Universidad Tecnológica Nacional
Facultad Regional Avellaneda

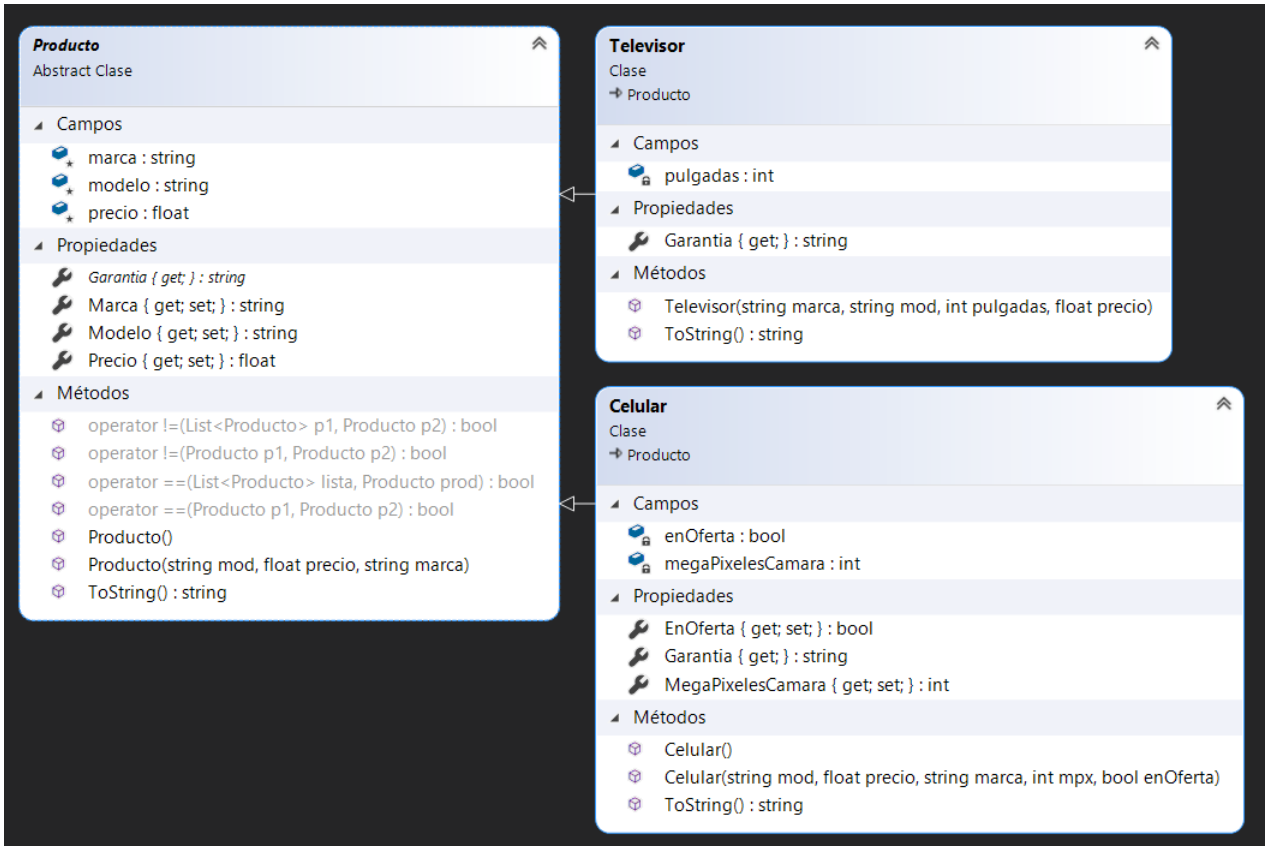
Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos										
Materia: Laboratorio de Programación II										
Apellido:					Fecha:		29-07-2021			
Nombre:					Docente(2):					
División:					Nota(2):					
Legajo:					Firma(2):					
Instancia(1):	**PP**		**RPP**		**SP**		**RSP**		**FIN**	X

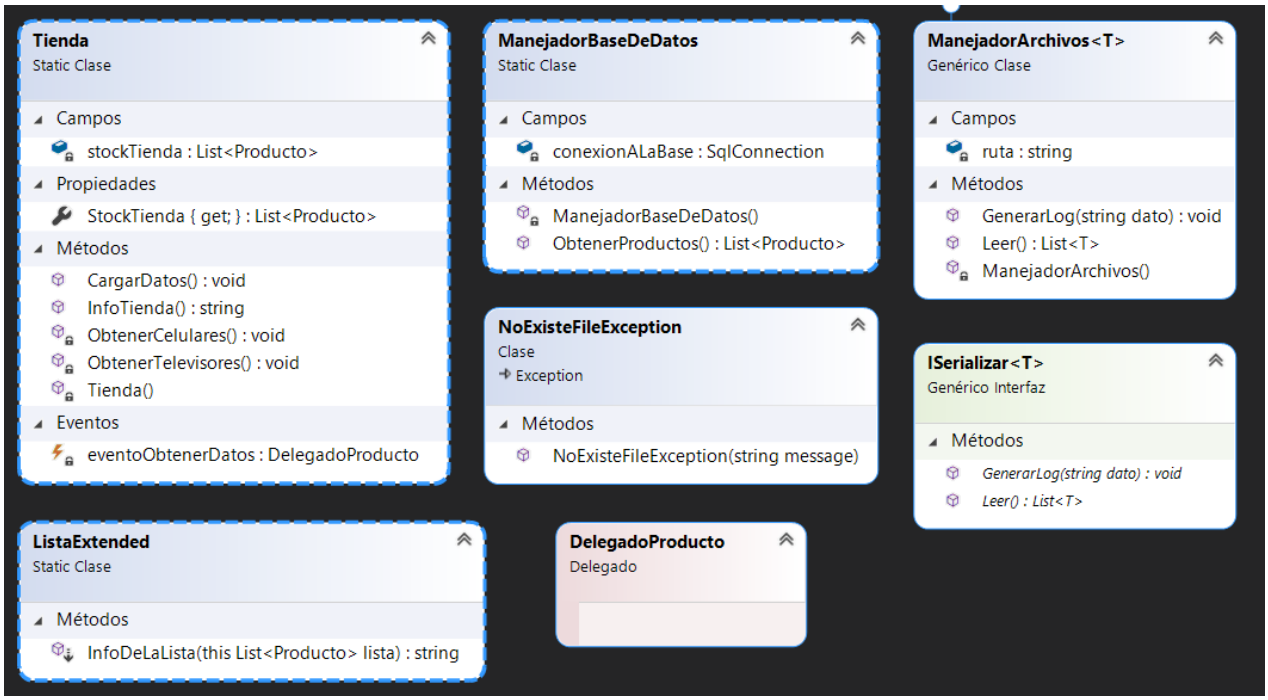
(1) Las instancias validas son: 1er Parcial (**PP**), Recuperatorio 1er Parcial (**RPP**), 2do Parcial (**SP**), Recuperatorio 2do Parcial (**RSP**), Final (**FIN**). Marque con una cruz.
(2) Campos a ser completados por el docente.

IMPORTANTE:

- **2 (dos) errores en el mismo tema anulan su puntaje.**
- La correcta documentación y reglas de estilo de la cátedra serán evaluadas.
- 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.

1. Crear un proyecto del tipo Biblioteca de Clases y colocar el siguiente esquema de clases:


Producto
Abstract Clase
Campos
marca : string
modelo : string
precio : float
Propiedades
Garantia { get; } : string
Marca { get; set; } : string
Modelo { get; set; } : string
Precio { get; set; } : float
Métodos
operator !=(List<Producto> p1, Producto p2) : bool
operator !=(Producto p1, Producto p2) : bool
operator ==(List<Producto> lista, Producto prod) : bool
operator ==(Producto p1, Producto p2) : bool
Producto()
Producto(string mod, float precio, string marca)
ToString() : string
Televisor
Clase
Producto
Campos
pulgadas : int
Propiedades
Garantia { get; } : string
Métodos
Televisor(string marca, string mod, int pulgadas, float precio)
ToString() : string
Celular
Clase
Producto
Campos
enOferta : bool
megaPixelesCamara : int
Propiedades
EnOferta { get; set; } : bool
Garantia { get; } : string
MegaPixelesCamara { get; set; } : int
Métodos
Celular()
Celular(string mod, float precio, string marca, int mpx, bool enOferta)
ToString() : string


Tienda
Static Clase
Campos
stockTienda : List<Producto>
Propiedades
StockTienda { get; } : List<Producto>
Métodos
CargarDatos() : void
InfoTienda() : string
ObtenerCelulares() : void
ObtenerTelevisores() : void
Tienda()
Eventos
eventoObtenerDatos : DelegadoProducto
ManejadorBaseDeDatos
Static Clase
Campos
conexionALaBase : SqlConnection
Métodos
ManejadorBaseDeDatos()
ObtenerProductos() : List<Producto>
ManejadorArchivos<T>
Genérico Clase
Campos
ruta : string
Métodos
GenerarLog(string dato) : void
Leer() : List<T>
ManejadorArchivos()
NoExisteFileException
Clase
Exception
Métodos
NoExisteFileException(string message)
ISerializar<T>
Genérico Interfaz
Métodos
GenerarLog(string dato) : void
Leer() : List<T>
ListaExtended
Static Clase
Métodos
InfoDeLaLista(this List<Producto> lista) : string
DelegadoProducto
Delegado

2. Clase **Producto**:
    a. Producto será abstracta. También deberá **incluir** en la serialización de xml a la clase Celular.
    b. La sobrecarga del operador == que recibe dos productos comparará si un producto es igual a otro comparando por sus marcas,modelos y tipos (utilizar GetType). Devolverá true si tienen esos datos en común.
    c. La sobrecarga del == que recibe una lista y un producto devolverá true si el producto enviado por parámetro está en la lista. Utilizar funcionalidad desarrollada en puntos anteriores.
    d. El método ToString() mostrará por pantalla la información de los atributos del producto. Utilizar StringBuilder.

3. Clase **Televisor**:
    a. Hereda de Producto
    b. La propiedad Garantía devolverá “Garantía de 48 meses” si las pulgadas del televisor son mayores a 40. Si no, devolverá “Garantía de 24 meses”.
    c. El método ToString() utilizará el método ToString de su clase base, y agregará como encabezado “Televisor Smart TV de” y la cantidad de pulgadas que tiene.

4. Clase **Celular**:
    a. La propiedad Garantía devolverá :
        i. “Garantia de 12 meses” si la marca del celular es Noblex.
        ii. “Garantia de 36 meses” si los megapíxeles de la cámara son mayores a 12.
        iii. “Garantia de 24 meses” si no se cumple ninguna de las dos opciones anteriores.
    b. El método ToString() utilizará el método ToString() de su clase base, mostrará también los megapíxeles del celular, y por ultimo agregará como encabezado “Teléfono Celular ”.

5. Clase **ManejadorDeArchivos**
    a. Es genérica.
    b. Tiene un constructor por defecto donde **setea** en el atributo “ruta” para que apunte a MisDocumentos\FinalLabo2021\.
    c. Implementa la interfaz genérica ISerializar<T>
    d. El método GenerarLog recibirá un string que deberá guardarse en un archivo de texto en la ruta ya provista en la clase (MisDocumentos\FinalLabo2021\) llamado “Log_xxxxxxxxx” donde xxxxxxxxx será la fecha de hoy.
    e. El método leer devolverá una lista de tipo genérico, la cual deberá leer de la ruta provista en la clase (MisDocumentos\FinalLabo2021\ ) el archivo “Celulares.xml” ya provisto. Deberá arrojar una excepción personalizada de tipo NoExisteFileException con el mensaje "No existe

el archivo Celulares.xml". Ese texto deberá ser guardado en un archivo de texto, y luego esa excepción deberá ser relanzada.

6. Clase **ManejadorBaseDeDatos**:
    a. Es una clase estática.
    b. El constructor estático instanciará el objeto SqlConnection y seteará la connectionString:
    "Data Source=.; Initial Catalog=Final_1erafecha_2021; Integrated Security=True;"
    c. El método estatico ObtenerProductos() devolverá una lista de tipo producto cargada con toda la información de los televisores que esta en la tabla Televisor en la base de datos.
7. Clase **Tienda**
    a. Es estática.
    b. Tendrá un delegado llamado DelegadoProducto el cual no devolverá ni recibirá parámetros.
    c. Tendrá un evento del tipo DelegadoProducto llamado eventoProducto.
    d. El método CargarDatos permitirá invocar el evento desde otra clase.
    e. En el constructor privado se instanciará la lista stockTienda. Tambien se asignarán al evento los manejadores ObtenerTelevisores y ObtenerCelulares.
    f. El método ObtenerTelevisores deberá hacer un addRange a la lista stockTienda agregándole la lista de productos que se obtiene desde la base de datos.
    g. El método ObtenerCelulares obtendrá los celulares del archivo Celulares.xml, y los agregará a la lista stockTienda. No podrán haber celulares repetidos.
    h. El método InfoTienda deberá mostrar toda la información del stock de la tienda, llamando al método de extensión “InfoDeLaLista”, el cual devolverá un string con toda la información de esa lista.
8. Formulario **Visual**:
    a. Deberá tener un Thread llamado "hilo", que será instanciado en el constructor del formulario. El mismo deberá lanzar el método MostrarOfertas.
    b. En el método Visual_Load se invocará el evento eventoObtenerDatos que está en tienda.
    c. El método MostrarOfertas deberá recorrer el stock de la tienda, y por cada producto que sea un Celular, deberá llamar al método ActualizarCampo y luego esperar 2 segundos antes de volver a llamarlo.
    d. El método ActualizarCampo recibirá el producto enviado por parámetro, y actualizará el richtextbox “rtb_oferta” con la información de ese producto.
    e. El evento FormClosing abortará el hilo.
    f. El evento btn_MostrarOfertas arrancará el hilo.

9. Proyecto **Unit Testing:**

a. Crear tres métodos a elección.

10. **Script Base de datos**

a. Ejecutar el siguiente script

```
USE [master];
CREATE DATABASE [Final_1erafecha_2021];
GO
USE [Final_1erafecha_2021]
CREATE TABLE [dbo].[Televisor](
	[Marca] [nvarchar](50) NOT NULL,
	[Modelo] [nvarchar](50) NOT NULL,
	[Pulgadas] [integer] NOT NULL,
	[Precio] [float] NOT NULL
) ON [PRIMARY]
GO
INSERT [dbo].[Televisor] ([Marca], [Modelo], [Pulgadas], [Precio]) VALUES (N'LG', N'43LM6350PSB ', 43, 47000)
INSERT [dbo].[Televisor] ([Marca], [Modelo], [Pulgadas], [Precio]) VALUES (N'LG', N'50UN7310PSC', 50, 65000)
INSERT [dbo].[Televisor] ([Marca], [Modelo], [Pulgadas], [Precio]) VALUES (N'LG', N'55NANO81UNA', 55, 115000)
INSERT [dbo].[Televisor] ([Marca], [Modelo], [Pulgadas], [Precio]) VALUES (N'SAMSUNG', N'UN50TU7000GCZB',50, 70000)
INSERT [dbo].[Televisor] ([Marca], [Modelo], [Pulgadas], [Precio]) VALUES (N'NOBLEX', N'DM43X7100', 43, 35000)
INSERT [dbo].[Televisor] ([Marca], [Modelo], [Pulgadas], [Precio]) VALUES (N'NOBLEX', N'DJ50X6500', 50, 40000)
```