

// A TENER EN CUENTA:

. Dentro de la carpeta del tp, se encuentra el script de la base de datos, de ser necesario modificar el string de conexión a la base de datos (en mi pc, funciona con el SERVER: localhost) ir viendo si funciona con . O con localhost SQLEXPRESS.

. De preferencia ejecutar visual studio como administrador, ya que con ciertos antivirus si no se da permiso de escritura de archivos tira excepción Permision Denied (La excepción se maneja, pero puede limitar funcionalidades) ya depende del antivirus usado.

//-----EXPLICACION BREVE APP:

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 y reclamos. Inicia con un form principal, con formato menú de 5 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. También se almacena en la base de datos de reclamos pendientes.

. Gestor de Reclamos: Trae reclamos pendientes de base de datos y permite solucionarlos (eliminarlos de la base)

//-----

//-----EXPLICACION TP4-----

Actualizaciones:

. Se agrego la opcion Gestionar Reclamos en el menu principal (USO DE BASE DE DATOS):

-Esta opcion se relaciona con el guardado de reclamos en .txt del tp3, dicho archivo simulaba el envio de un mail con el reclamo a un moderador de la app, siguiendo con este hilo, todos los reclamos ademas de ser “mandados via mail a un moderador”, son almacenados en una base de datos. La opcion del menu Gestionar reclamos, lee dicha base de datos, trae los reclamos pendientes y permite la opcion de “Liberar” reclamo, simulando la gestion a realizar una vez resuelta la consulta o el reclamo. Tambien al precionar el boton de Servicio tecnico, ademas de mantener el envio del mail (guardado .txt), dicho reclamo es almacenado en la base de datos como reclamo pendiente.

. Uso de EVENTOS:

-Se implementa eventos, en la clase Carrito, anula el boton agregar producto cuando se alcanza el maximo de productos del carrito (cliente premium 20 libros, regulares 10 productos) e informa en un label debajo del listbox de productos del carrito, que se llevo al limite, si se elimina un producto, se vacia el carrito o se concreta la venta, se reanuda el boton y el label vuelve a su default:

. El delegado asociado se encuentra en el pie de la clase, abajo del namespace

. El evento se encuentra como atributo de la clase Carrito y es lanzado en el metodo AgregarProducto()

. El manejador se encuentra en el form Generador_Venta y su metodo manejador es DeshabilitarBoton_AgregarAlCarrito

. Uso de Delegados:

-Se implementa en el form principal, al cargar los listados de ventas, clientes y publicaciones, trabaja en conjunto con metodos asincrónicos e hilos para cargar los listados sin frenar la ejecucion normal de la aplicacion. El delegado carga los 3 metodos para cada listado y los ejecuta.

. El delegado se llama TraerListas_Delegate y se declara justo debajo del namespace del form

. Se lo invoca en el metodo Load del form Principal.

. Se le asocian los delegados en el metodo Load del form Principal.

. Uso de Hilos:

-Se implementan en el form principal, se crean 3 metodos asincrónicos, los cuales invocan dentro a la clase LongTask y a su respectivo metodo que trae la data de los archivos, estos tambien trabajan de forma asincronica para lograr traer la informacion sin frenar la app. Estos metodos de la clase estatica LongTask (en la carpeta Gestores), poseen un Thread.Sleep() para simular una carga pesada de datos.

. Uso de Expresiones Lambda:

-Se usan dentro de los metodos de la clase LongTask, para respetar la invocacion del metodo Task.Run().

. Uso de Metodos de Extension:

- Se implementa en la clase nueva Validadores_Extension (en la carpeta Gestores), esta extiende la funcionalidad de la clase Validadores, con un metodo que retorna la edad a partir de un objeto de tipo DateTime.

//-----EXPLICACION TP3 (Vieja)-----

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 debug del proyecto de windows forms.

.xml se aplica en el Form principal, para serializar y leer clientes, estos archivos se almacenan en la carpeta debug 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 ejecución del programa.

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