

Aplicación BookCloud, es una aplicación orientada a la gestión detrás de mostrador de una tienda de libros. Permite gestionar alta y baja de clientes, vender publicaciones a estos y tomar un registro de las ventas concretadas.

Inicia con un form principal, con formato menú de 4 opciones:

- .Gestión Clientes: Contiene un listado de clientes, al oprimir cada uno muestra el detalle, y da la opción de eliminar un cliente del listado o agregar uno nuevo a este.

- . Gestor Venta: Contiene 2 listbox con información de clientes Regulares (máximo de 10 productos por compra) y Premium (límite de 20 productos y descuento sobre el total del 15%), se puede seleccionar un cliente y agregar o eliminar productos del carrito, si se confirma la venta, se agrega al listado de ventas.

- . Registro de ventas: Abre un listbox mostrando el registro de ventas al momento

- . Soporte técnico: Abre una ventana pidiendo la consulta o reclamo, y esta se almacena en un archivo de quejas, este siempre almacena la última queja.

La idea sería simular un gestor de quejas, donde la queja es enviada a un mail o contacto

Con soporte técnico. Por esto se almacena solamente la última, ya que apenas llega se manda.

//-----

Aplicación de temas:

.Generics:

Se usa en la clase Carrito, dentro de la biblioteca de clases BookClud\_Entidades, dentro de la carpeta gestores, esta la hice genérica y aunque no se usa de forma genérica en el proyecto, fue creada con la intención de mostrar que se entendió como usarla y también pensada para reutilizarla en otros proyectos. La aplico en la clase Cliente\_BookCloud, creando un carrito de Publicaciones.

También se emplea generics, en las clases Serializadoras de archivos Xml y Json, para serializar y leer cualquier tipo de objeto o colección de objetos. Se emplea al guardar el listado de clientes y publicaciones (xml) y ventas (json) en el formulario principal.

Interfaces:

. IComun: Se emplea en, Usuario , Publicacion, Carrito, Venta.

.ISerializados: Se emplea en, Serializado\_Json , Serializado\_Xml

.IComunes\_Forms: Se emplean en, Form\_GeneradorDeVenta, Form\_GestionClientes,  
Form\_SeleccionDeCliente, FormAltaCliente.

Archivos y Serialización: Se emplean al momento de Serializar y Deserializar la data de:  
Clientes(xml),Publicaciones(xml),Ventas(json) y Quejas(.txt).

.txt se aplica en el form Soporte\_Tecnico, se almacena solamente la última  
queja. Puede verificarse entrando al archivo: Reclamos\_File.txt en la carpeta  
debbbug del proyecto de windows forms.

.xml se aplica en el Form principal, para serializar y leer clientes, estos  
archivos se almacenan en la carpeta debbug del proyecto de consola,  
puede verificarse la funcionalidad, cortando la ejecución del proyecto y  
volviéndola a abrir, así se corrobora que se haya mantenido los clientes  
agregados,eliminados,registro de ventas etc.

La actualización de archivos es automática, al cerrar la app por completo se  
actualizan automáticamente todos los archivos.

Excepciones: Se utilizan para controlar el alta de clientes y controlar diferentes acciones que puedan  
poner en peligro la ejecucion del programa.

TestUnitarios: Se prueban las clases de clientes,publicaciones y ventas.