Universidad Tecnológica Nacional Facultad Regional Avellaneda			
Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos			
Materia: Laboratorio de Programación II			
Apellido:	Langer	Fecha:	14/06/2021
Nombre:	Denise Rocio	Docente(2):	
División:	2°D	Nota(2):	
Legajo:	110053	Firma(2):	

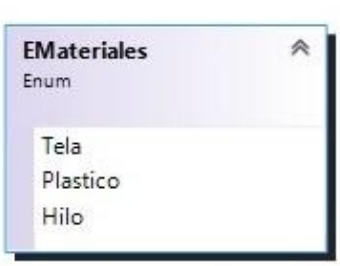
1. Introducción:

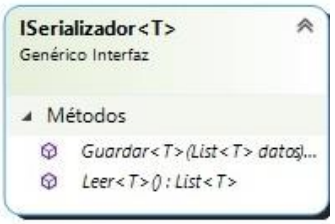
El sistema desarrollado para este trabajo práctico representa el proceso de fabricación para distintos tipos de juguetes (específicamente Muñecos, Peluches e Inflables), permitiéndole al usuario:

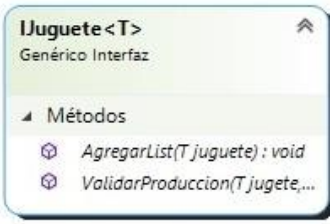
- Comprar la materia prima necesaria para continuar con el proceso.
- Seleccionar el/los juguetes a ser fabricados, pudiendo indicar distintas características y especificaciones según el producto.
- Editar el diseño completado previamente en caso de necesitar modificar ciertos atributos antes de su fabricación.
- Ver el historial de los últimos juguetes fabricados.

2. UML y Relación con temas explicados:

a. *Enumerados e Interfaces*





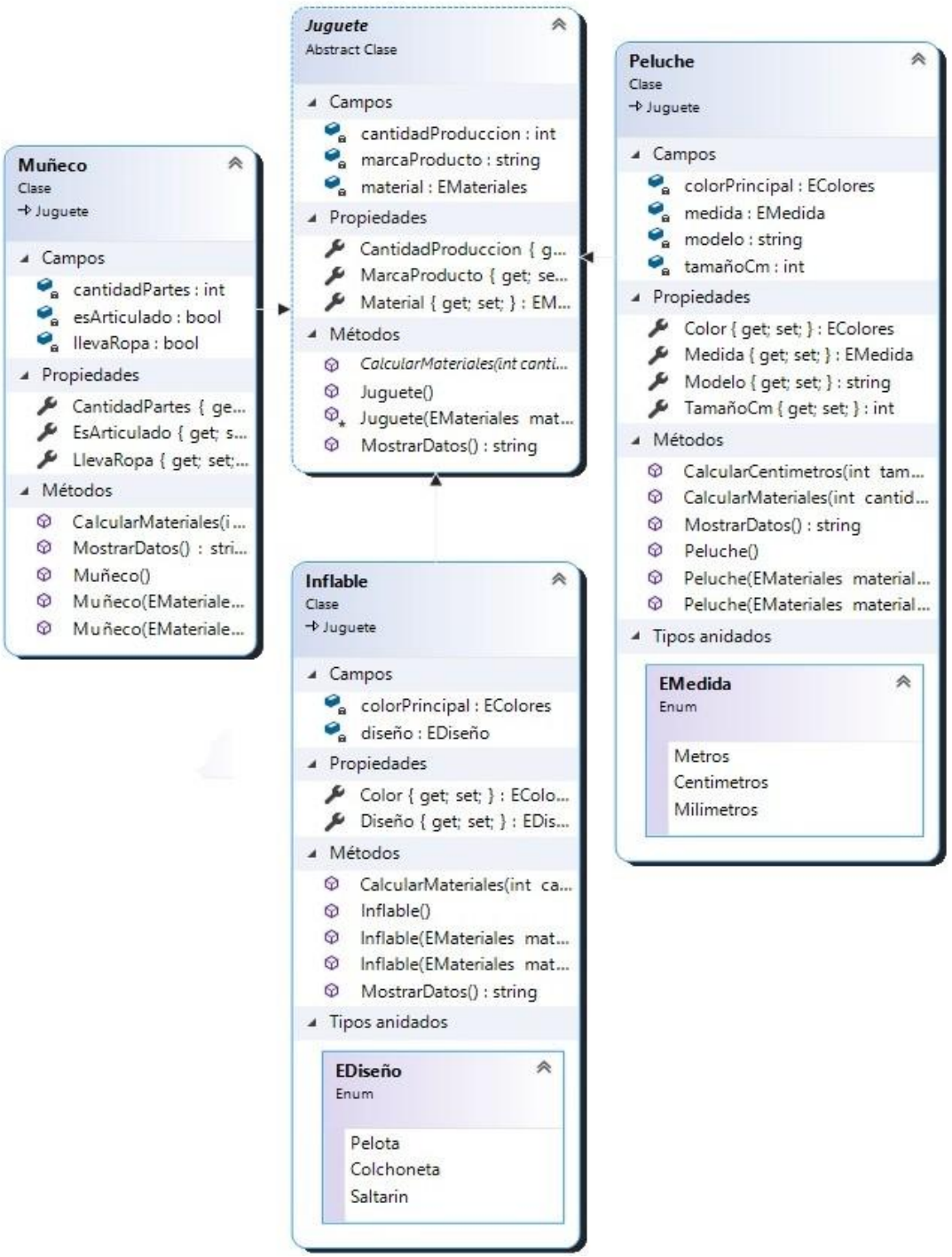


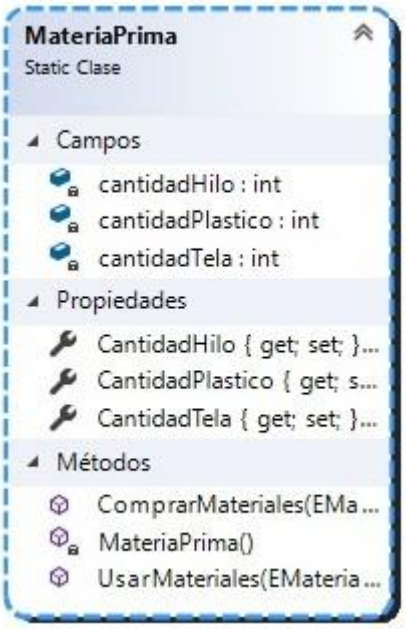
- **ISerializador<T>:** Interfaz genérica que contiene métodos para serializar y deserializar los archivos con formato XML.

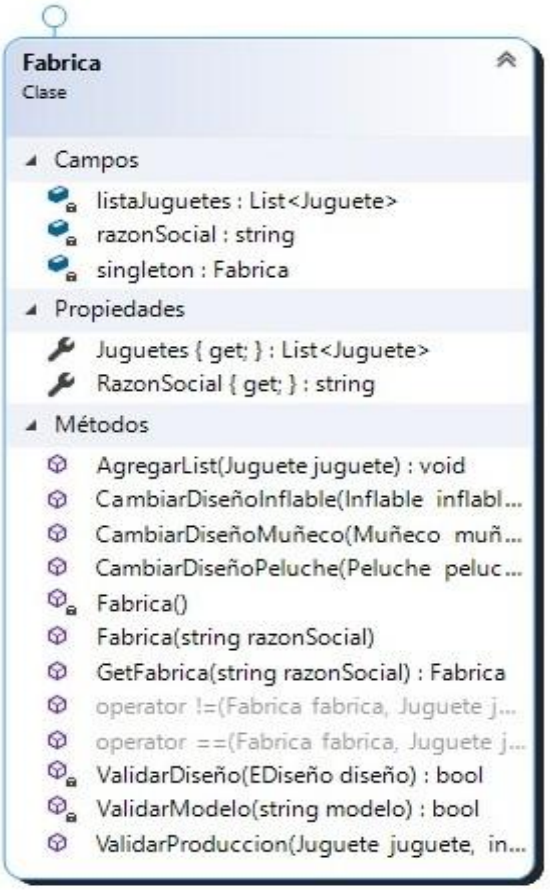
- ***Guardar<T>:*** Serializa los datos que se ingresa como parámetro en un archivo del tipo XML. En caso de no existir el archivo, lo crea.
- ***Leer<T>:*** Deserializa los datos de un archivo.XML a una lista de objetos, validando que exista.

- **IArchivos<T>:** Interfaz genérica que contiene métodos para la creación y lectura de archivos de texto.
  - **Guardar**: Escribe y agrega datos en un archivo de texto. En caso de no existir el archivo, lo crea.
  - **Leer**: Valida que exista un archivo de texto y retorna todo su contenido. En caso de no existir el archivo arroja una excepción.

b. Clases



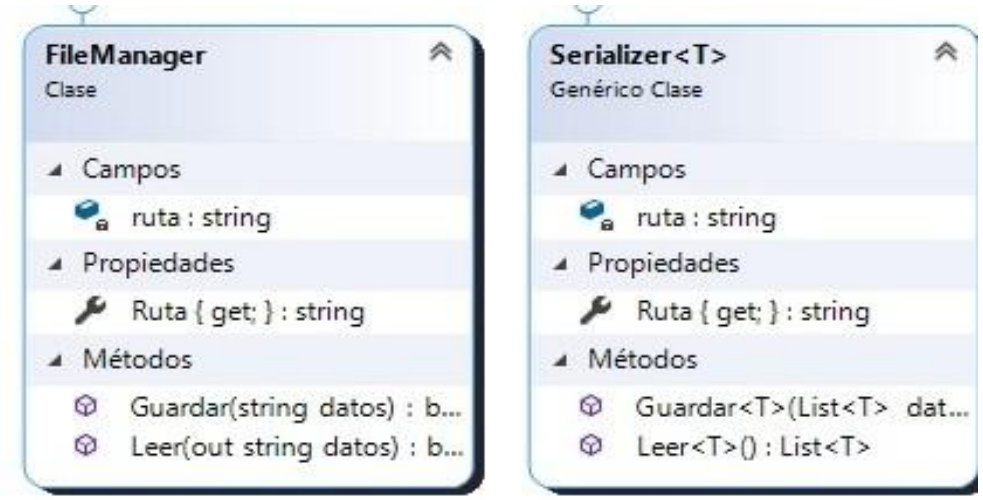




- **Juguete**: Clase abstracta que contiene los atributos, propiedades y métodos a ser heredados por sus clases derivadas (Muñeco, Inflable y Peluche).
- **Fábrica:** Clase que aplica singleton y la interfaz IJuguete<T>.
    - **GetFabrica**: Método estático que utiliza el patrón Singleton. Verifica que ya exista una instancia de Fábrica y la retorna. En caso contrario, instancia la clase y lo retorna.
    - **AgregarList:** Agrega un Juguete a la lista, validando que no se repitan (según criterios específicos de cada tipo de Juguete). En caso de que ya exista, arroja una excepción del tipo JugueteYaExisteException.
    - **ValidarProduccion:** Valida que haya materia prima suficiente para crear un Juguete, según la cantidad que se indique y el tipo de Material elegido. En caso de ser insuficiente, arroja una excepción del tipo NoMaterialesException.
    - **CambiarDiseño:** Modifica ciertos atributos de un Juguete (dependiendo de la clase) y actualiza la lista de Juguetes. En caso de error arroja la excepción JugueteYaExisteException.
    - **Sobrecarga del == :** Valida si un Juguete se repite en listaJuguetes y retorna el resultado con un valor booleano.
    - **ValidarModelo y ValidarDiseño**: Compara si el atributo del Juguete ingresado como parámetro se repite dentro de la lista.
- **Materia Prima:** Clase estática que contiene todos sus elementos estáticos.
    - **ComprarMateriales:** Agrega cantidad de materiales (número entero) a la Materia Prima que se indica como parámetro. En caso de fallas, arroja una excepción del tipo NoMaterialesException.

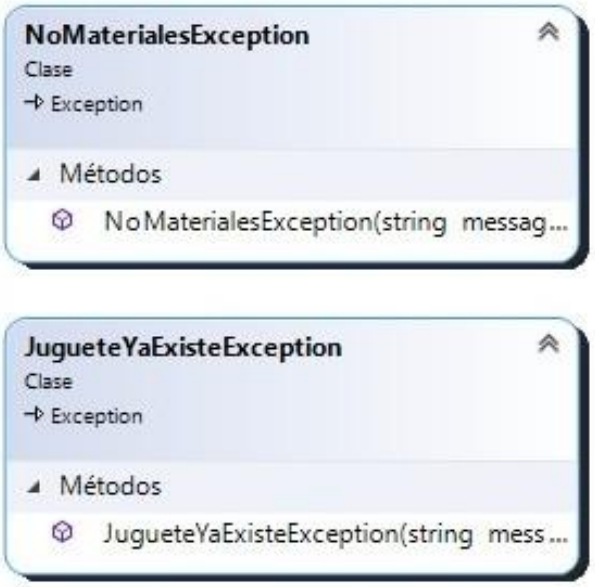
- **UsarMateriales:** Resta una cantidad de materiales (número entero) a la Materia Prima que se indica como parámetro. En caso de fallas, arroja una excepción del tipo NoMaterialesException.

c. *Archivos y Serialización*



- **FileManager:** Clase que implementa la Interfaz IArchivos<T>. Contiene los métodos y propiedades necesarios para la creación, escritura y lectura de archivos de texto. El archivo será creado en la ruta AppDomain.CurrentDomain.BaseDirectory con el nombre de “Errores.txt”.
- **Serializer**: Clase que implementa la Interfaz ISerializador<T>. Contiene los métodos y propiedades necesarios para la creación, escritura y lectura de archivos con formato XML. Los archivos serán creados en la ruta Environment.CurrentDirectory con los nombres de cada tipo de juguete.

d. *Excepciones*



- **JugueteYaExisteException**: Excepción que se lanzará cuando ya se registró un Juguete con la misma Marca y/o características.
- **NoMaterialesException**: Excepción que se lanzará cuando no haya suficientes materiales para fabricar un Juguete.

e. *Test Unitarios*

Se encuentran dentro del proyecto “EntidadesTests”. Se realizan los test unitarios para las funcionalidades principales desarrolladas en la biblioteca de clases, tanto para los casos de éxito como para los casos de fallas (validando que se arroje la excepción detallada).

3. Aclaraciones

a. Se crea un proyecto de consola con el nombre “Test” para probar las funcionalidades principales del proyecto (creación de los objetos, agregarlos dentro de la lista, modificación de sus atributos y mostrar los datos actualizados)

b. El tema relacionado a Genericos se implementa en las interfaces.

c. Dentro del formulario principal se instancia un objeto del tipo SoundPlayer, el cual también cuenta con la validación de que exista el archivo especificado.

d. El archivo XML se crea al momento de Fabricar los juguetes de la lista y el archivo de texto se crea al momento de arrojar la primera excepción, guardando todas las excepciones que serán lanzadas durante la ejecución.