

Universidad Tecnológica Nacional
Facultad Regional Avellaneda



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

Materia: Laboratorio de Programación II

Apellido:		Fecha:	14-05-2020
Nombre:		Docente ⁽²⁾ :	Dávila-Cerizza-Rodríguez
División:	2°C	Nota ⁽²⁾ :	
Legajo:		Firma ⁽²⁾ :	
Instancia ⁽¹⁾ :	<div style="display: flex; justify-content: space-around;"> PP RPP X SP RSP FIN </div>		

(1) Las instancias validas son: 1^{er} Parcial (**PP**), Recuperatorio 1^{er} Parcial (**RPP**), 2^{do} Parcial (**SP**), Recuperatorio 2^{do} 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 se 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. Se deberá desarrollar un software que facilite al usuario la gestión de una **ferretería**.
2. La solución deberá contener una biblioteca de clases, con toda la lógica desarrollada allí, y un proyecto de Windows Forms.

Biblioteca de clases

La biblioteca de clases deberá contener:

- Enumerados:
 - Se deberá crear un enumerado que especifique el tipo de producto en base a su ámbito recomendado de uso:
 - Profesional
 - Hogar
 - Se deberá crear un enumerado que especifique el tipo de producto en base a su usuario recomendado:
 - Principiante
 - Experimentado
 - Experto

▪ Clase **base** y sus **derivadas**:

- Al menos tres clases **derivadas** que hereden de la clase anterior quienes deberán implementar los miembros abstractos que correspondan, y de ser necesario, sobrescribir los métodos virtuales heredados.
- Las clases deberán recibir en el constructor un tipo **enumerado** que recomiende el **usuario recomendado** y el **ámbito recomendado**.
- Una clase **base** que será abstracta que, además de la información que ustedes crean conveniente, deberá contener:
 - Una propiedad abstracta llamada CalcularMargenGanancias:
En las clases hijas, esta propiedad devolverá la ganancia calculada en teniendo en cuenta el precio del producto.
Los siguientes márgenes de ganancia serán:
 - Clase 1: 10%
 - Clase 2: 13%
 - Clase 3: 16%
 - Un método abstracto llamado InstruccionesDeUso (Devolverá mediante un String Builder, la información de cómo usar ese producto).
 - Sobrecribir el metodo ToString() para mostrar la información del producto.

▪ Clase **Inventario**:

- Deberán de crear una clase más, que será llamada **Inventario** la cual será la encargada de manejar la información del comercio (stock, lista de los productos, agregar nuevos productos, y removerlos por falta de stock).
- Tendrá un operador == buscará por nombre y categoría del producto ingresado por el usuario desde el Form, y mostrará la información si el mismo se encuentra en stock ahora.

▪ Clase **Pedido**:

- Deberá crear esta clase que será quien administre la información del pedido del usuario y, además:
 - La carga de elementos será gestionada a través de sobrecarga del operador +.
 - La eliminación de elementos será gestionada a través de la sobrecarga del operador –.

- Clase Estática Tienda:
 - Deberá crear esta clase que será quien contenga administre el pedido realizado y un historial de pedidos ya confirmados.
 - Deberá contener también un objeto de tipo Inventarlo, para poder llevar registro del mismo.

Windows Forms

Pantalla Login

- En el deberá tener un botón con datos hardcodeados que automáticamente logue el sistema como Usuario, y otro como Administrador para agilizar su corrección.
- Deberá estar centrado y sin posibilidad de maximizar o minimizar la Ventana.

Form principal

Al loguearse como **comprador**:

- Fondo del sistema de color naranja.
- En el título debe mostrar “Bienvenido XXXXXXXXXXXX” donde XXXXXXXXXXXX es el nombre del usuario.
- Deben tener las siguientes opciones:
 - Buscar producto.
 - Mostrar todos los productos: utilizar polimorfismo para mostrar las instrucciones de uso de ese producto.
 - Agregar a carrito: deberá ser un botón con imagen con la imagen de un carrito de compras.
 - Remover del carrito: removerá del carrito un pedido)
 - Realizar compra: Solo podrá habilitarse esta función cuando el carrito tenga al menos un elemento. Al realizar la compra, mostrar en una ventana la información de los productos comprados, con su precio total.

Al loguearse como **vendedor**:

- Fondo color violeta
- En el título deberá decir “Modo Vendedor”
- Deberá tener las siguientes opciones:
 - Deberá haber una opción que permita listar todo el stock completo de la tienda
 - Deberá haber una opción que permita hacer un alta de un nuevo producto al inventario (Cargar los combobox de Ámbito Recomendado y Usuario recomendado utilizando el respectivo enumerado. Investigar Enum.GetValues).

- Deberá haber una opción que muestre por separado:
 - Producto más vendido
 - Facturación total
 - Producto con más stock
 - Producto más rentable (se tiene en cuenta quien más dinero hizo ganar al establecimiento en base a sus ventas y su “margen de ganancia”)

***A tener en cuenta:** Sin importar quien inicie sesión, debe haber una opción para poder hacer una carga automática de productos ya hardcodeados en algún lugar visible, para poder facilitar el testeo de la funcionalidad.

Criterios de evaluación mínimo

Se tendrán en cuenta:

- **La solución deberá compilar sin errores.**
- No puede haber funcionalidad que rompa el programa.
- Uso de validaciones.
- Uso conceptos de POO vistos hasta el momento en clase.
- Uso de las reglas de estilo vistas en clase.
- Deberán utilizarse las buenas prácticas de programación conocidas ya practicadas en la cursada.
- Deberá estar comentado el código (**summary**).
- Los formularios NO pueden tener color por defecto.
- Los nombres de los formularios y sus títulos NO pueden ser el por defecto (Los controles **NO** pueden tener nombre por defecto y **DEBEN** ser descriptivos).

Aquellos parciales que estén en condiciones de aprobar luego de ser revisados por el docente asignado, serán llamados a una etapa de exposición donde mostraran su código, se les harán preguntas sobre el código, sobre el funcionamiento del sistema, y se les hará la devolución pertinente.