

Trabajo Práctico 4

Ejércitos en el mundo de espada negra

Esta aplicación es un ABM de ejércitos en un mundo ficticio, con la posibilidad de filtrar los ejércitos con alguno de los criterios dados y guardar los cambios en archivos de extensión .csv, .xml y/o .json.

Siempre que se abra la aplicación está la opción de importar cambios ya guardados y trabajar con ellos o trabajar de cero sin importar nada.

Excepciones: En el proyecto Entidades se encuentra la clase EjercitoRepetidoException, implementada en la clase FrmAgregarEjercito del proyecto Formularios, en la línea 85. Los try catch son fáciles de encontrarlos en el código, ya que hay bastantes, pero para nombrar algún ejemplo en la clase Archivo de la línea 26 a 62 hay un try con sus catch correspondientes.

Pruebas unitarias: En la clase Ejercito_Test del proyecto TestsUnitarios hay dos métodos que testean el alta y baja del ejército.

Generics: Clase Ejercito y Pueblo

Con generics, en mi caso facilito la entrada de datos mediante el formulario gracias al enumerado y para exportar e importar datos usando el string. Ejemplo de Implementación en Archivo línea 213.

Interfaces: Tanto la clase Ejercito como Milicia contienen la interfaz IFuerzaMilitar. En el proyecto de consola Test, en el program, línea 17 a 21 se ve la implementación.

Archivos y serialización: En la clase Archivo del proyecto Entidades se puede ver la importación y exportación en archivos .csv, .xml y .json. Los datos en los archivos .csv están separados por punto y coma.

SQL: Uso la base de datos de manera “automática”, o sea, cuando abro la aplicación, importo los datos, cuando agrego un ejército inserto los datos, cuando borro un ejército también lo elimino de la base de datos y cuando hago alguna modificación también la hago en la base de datos. En la clase BaseDeDatos se puede ver como hice la conexión a SQL y como hice los SELECT, INSERT, UPDATE, DELETE.

Delegados: En la clase Ejercito, línea 7 se ve la declaración del delegado DelegadoExportacion quién es usado en la línea 18 por un evento.

Expresiones lambda: Su utilización se puede ver, por ejemplo, en la clase ListaEjercitoExtension, en la línea 21 o en la clase FrmExportar, línea 73.

Hilos: Con la implementación de hilos, logro exportar los 3 archivos al mismo tiempo con la información que se ve pulsando el botón 'Mostrar'. Su implementación está en la clase FrmExportar, empezando de la línea 73 hasta la 81.

Eventos: Uso eventos para notificar cuando se agrega un ejército, y a la hora de exportarlos, con el botón 'Ver cambios sin guardar' se pueden ver todos los ejércitos añadidos que no fueron exportados.
La declaración del evento está en la clase Ejercito, línea 18.
Y en la clase FrmAgregarEjercito, línea 116, añadido el metodo MostrarEjercito.

Métodos de extensión: Arme la clase ListaEjercitoExtension donde están todos los métodos de extensión para facilitar el acceso al filtrado de datos.
En la clase Filtro, línea 20 se puede ver la llamada a uno de los métodos.

Pruebas unitarias: En la clase Filtro_Test del proyecto TestsUnitarios2 hay tres métodos que testean filtros de listas.