad Regional Avellaneda**



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

**Materia: Laboratorio de Programación II**

Apellido:		Fecha:	11/11/2021
Nombre:		Docente <sup>(2)</sup> :	
División:	2°C	Nota <sup>(2)</sup> :	
Legajo:		Firma <sup>(2)</sup> :	
Instancia <sup>(1)</sup> :	<div style="display: flex; justify-content: space-around;"> <span><b>PP</b></span> <span><b>RPP</b></span> <span><b>SP</b></span> <span><b>RSP</b></span> <span><b>FIN</b></span> </div>		

(1) Las instancias validas son: 1<sup>er</sup> Parcial (**PP**), Recuperatorio 1<sup>er</sup> Parcial (**RPP**), 2<sup>do</sup> Parcial (**SP**), Recuperatorio 2<sup>do</sup> Parcial (**RSP**), Final (**FIN**). Marque con una cruz.

(2) Campos a ser completados por el docente.

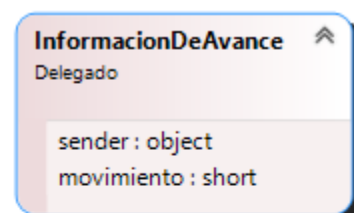
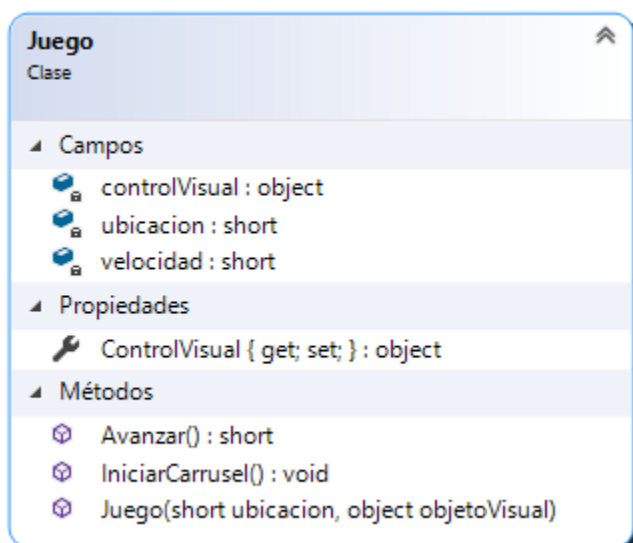
**IMPORTANTE:**

- 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 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.

*TIEMPO MÁXIMO PARA RESOLVER EL EXAMEN 90 MINUTOS.*

- Partir de la solución entregada. Modificar su nombre con el siguiente formato: [APELLIDO].[NOMBRE]

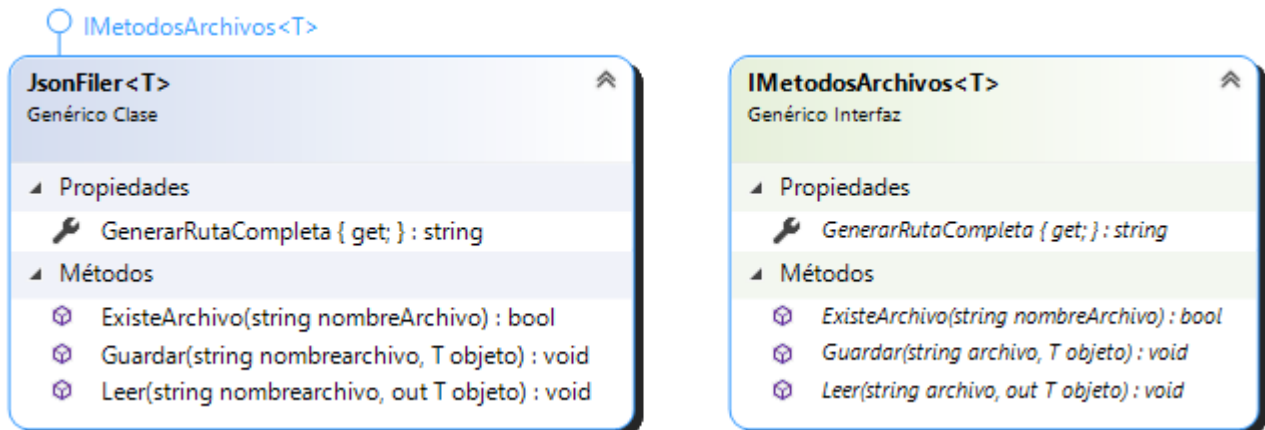
**Entidades**



- La velocidad será estática y su valor debe ser 8 al construir el primer objeto.
- Crear el delegado dentro del archivo de la clase Juego, como parte del namespace.
- Agregar el evento InformarAvance en la clase Juego.
  - Movimiento corresponderá con la nueva ubicación del elemento
  - Sender será si mismo, this.
- Completar el método IniciarCarrusel con la invocación del evento.
- Completar todo lo que haga falta para que se pueda serializar en Json guardando los datos velocidad y ubicación.
- Completar todo lo que haga falta para que el método IniciarCarrusel pueda ser cancelable al ejecutarse dentro de un hilo secundario.

## Archivos

- Dentro del proyecto **Archivos** se deberá respetar el siguiente esquema:



- Se deberán poder guardar los atributos velocidad y ubicación de la clase Juego. Modificar dicha clase para que esto pueda lograrse.
- Las clases implementarán la interfaz *IMetodosArchivos* para escribir y consultar los datos según corresponda.
- Los atributos de los métodos serán:
  - archivo: sólo el nombre del archivo, sin path (será válido "nombre.ext" y no "C:\nombre.ext").
  - Objeto a guardar.
- La propiedad `GenerarRutaCompleta` retornará la ruta al escritorio (debe ser independiente de la máquina en la que se abra el programa). Concatenar la barra final \ siendo la ruta retornada: `C:\...\Desktop\`.
- ExisteArchivo:**
  - Recordar concatenar `GenerarRutaCompleta` con `nombreArchivo` recibido antes de validar dicha existencia.
  - Comprobará si el archivo existe o no.
- De producirse excepciones en los métodos `Leer` y `Guardar` deberá encapsularse esa excepción en una nueva excepción propia `ErrorArchivosException` y lanzarla. Crear esta excepción con al menos 2 constructores dentro del proyecto de archivos.

## Formulario

- En el constructor deberán instanciarse los atributos `juegos` y `json`.
- En el método manejador de Load del Form:
  - Si el archivo dado por el atributo `archivoPath` existe, leerlo y ubicar su resultado en el atributo `juegos`.
  - Agregar al título de la ventana su apellido y nombre.
  - Asociar el evento con cada juego.
- En el evento click del botón Simular agregar un `foreach`, donde dentro se deberá:
  - Crear las tareas para el método `IniciarCarrusel` (un hilo por cada planeta).
  - Iniciar cada hilo.
- En el método `AvanceJuegos` lograr que se llame al hilo principal al método `CalcularUbicacion`
- En el método `FinalizarTasks` cancelar todos los hilos que aun estén ejecutándose.
- En el método manejador de `FormClosing`:

- a. Finalizar las tareas activas.
  - b. Guardar el archivo json con la información de los juegos.
14. Agregar un NumericUpDown llamado nudVelocidad. Al modificarse el valor, deberá modificar el atributo estático velocidad de la clase Juego.
15. El resultado deberá ser que al iniciar con el botón, nuestros "animales" van a empezar a girar en la calesita a la velocidad colocada en el campo de arriba a la izquierda; pudiendo esta acción ser detenida y reiniciada cuando queramos. Visualmente el formulario deberá ser este:



### Test Unitarios

16. Agregar 2 test unitarios:
- a. Intentar guardar y leer un archivo correctamente, comprobando que los datos guardados sean iguales a los recuperados.
  - b. Intentar guardar un archivo en una ruta inválida (una ruta vacía o con caracteres inválidos servirá para dicho plan), comprobando que se lance la excepción `ErrorArchivosException`.
  - c. Para obtener la sortija y ganarte otra vuelta: Hacer un test más, original, probando alguna funcionalidad compleja del sistema.

Al finalizar, colocar la carpeta de la carpeta de la Solución completa en un archivo ZIP que deberá tener como nombre Apellido.Nombre.division.zip y compartir este por Slack sólo con el docente titular de la cursada.