- (1 pto) Realizar un método de extensión, llamado *EsPar*, para la clase Int32, que permita determinar si el número es par o no. Crear otro método de extensión, llamado *EsImpar*, para la clase Int32, que determine si el número es impar o no. Reutilizar código.
- 2) (1 pto) Crear un objeto de tipo Stack<Double>. Apilarle la siguiente secuencia de números: 1, 2, 3. Realizar un algoritmo que permita tener la secuencia ordenada de manera inversa en la misma colección, es decir: 3, 2, 1. Utilizar sólo colecciones de tipo Stack<Double> ó Queue<Double>, de ser necesario.

Facultad Regional Avellaneda										
Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos										
Materia: LABORATORIO II										
Apellido:					Fecha:					
Nombre:					Docente <sup>(2)</sup> :		NEINER			
División:					Nota <sup>(2)</sup> :					
Legajo:					Firma <sup>(2)</sup> :					
Instancia <sup>(1)</sup> :	PP		RPP		SP		RSP		FIN	X

- 3) (2 ptos) Crear dos objetos de tipo Deposito, cada uno de estos objetos contiene un Array de la clase Producto (de hasta 3 elementos). La clase Producto tiene dos atributos: Nombre y Stock.
  Se debe poder sumar los Array de los dos depósitos (con la sobrecarga de un operador en la clase Deposito) y guardar el valor que retorna en un Array de Productos, recordar que si un producto está en los dos Arrays, se debe sumar el stock y no agregar dos veces al mismo producto.
- 4) (1 pto) Sabiendo que estas líneas de código son correctas, que los atributos de las clases son públicos y además que todas las clases poseen un solo constructor, realice los constructores, de cada una de las clases, sabiendo que ProdVendido hereda de ProdExport, que ProdExport hereda de ProdImpuesto y que éste último hereda de Producto.

Producto pro = new Producto("Pala", 22); ProdImpuesto pI = new ProdImpuesto(pro.Nombre, pro.Stock, 600.33); ProdExport pEX = new ProdExport(pI, "Argentina"); ProdVendido p = new ProdVendido (pEX, "Cliente Juan");

- 5) (1 pto) Crear en la clase Comparadora el método bool GetElDelMedio(int uno, int dos, int tres, out int elDelMedio). La funcionalidad de este método de instancia es la siguiente: Retornará un true, si encuentra el valor medio entre los tres parámetros recibidos (y lo alojará en el parámetro de salida elDelMedio). Retornará false si no encuentra un valor medio. Ejemplo 1: uno = 6; dos = 9; tres = 8; => true; elDelMedio = 8. Ejemplo 2: uno = 5; dos = 1; tres = 5; => false; elDelMedio = 0
- 6) (2 ptos) Crear la clase Galpon, que contenga una lista genérica de tipo Deposito, con una propiedad "CantidadDepositos" que sólo permita asignar un valor (entero) al atributo "\_cantidadDepositos". Si el valor que se intenta asignar es cero, se deberá lanzar una excepción de tipo ArgumentException informando de lo acontecido. Si el valor es par (utilizar lo hecho en el punto 1), dejar asignarlo. Si el valor es impar (utilizar lo hecho en el punto 1) escribir en un archivo de texto (log.txt) la fecha (hh:mm:ss y el valor) y asignarlo.
- 7) **(1 pto)** Realizar una estructura try-catch (en el Main) para la escritura de la propiedad del punto anterior que, al capturar la excepción, muestre el mensaje.
- 8) **(1 pto)** Realizar el burbujeo de una excepción **propia**, comenzando en un método de instancia, pasando por un método de estático y capturado por última vez en el Main.
- 9) (2 ptos) Crear la interface IGuardarXML { bool SerializarXML(); }. Implementarla en la clase Galpon.
  Agregar a una instancia de tipo Galpon un Deposito (que contenga al menos un objeto de tipo Producto, otro de tipo Prodlmpuesto, otro de tipo ProdExport y otro de tipo ProdVendido) y generar una serialización XML del galpón.
  Modificando lo que crea conveniente para poder serializar todos los atributos de todos los objetos intervinientes, guardando en el archivo archivo.xml.
- 10) **(2 ptos)** Crear la clase genérica Cosa, utilizando una colección a elección, que exponga los métodos Add y Remove y la propiedad Count y un único constructor por defecto que inicialice la colección.

Nota: dos errores en el mismo punto lo anulan, errores de conceptos de POO anulan el punto. Utilizar proyecto de tipo classLibrary para todas las clases que utilice.