

TP3. Proyecto de Heladeria

Alumno: Francisco Allende

División: 2A

Docente: Maximiliano Neiner

El proyecto empieza permite procesar un pedido al venderlo y manejar el Stock desde la app. Esas son las dos opciones principales



El programa comienza con el Form padre cargando listas dinámicas como atributos que se usan a lo largo del programa. Estas listas son deserializadas de un xml si existe el folder y el file, sino, las hardcodea. Esto sucede en el método fill list del FormPrincipañ

Administrar stock

Ya deserializada la lista, o hardcodeada si no existe, accedemos, siguiendo el caso del stock, a un **datagrid con la lista como base de datos de la heladería, en la cual se establece un sistema CRUD.**

La heladera es el stock, siendo una heladera industrial de gran capacidad

- El form de stock permite remover un postre de la heladera
- Añadirlo si fue removido
- Editar su cantidad
- Update de la lista y de la tabla, cualquier operación es de inmediato reflejada en la tabla y guardada tanto en memoria, como en archivo si se decide a guardar
- Ordenar por Id
- Ordenar por mayor a menor cantidad
- Guardar los cambios, ya sea de manera voluntaria o preguntado como recordatorio antes de salir.

	Id	Postre	Sabor	Stock (en kilos)
▶	0	Helado	Chocolate	25
	1	Helado	Chocolate Amargo	10
	2	Helado	Crema Americana	25
	3	Helado	Dulce De Leche	30
	4	Helado	Dulce De Leche Granizado	28
	5	Helado	Frutilla al Agua	30
	6	Helado	Frutilla a la Crema	15
	7	Helado	Maracuya	10
	8	Helado	Sambayon	10
	9	Helado	Tiramisu	12
	10	Helado	Naranja	30
	11	Yogur	Natural	52
	12	Yogur	Frutilla	1
	13	Yogur	Vainilla	8
	14	Yogur	Durazno	5

Agregar

Remover

Editar Cantidad

Ordenar por Id

Ordenar por Cantidad

Guardar

Volver

Salir

Capacidad Total de la Heladera

500

Ocupada

291

Disponibile

209

También calcula el stock total, el espacio disponible en la heladera y el sobrante. Permite advertir si se edita la cantidad por una superior a la capacidad disponible, anulando esa edición y pidiendo por una cantidad de stock más baja (a menos que se libere)

No se pide Id para realizar operaciones. El boton de agregar muestra un formulario modal donde cargar los datos del nuevo postre ,mientras que remove y edit se seleccionan con la flechita del datagrid, solo seleccionandolo.

Stock, al igual que TomarPedido, implementan una interface ITabla con operaciones comunes a tener una tabla con BBDD.

Tomar pedido

Agregar Pedido		Remove		Cobrar		Volver		Salir	
	Id Pedido	Nombre Cliente	DNI	Tipo	Sabor	Cantidad	Direccion	Precio	
Factura								Total	<input type="text"/>

- Este formulario permite tomar un pedido y realizar una venta
- Se remueve al igual que en stock, seleccionando con la flechita
- Se agrega un pedido como en stock, con un formulario modal
- Se acumulan los pedidos de los clientes con el mismo DNI
- Se calcula el total a pagar de ese cliente y luego se cobra y se imprime la factura en un .txt

Correcciones del recuperatorio (Última instancia de reentrega):

- 1. Correcciones hechas, pero siguen habiendo cosas no del todo bien hechas*
- 2. Se usan textBox en lugares donde deberían ser labels o por lo menos no permitir escribir a mano en ellos*
- 3. Falta PDF explicando la funcionalidad del programa*
- 4. Las acciones de editar, eliminar y agregar se hacen engorrosas al tener que ingresar constantemente el id, debería permitir modificar el que está seleccionado.*
- 5. No debería dejar elegir ciertos sabores si no van a estar disponibles para cierta clase de postre*

2) Cómo fue recomendados, se cambiaron los textbox editables por labels no editables

3) PDF hecho con funcionalidad del programa y correcciones

4) Ya no se requiere del Id para realizar ninguna accion del CRUD. Con seleccionarlo es suficiente

5) Se unificaron los sabores. Ahora tanto yogures como helados aceptan los mismos sabores