

1° Parcial Laboratorio II - 2°D

1°Cuatrimestre 2023

CARNICERÍA

Se nos pide generar una aplicación de escritorio para una carnicería.
La misma deberá contar con las siguientes especificaciones

Se debe diferenciar entre 2 tipos de usuarios:

Vendedor

- Puede vender productos seleccionando a un cliente de la lista de clientes.
- Puede ver los detalles de los productos
- Puede reponer los productos de la heladera principal
- Fijar precios por kilo
- Fijar tipos de cortes de carnes

Cliente

- Compra productos
- Debe especificar el monto máximo de dinero con el que cuenta antes de poder comprar

Debe contar con una pantalla para el inicio de sesión con Mail y contraseña(*hardcodeados para cada usuario*)

Login

- Si sos vendedor, te lleva a un form Heladera
- Si sos cliente, te lleva al form Venta en primero se debe solicitar el monto máximo a gastar

Heladera

- Solo para el Vendedor, se le mostrará un form con la lista de productos disponibles para vender, sus kilos en stock, precios por kilo y ver el detalle.

Venta

- Las ventas se realizarán si se cumplen todas las condiciones (hay stock, el cliente tiene el dinero suficiente)
- Debe permitir buscar por nombre de corte para facilitar el proceso de compra
- Si paga con Crédito tiene un 5% de recargo
- Al efectuar la venta, se deberá mostrar un modal con el detalle de la compra. Si se confirma, se efectúa la venta generando la factura. Si no, se cancela la operación

Condiciones de aprobación

Los 2 proyectos (Biblioteca de clases y Formularios) deben cumplir las reglas de estilo:

- Código correctamente comentado con Summary.
- Nombres de controles con los estilos correspondientes.
- Los formularios no pueden tener ni color ni nombre ni icono por defecto.
- Deben estar presentes las funcionalidades requeridas anteriormente.
- La aplicación debe ser intuitiva y debe además estar distribuida en secciones de fácil navegación y acceso.
- Deben haber al menos 3 formularios diferentes.

Deben ser utilizados todos los temas mencionados a continuación:

- Clases y métodos estáticos
- POO
- Sobrecarga
- Winforms
- Colecciones
- Enumerados
- Encapsulamiento
- Herencia
- Polimorfismo

Se exigirá que:

- No debe generar ningún tipo de error (ni de compilación, ni de ejecución).
- Se deben respetar los 4 pilares de POO.
- Deben existir botones que completen los datos de un cliente o de un vendedor automáticamente para loguear la app (No loguearse automáticamente, sino completar la información en los textbox correspondientes para agilizar el ingreso de información).
- La aplicación deberá tener hardcodeada información previamente para poder probarla sin cargar nada a mano.
- Serán valorados a la hora de evaluar la creatividad que se aplique al programa en cuanto al diseño de la app.
- Mínimo 3 forms
- Reutilizar forms siempre que se pueda

SEGUNDO PARCIAL

Hacer todas las correcciones dadas y agregar los siguientes temas:

- Control de Excepciones (al menos una excepción propia)
- Pruebas Unitarias (al menos 3 test unitarios)
- Tipos Genericos
- Interfaces
- Guardar en archivo txt el historial de venta por cliente o factura.
- Serialización y Deserialización en JSON y XML de Productos(seleccionar uno o varios).
- Implementar CRUD (crear, leer, actualizar y borrar) todo desde base de datos. Y al logearnos contrastar los datos con la base de datos.
- Delegados (al menos uno propio)
- Programación Multihilo y Concurrencia
- Eventos
- Métodos de Extensión (al menos dos)

Formato de entrega

El examen será entregado el día 27 de junio a las 18:30

Por medio de un form de google se pedirá que registren el repositorio de github donde se encontrará ubicado el código. El mismo debe configurarse como privado y agregar los usuarios de los docentes para poder ver el código.

Formato:

Apellido-Nombre-Comision-Año

Ej: *BustosGil-Felipe-2D-2023*

Defensa del parcial

Todos los alumnos deberán realizar la defensa del parcial de forma oral.

Con el fin de probar que los temas están comprendidos y que el desarrollo fue hecho por el/la alumno/a, la instancia de defensa podrá incluir:

- Preguntas técnicas y/o funcionales sobre el desarrollo presentado.
- Modificación de una parte del código en vivo a criterio del docente.
- Preguntas teóricas sobre qué cambios deberían de realizarse si se quisiera contemplar algún otro escenario.

La defensa permitirá:

- Subir,bajar o nivelar la nota del alumno.
- Desaprobación total del mismo (por más que esté correctamente desarrollado) en caso de que el alumn@ no demuestra conocimiento sobre los temas aplicados/desarrollados planteando la duda si fue hecho por otra persona.