Trabajo Práctico Final

Alumno: Guido Cesa.
División: 2do D.

La idea de este programa está centrada en el manejo de un depósito de una fábrica. Se puede agregar inventario al depósito como realizar un pedido al mismo.
Para agregar ítems se deberá hacer click en “Add new parts” y luego seleccionar de los diferentes menús las características de los mismos y la cantidad, y luego el cambio se verá reflejado en un listado que aparecerá en pantalla. Dicho listado muestra los Id, que se pueden interpretar de la sig manera: La primer letra es el tipo de parte, lo que le sigue a la L es el largo (Length), lo que sigue a la D es el diámetro(Diameter), a la T dientes en el caso de engranajes(Teeth/Cogs) y el tipo de tornillo se encuentra impreso entero.
En caso de querer realizar un pedido, al hacer click en el botón “Make Request”, se abrirá una ventana con el stock actual y una columna con un dropdown para seleccionar la cantidad a pedir. Mientras se realiza el pedido se abrirá una animación, y quedará inhabilitado el botón de “Make Request” hasta que esta finalice. En caso de no haber stock se avisará por pantalla.
Luego estan los botones de carga y guardado, tanto desde un XML como de una base de datos. Ambos informaran si los datos fueron modificados con exito.

Temas requeridos en el TP:

-Excepciones: Se utilizan excepciones a la hora de cargar y guardar archivos y la hora de recibir pedidos de más partes de las que hay stock. Esto se puede ver en los métodos RecieveRequest, Load y Save de Warehouse, en la clase XML y en los métodos de los botones de Load y Save en el Form. En caso que se lancen estas excepciones, se informará por pantalla cual fue el problema.

Test Unitarios: Se realizo un proyecto a parte con algunos test unitarios que prueban algunas de las funcionalidades basicas internas del programa.

Tipos Genericos: Se utilizo tipos genéricos en la clase Entry y XML, y en el método no utilizado RequestStock. En la clase entry fue necesario el uso de genéricos para poder serializar el inventario ya que no supe resolver la serialización de mi lista de objetos como objetos en si. Luego el método request stock se pensaba utilizarlo con una interfaz a parte para realizar pedidos haciendo click en objetos del inventario pero no se pudo resolver por falta de tiempo.

Interfaces: Se utiliza la interfaz IStockeable para todos las clases que vayan a ser guardadas por el warehouse, asi se garantiza que cuenten con los métodos necesarios para realizar las operaciones.

Archivos y serialización: Esto se encuentra implementado en la clase XML, aunque se necesitó de una clase auxiliar Entry por el problema ya mencionado, el programa guarda y

carga el warehouse desde un archivo, además también guarda todos los pedidos al warehouse exitosos también como archivos con la fecha y hora de realización del mismo.

SQL: Se implemento a una conexión a una base de datos para guardar y cargar el Warehouse actual.

Threads: El uso de threads se puede ver al correr la animación ya que se puede seguir utilizando el resto del programa mientras está corre.

Eventos y delegados: Se utilizaron para hacer funcionar correctamente la animación.

Métodos de extensión: Se pasó el método que convierte de String a CarPart como una extensión de String.