Universidad Tecnológica Nacional

Facultad Regional Avellaneda

MODELO SEGUNDO PARCIAL LAB II – A321 – 2024

Criterios de evaluación

- Sus datos personales deben estar en el nombre de la carpeta principal y la solución: *Apellido.Nombre.SP. Ejemplo: Pérez.Juan.SP*. No se corregirán proyectos sin autor identificable.
- No se corregirán exámenes que no compilen.
- No se corregirán exámenes que posean commits luego de la finalización del parcial.
- No se corregirán exámenes que no estén correctamente publicados en el GitHub.
- Respetar todas las consignas dadas.
- Todas las clases, métodos, atributos, propiedades, etc. Sean nombradas exactamente como fue pedido en el enunciado.
- Se introduzca el código de la función Main sin modificaciones.
- El proyecto no contiene errores de ningún tipo.
- El código compile y se ejecute de manera correcta.
- El código no se encuentre repetido con otro compañero (queda anulado ambos parciales).
- La salida por pantalla será copia fiel de la entregada en este mismo documento.
- Se deberá reutilizar código cada vez que se pueda, aunque no esté explícitamente en el contenido del texto.
- Se deberá documentar el código según las reglas de estilo de la cátedra.

Forma de entrega

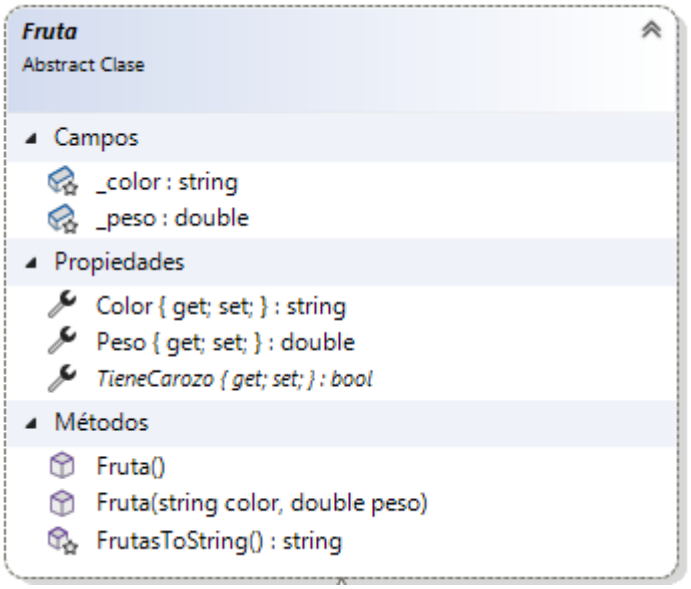
- Se deberá subir al repositorio de GitHub declarado.
- Deberá estar correctamente publicado y accesible para su descarga por cualquier método.

Consigna

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

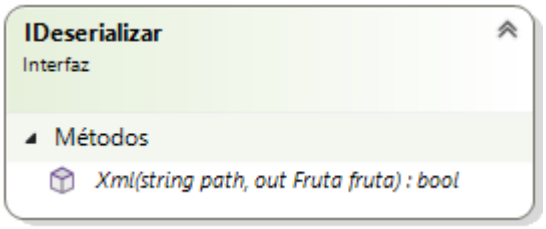
Dentro del proyecto **Archivos** se deberá de respetar el siguiente esquema:

Fruta, abstracta

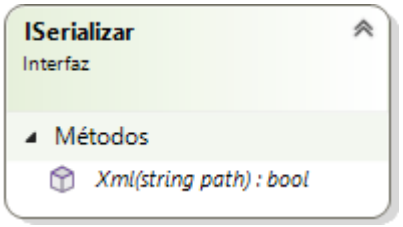


- Posee dos **atributos** propios protegidos.
- La **propiedad** que posee TieneCarozo es abstracta, de lectura y público. El resto de las propiedades son publicas de lectura y escritura.
- Posee un **constructor** público que inicializa sus atributos. Posee un constructor vacio.
- **Método** virtual **FrutasToString** que devolverá un string con toda la información de la clase.

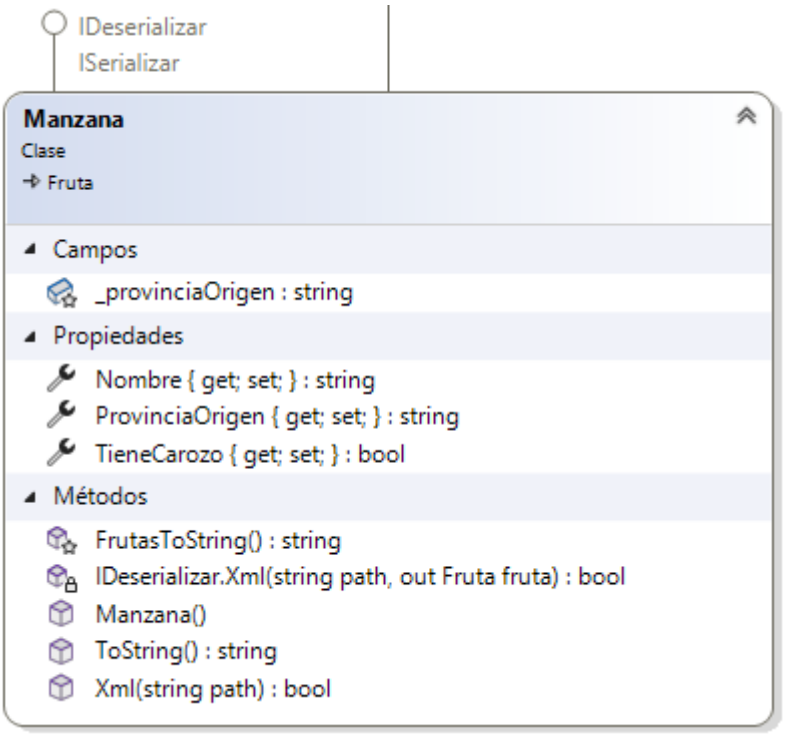
IDeserealizar, interface



ISerealizar, interface

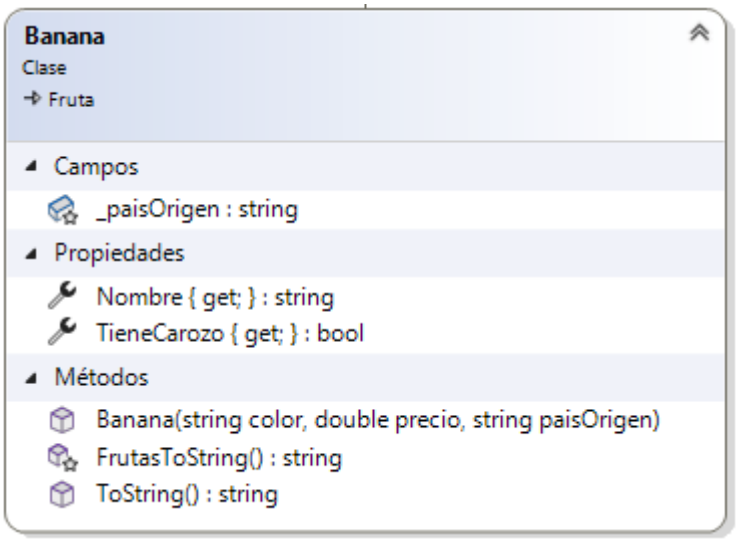


Manzana, hereda de fruta, posee interface IDeserealizar



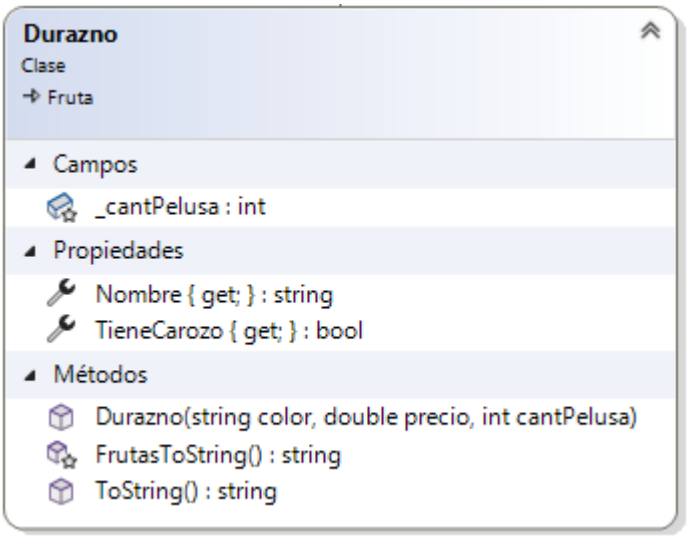
- Posee un **atributo** propio protegido.
- Contiene dos **propiedades** públicas de lectura y escritura.
- **Constructor** que inicializa a su atributo (reutilizar). Posee un constructor vacio.
- Método **XML** de forma explícita, que se encargará de deserealizar el path que se le pase por parámetro y devolverá la manzana deserealizada. Si se serealizo con éxito retorna verdadero, caso contrario, falso.
- **Sobreescritura:**
    - **FrutasToString**, muestra la información de la manzana (reutilizar)
    - **ToString**, reutilizar FrutasToString

Banana, hereda de fruta



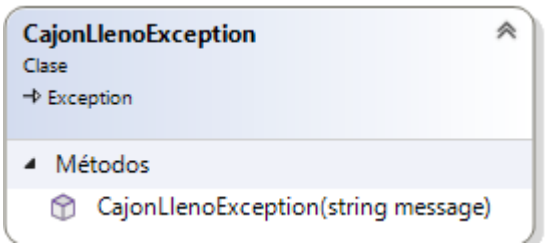
- Posee un **atributo** propio protegido.
- Contiene dos **propiedades** públicas de lectura.
- **Constructor** que inicializa a su atributo (reutilizar).
- **Sobreescritura:**
    - **FrutasToString**, muestra la información de la banana (reutilizar).
    - **ToString**, reutilizar FrutasToString.

Durazno, hereda de fruta

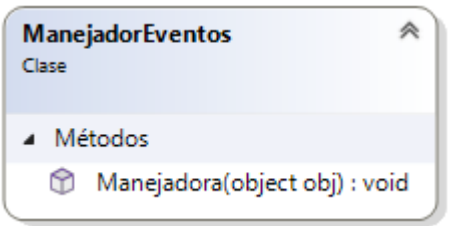


- Posee un **atributo** propio protegido.
- Contiene dos **propiedades** públicas de lectura.
- **Constructor** que inicializa a su atributo (reutilizar)
- **Sobreescritura:**
    - **FrutasToString**, muestra la información del durazno (reutilizar)
    - **ToString**, reutilizar FrutasToString

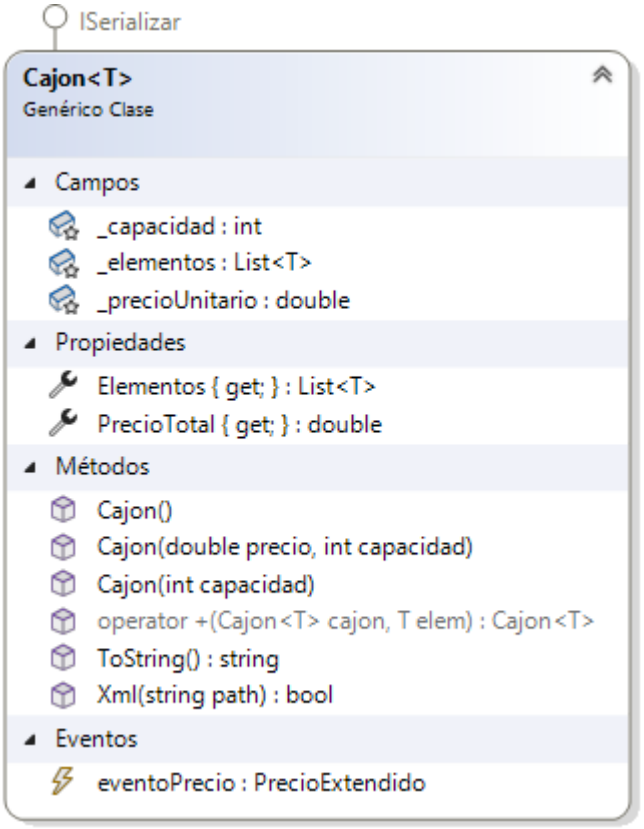
CajonLlenoException



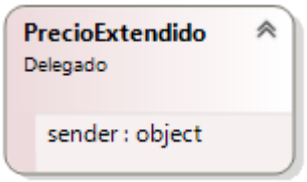
ManejadorEventos



Cajon, generico, posee interface ISerealizar



- Dentro del **namespace** donde se encuentra la clase cajon, contiene un delegado que no retorna nada.



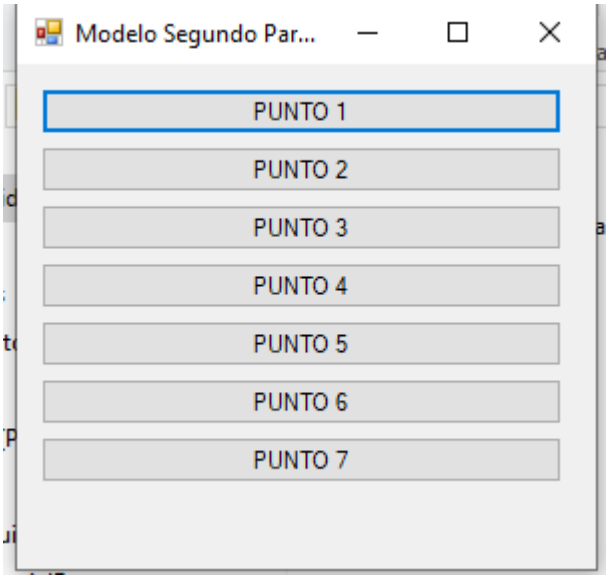
- Posee tres **atributos** propios protegidos.
- Posee un **evento** que llama al delegado.
- Contiene dos **propiedades** de lectura y escritura , Elementos y PrecioTotal, del cual este último hara la multiplicacion del precio por la cantidad de elementos. Si el precio total del cajon supera los 55 pesos, se disparará el evento EventoPrecio.
- **Constructores**: (reutilizar)
    - Vacío que incializa la lista público.
    - Recibe por parámetro capacidad publico.
    - Recibe por parámetro precio y capacidad, es publico.
- El **método** XML se encargará de serealizar el cajón.
- **Sobrecarga**:
    - ToString(), mostrará en formato de tipo string, la capacidad, la cantidad total de elementos, el precio total y el listado de todos los elementos contenidos en el cajón. (reutilizar)

- (+), será el encargado de agregar elementos al cajón, siempre y cuando no supere la capacidad del mismo. Si supera, el mensaje que tiene que mostrar es “El cajón ya se encuentra lleno”.

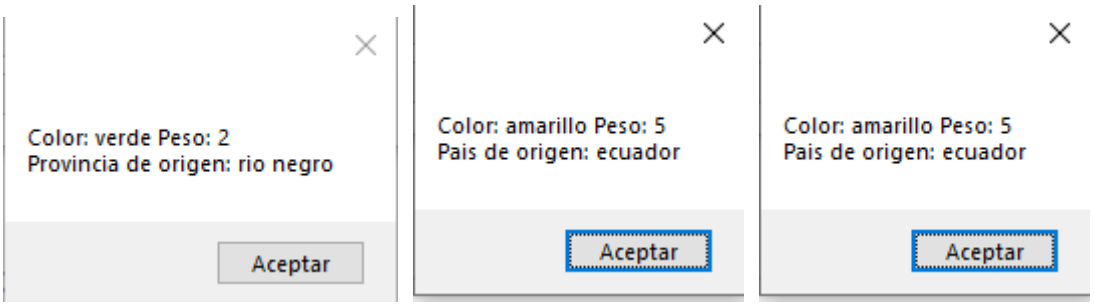
Dentro del proyecto **Formulario** se deberá de respetar las siguientes características:

- El formulario contará con 8 botones. Al presionar los botones se mostraran ciertas características de lo que se fue desarrollando.

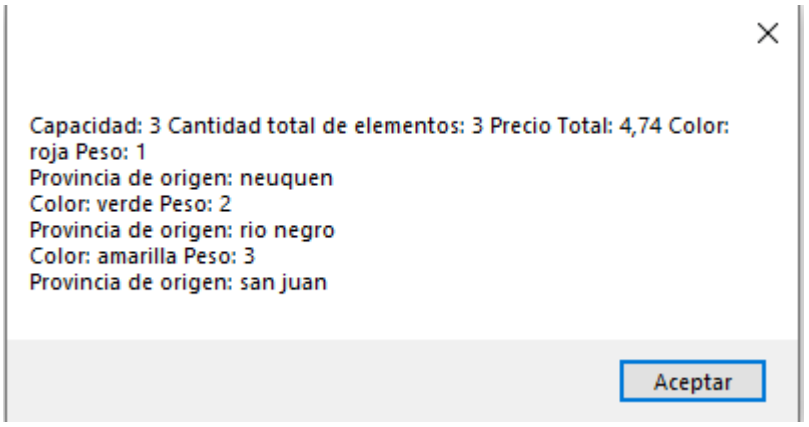
- En los eventos de los botones, se describe la funcionalidad requerida para lograr el objetivo del examen.

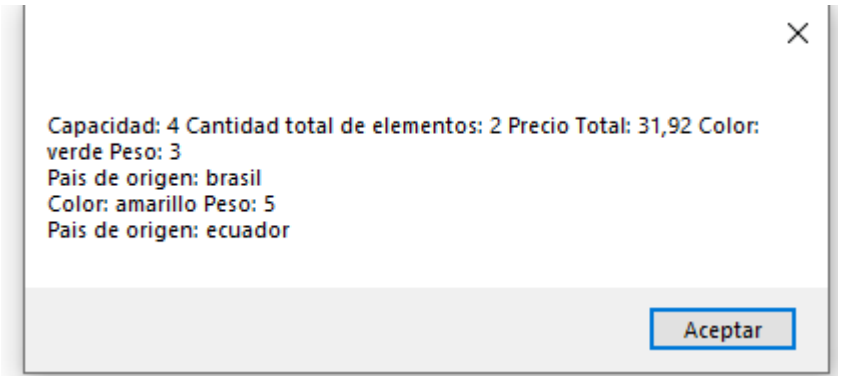


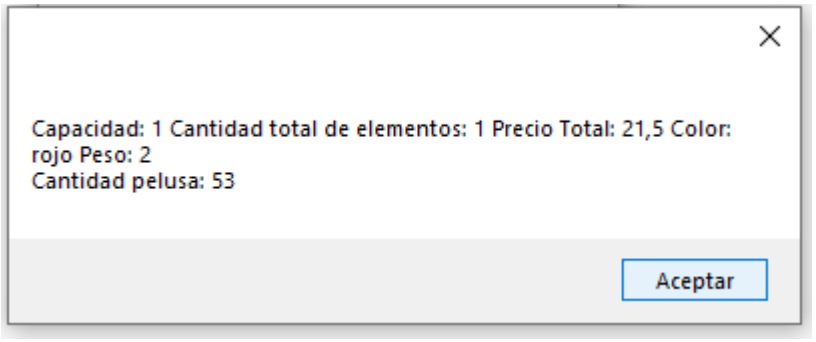
Punto 1:



Punto 2:



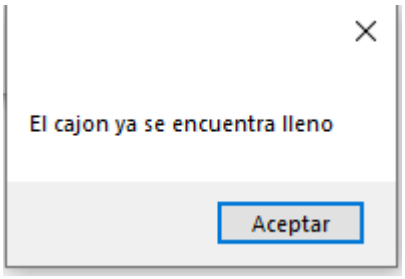




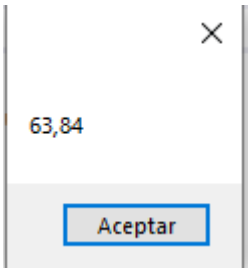
Punto 3:

Se debe serealizar y deserealizar manzana, este ultimo lo muestra por pantalla y se debe serealizar el listado de manzanas.

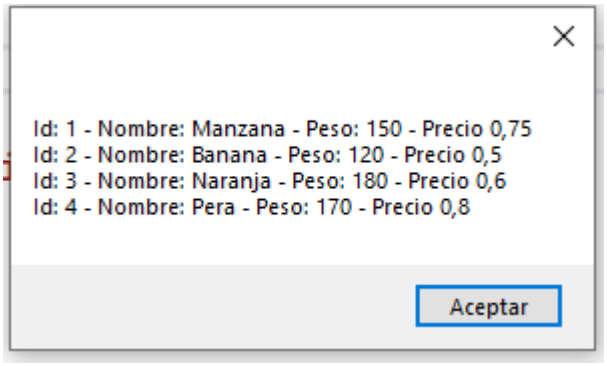
Punto 4:



Punto 5:



Punto 6:



Punto 7:
Se tiene que insertar en la base de datos.

Test Unitario:

- El parcial cuenta con un test unitario del cual se debe validar que pase la prueba correctamente. Además, se debe crear 1 test unitario.