

Es un programa en el cual se pueden realizar compras de un local que no solo vende Alimentos, sino que también vende Indumentaria, como puede ser el caso de un Walmart. Los formularios serían la interfaz gráfica que tiene un cajero del supermercado para poder realizar las ventas. Tiene un pequeño login en el cual el empleado que esté de turno debe elegir su botón de “Ingresar”. En el momento que llega un cliente con sus productos. El empleado debe ir agregando los productos y su respectiva cantidad al carrito del FormPrincipal. También puede desde los formularios agregar nuevos productos a la lista, y agregar stock a los productos existentes.

Test Unitarios: Generé 3 test unitarios que están en su clase correspondiente. En los summary de cada uno esta detallada su función.

Generics: En un principio usé Generics en el proyecto de Archivos(para la clase XML). Y luego también los utilicé en la clase Compra y la clase llamada ArtículoCompra. Estas dos últimas con la intención de que puedan manejar tanto objetos del tipo Alimento, como del tipo Indumentaria.

Interfaces: Usé Interfaces en la clase Xml del proyecto Archivos.

Archivos: Generé 2 clases, una encargada de los archivos de texto, y otra encargada de serialización y deserialización de archivos Xml.

Serializacion: Utilicé de deserialización para poder levantar la lista de empleados del Walmart. Siempre al inicio del programa.

SQL y Bases de datos: Usé base de datos para almacenar la lista de alimentos y la lista de indumentaria. Cuando se inicia el programa se levantan las listas automáticamente de la base de datos llamando al método Walmart.CargarDatos(). Cree una clase estática llamada ProductosDB donde tengo los métodos relacionados con la base de datos, encargados de mantener la Base de datos actualizada en todo momento: cada vez que se vende un producto, o se agrega stock, se agrega algún nuevo producto, etc.

Hilos: Usé hilos para manejar los Formularios “secundarios”, los formularios donde el empleado puede agregar nuevos productos o sumar stock. Para que pueda seguir haciendo ventas mientras esta realizando dichas cargas en caso de que sea necesario. Y los datagridview del formPrincipal se actualizan en tiempo real.

Eventos y Delegados: Cree un delegado y un evento para realizar el “eventoVenta”, esta declarado en el FormPrincipal, contiene 3 manejadores encargados de realizar la venta, lo cual consiste en actualizar los stocks de los productos salientes y la generación del ticket.

Métodos de extensión: Usé métodos de extensión para generar métodos encargados del manejo de listas: buscar objetos dentro de la lista, ya sea por su Id o Descripción.

Excepciones: Generé unas excepciones para la clase que maneja los archivos de texto y xml.