|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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(2): | | | |  | | | |
| División: | 2ºC | | | | Nota(2): | | | |  | | | |
| Legajo: |  | | | | Firma(2): | | | |  | | | |
| Instancia(1): | **PP** |  | **RPP** |  | | **SP** | X | **RSP** | |  | **FIN** |  |

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

**(2)** Campos a ser completados por el docente.

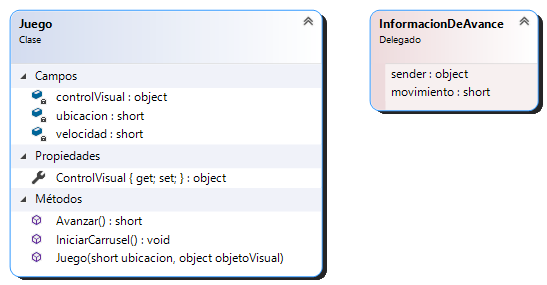
**IMPORTANTE:**

* **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 sé 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.*

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

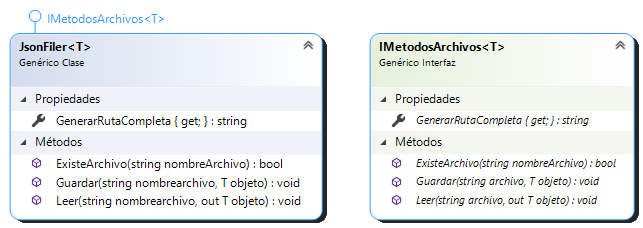
**Entidades**



1. La velocidad será estática y su valor debe ser 8 al construir el primer objeto.
2. Crear el delegado dentro del archivo de la clase Juego, como parte del namespace.
3. Agregar el evento InformarAvance en la clase Juego.
4. Completar el método IniciarCarrusel con la invocación del evento.
5. Completar todo lo que haga falta para que se pueda serializar en Json guardando los datos velocidad y ubicación.
6. Completar todo lo que haga falta para que el método IniciarCarrusel pueda ser cancelable al ejecutarse dentro de un hilo secundario.

**Archivos**

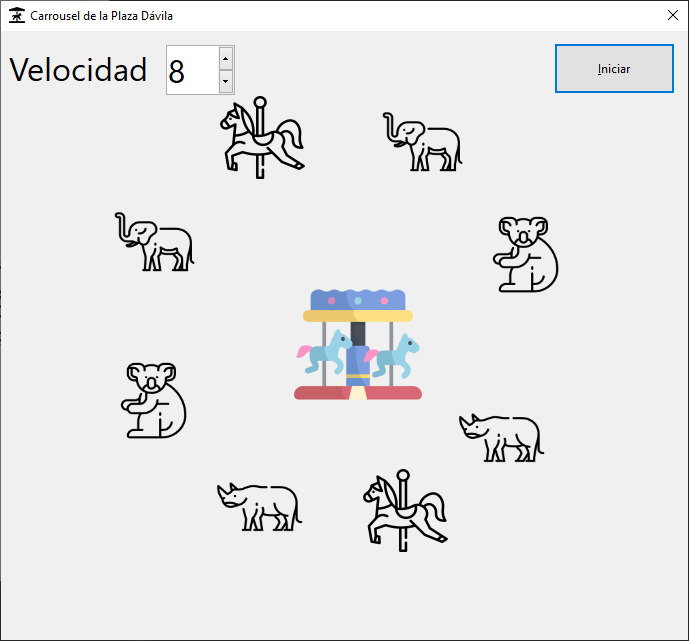
1. Dentro del proyecto **Archivos** se deberá respetar el siguiente esquema:



1. Las clases implementarán la interfaz *IMetodosArchivos* para escribir y consultar los datos según corresponda.
2. Los atributos de los métodos serán:
   1. archivo: sólo el nombre del archivo, sin path (será válido "nombre.ext" y no "C:\nombre.ext").
   2. Objeto a guardar.
3. 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**\**.
4. ExisteArchivo comprobará si el archivo existe o no (sumará GenerarRutaCompleta al nombreArchivo recibido).
5. De producirse excepciones en los método 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**

1. En el constructor deberán instanciarse los atributos juegos y json.
2. En el método manejador de Load del Form:
   1. Si el archivo dado por el atributo archivoPath existe, leerlo y ubicar su resultado en el atributo juegos.
   2. Agregar al título de la ventana su apellido y nombre.
   3. Asociar el evento con cada juego.
3. En el evento click del botón Simular agregar un foreach, donde dentro se deberá:
   1. Crear las tareas para el método IniciarCarrusel (un hilo por cada planeta).
   2. Iniciar cada hilo.
4. En el método AvanceJuegos lograr que se llame al hilo principal al método CalcularUbicacion
5. En el método FinalizarTasks cancelar todos los hilos que aun estén ejecutándose.
6. En el método manejador de FormClosing:
   1. Finalizar las tareas activas.
   2. Guardar el archivo json con la información de los juegos.
7. Agregar un NumericUpDown llamado nudVelocidad. Al modificarse el valor, deberá modificar el atributo estático velocidad de la clase Juego.
8. El resultado del formulario deberá ser este:



**Test Unitarios**

1. Agregar 2 test unitarios:
   1. Intentar guardar y leer un archivo correctamente, comprobando que los datos guardados sean iguales a los recuperados.
   2. 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.

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.