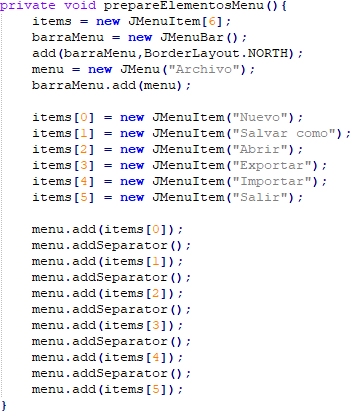
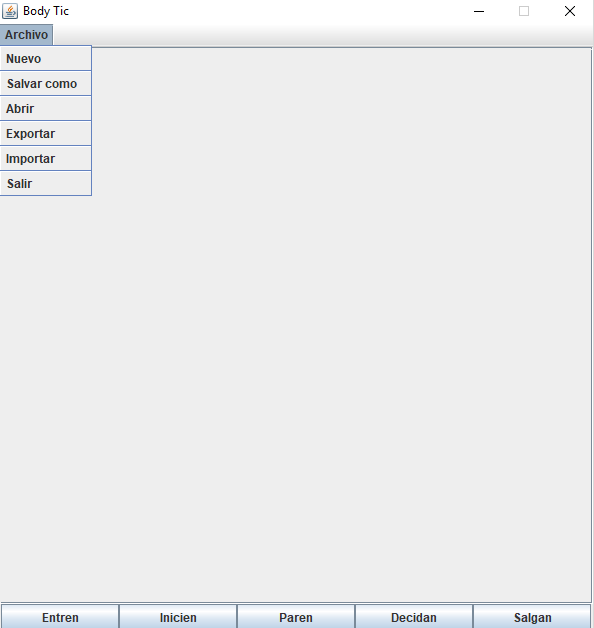
**Creando la maqueta**

**[En lab06.doc, \*.asta y \*.java] [NO OLVIDEN BDD y MDD]**

**En este punto vamos a construir la maqueta correspondiente a esta extensión siguiendo el patrón MVC.**

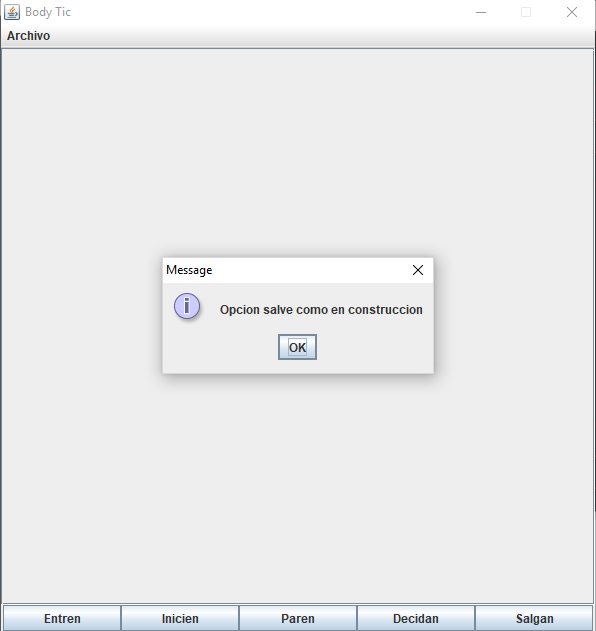
1. **MODELO: Preparen los métodos correspondientes a reiniciar y a las cuatro opciones básicas de entrada-salida (salve como, abra, exporte como, importe). Los métodos deben simplemente propagar una bodyTIcExcepcion con el mensaje: “Opción … en construcción”. Los métodos de entrada salida deben tener un parámetro File.**
2. **VISTA : Construyan un menú barra que ofrezca, además de las opciones básicas de entrada-salida, las opciones estándar de iniciar y salir. Para esto creen el método prepareElementosMenu. Capturen la pantalla correspondiente.**





1. **CONTROLADOR: Construyan los controladores correspondientes a estas acciones. Para esto creen el método prepareAccionesMenu y los métodos base del controlador. Estos últimos métodos, por ahora, sólo deben llaman directamente el método correspondiente de la capa de aplicación (archivo nulo). Capturen una pantalla significativa.**



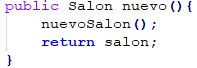
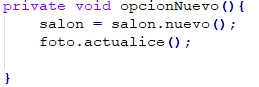


**Implementando salir e iniciar**

**[En lab06.doc, \*.asta y \*.java] [NO OLVIDEN BDD y MDD]**

**Las opciones salir e iniciar van a ofrecer los dos servicios estándar de las aplicaciones. El primero no requiere ir a capa de aplicación y el segundo sí.**

1. **Construyan el método opcionSalir que hace que se termine la aplicación. No es necesario incluir confirmación.**
2. **Construyan el método opcionIniciar que crea un nuevo bodyTIc. Capturen una pantalla significativa.**

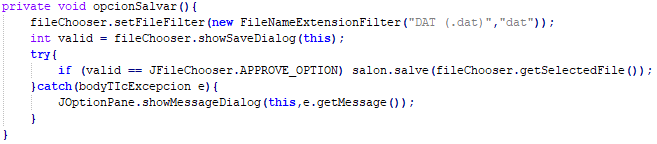


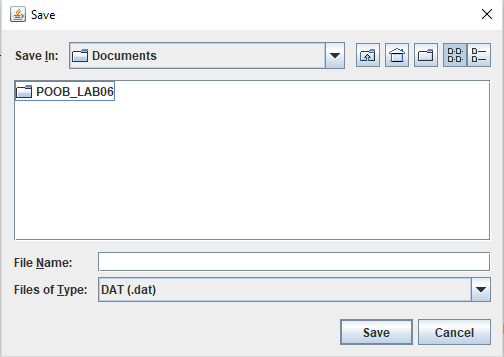
**Implementando salvar y abrir**

**[En lab06.doc, \*.asta y \*.java] [NO OLVIDEN BDD y MDD]**

**Las opciones salvar y abrir van a ofrecer servicios de persistencia de un bodyTIc como objeto. Los nombres de los archivos deben tener como apellido .dat**

1. **Construyan el método opcionSalvar que une de forma adecuada la capa de presentación con la capa de aplicación. Usen un FileChooser y atiendan la excepción. Ejecuten la aplicación probando las diferentes opciones del FileChooser y capturen una pantalla significativa.**

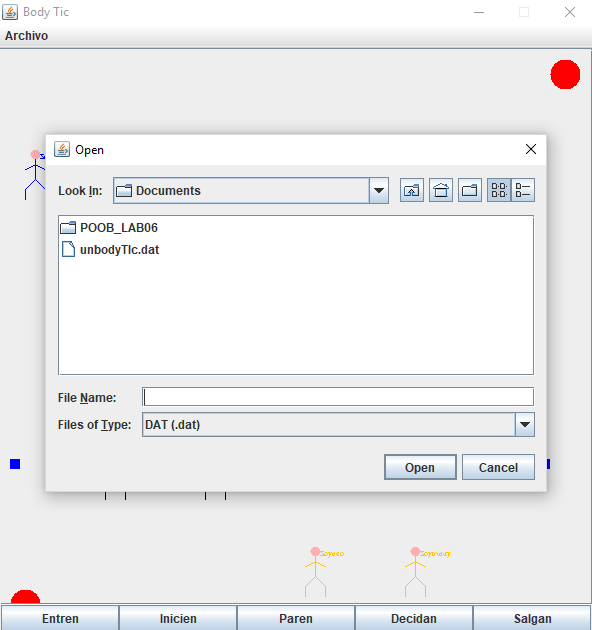


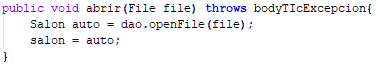


1. **Construyan el método salve que ofrece el servicio de guardar en un archivo el estado actual del bodyTIc.**
2. **Validen este método guardando el bodyTIc inicial después de dos clics como unbodyTIc.dat. ¿El archivo se creó en el disco? ¿Cuánto espacio ocupa?**

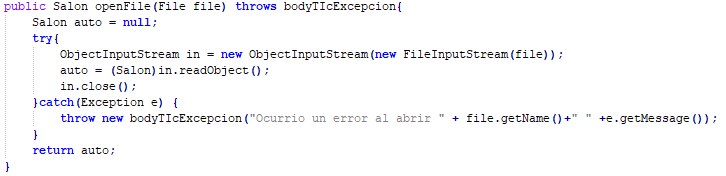
El documento se guardó en la dirección C:\Users\2152972\Documents y ocupa 2 KB de espacio.

1. **Construyan el método opcionAbrir que une de forma adecuada la capa de presentación con la capa de aplicación. Ejecuten la aplicación probando las diferentes opciones del FileChooser y capturen una pantalla significativa.**

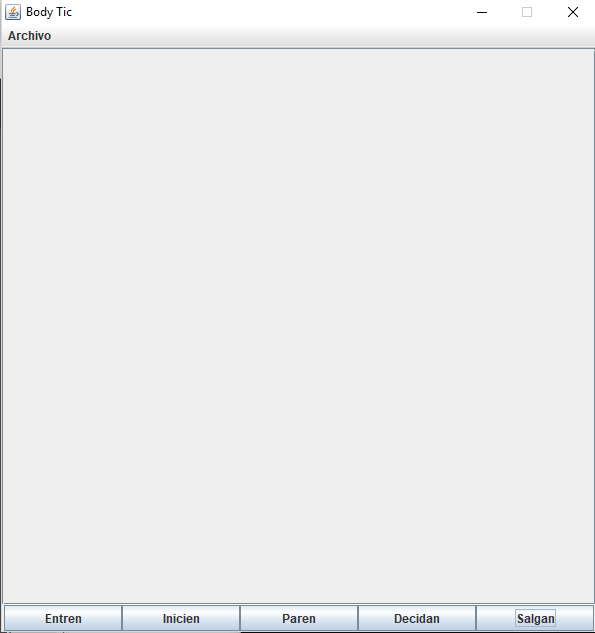


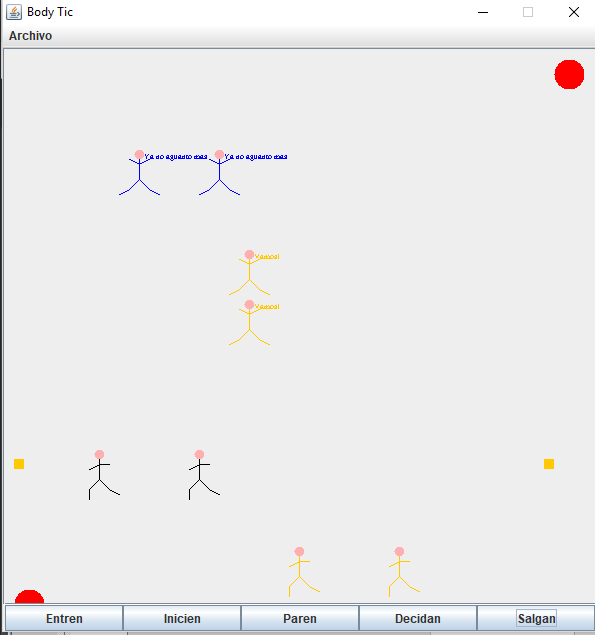
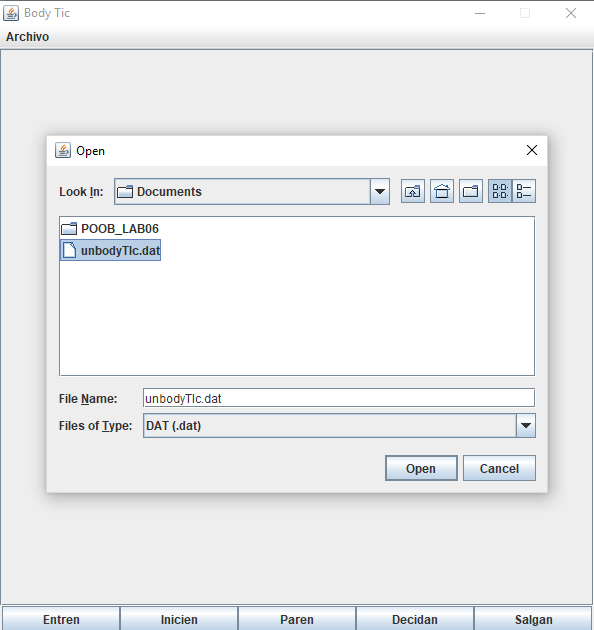


1. **Construyan el método abra que ofrece el servicio de leer un bodyTIc de un archivo. Por ahora para las excepciones sólo consideren un mensaje de error general.**



1. **Realicen una prueba de aceptación para este método iniciando la aplicación, creando una nueva situación en el bodyTIc y abriendo el archivo unbodyTIc.dat. Capturen imágenes significativas de estos resultados.**



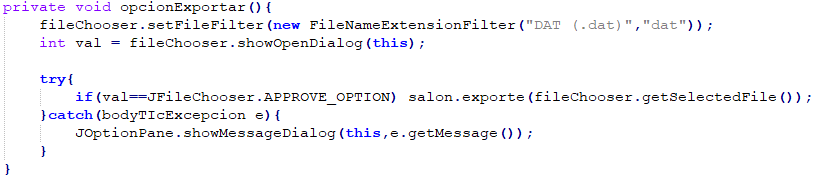


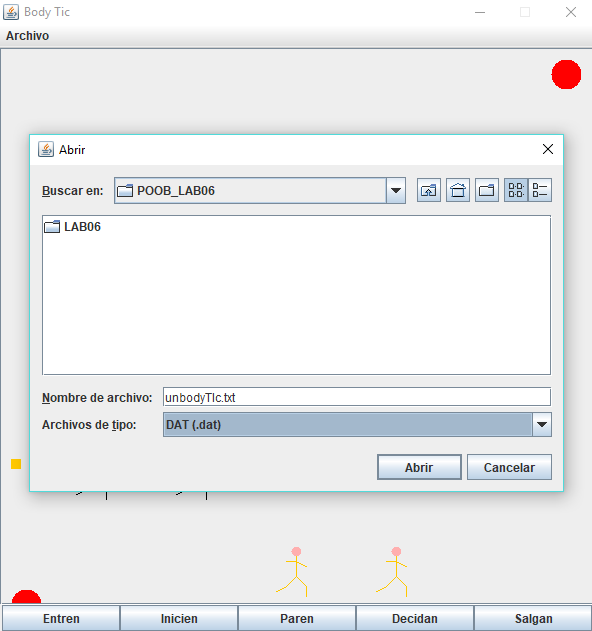
**Implementando importar y exportar**

**[En lab06.doc, \*.asta y \*.java] [NO OLVIDEN BDD y MDD]**

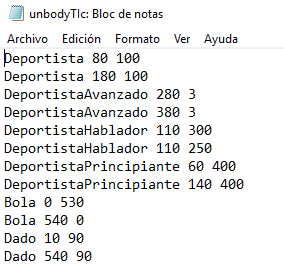
**Estas operaciones nos van a permitir importar información de un bodyTIc desde un archivo de texto y exportarlo. Los nombres de los archivos de texto deben tener como apellido .txt Los archivos texto tienen una línea de texto por cada elemento En cada línea asociada un elemento se especifica el tipo y la posición. Avanzado 20 20 neo Bola 50 50**

1. **Construyan el método opcionExportar que une de forma adecuada la capa de presentación con la capa de aplicación. Ejecuten la aplicación y capturen una pantalla significativa.**



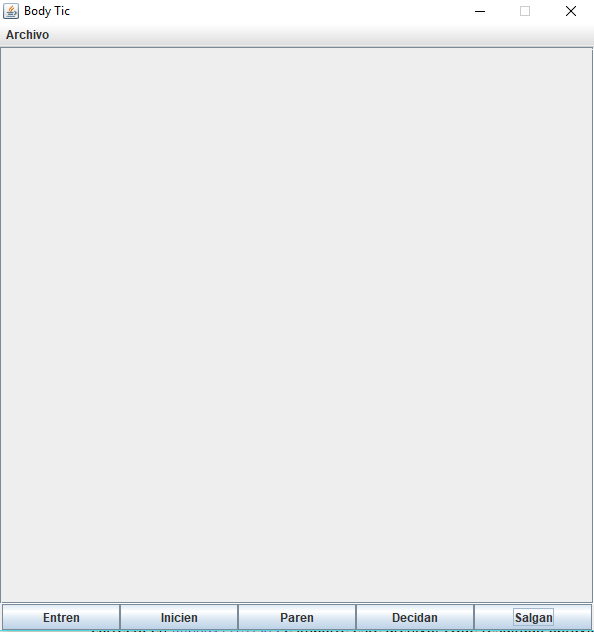


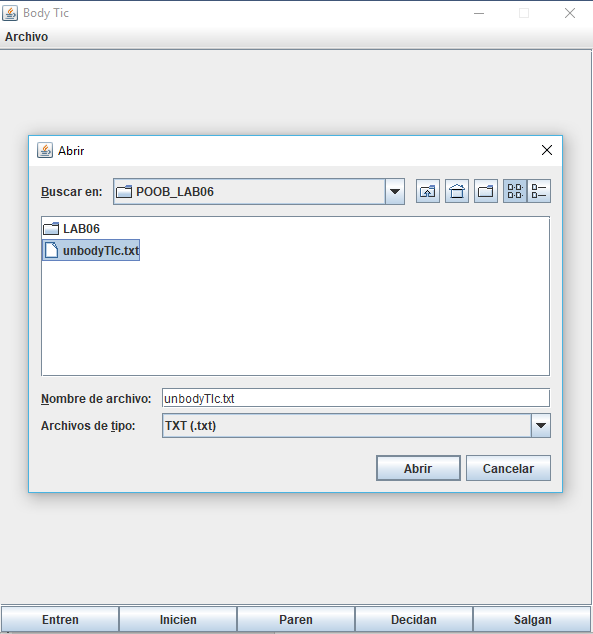
1. **Construyan el método exporte que ofrece el servicio de exportar a un archivo texto, con el formato definido, el estado del bodyTIc actual.**
2. **Realicen una prueba de aceptación de este método: iniciando la aplicación y exportando como unbodyTIc.txt. Editen el archivo y analicen los resultados. ¿Qué pasó?**

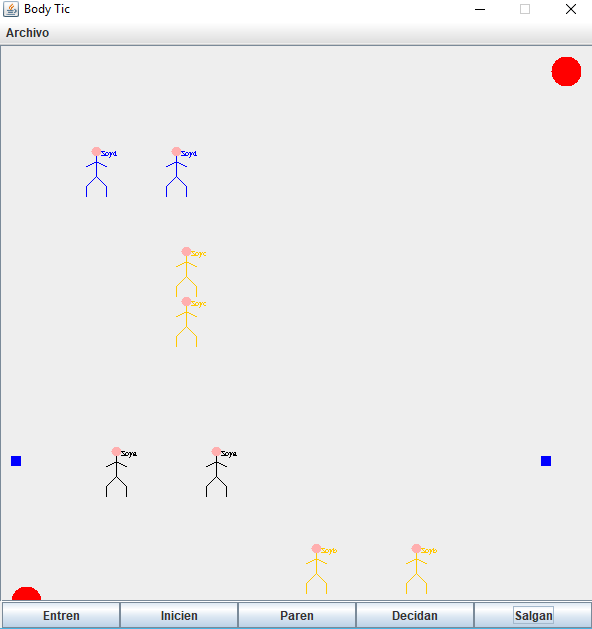


Se guardo el tipo del elemento que se encuentra en dentro del salón con su posición en la coordenada x y la coordenada y.

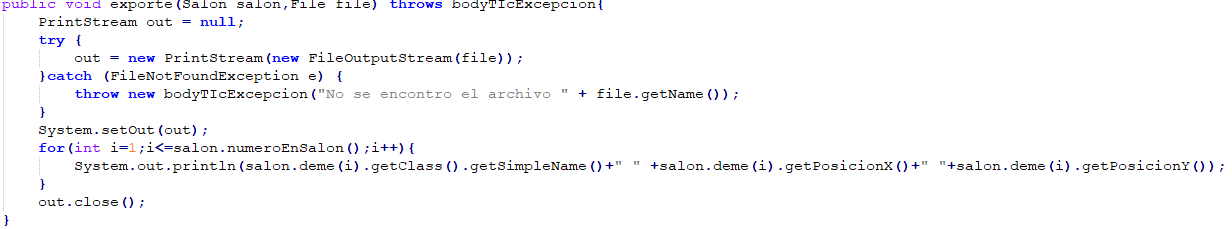
1. **Construyan el método opcionImportar que une de forma adecuada la capa de presentación con la capa de aplicación. Ejecuten la aplicación y capturen una pantalla significativa.**

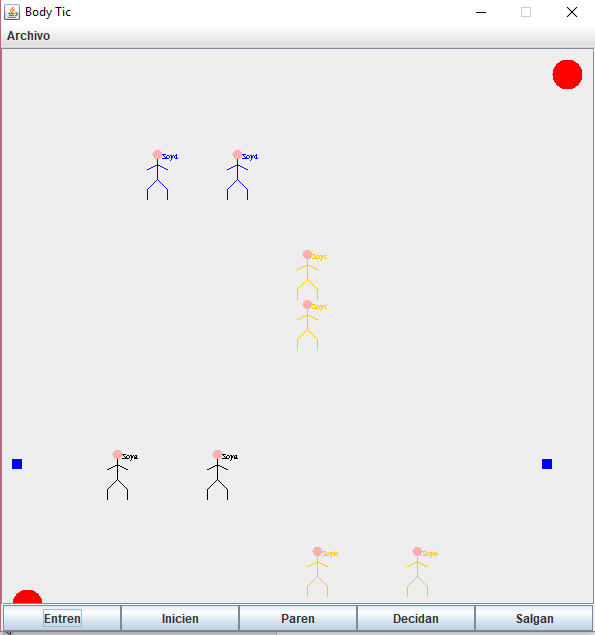






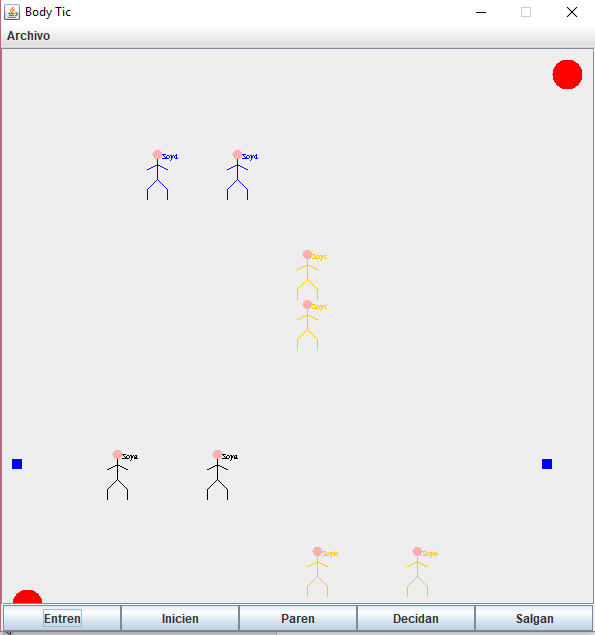
1. **Construyan el método importe que ofrece el servicio de importar de un archivo texto con el formato definido. Por ahora sólo considere un mensaje de error general. (Consulten en la clase String los métodos trim y split)**





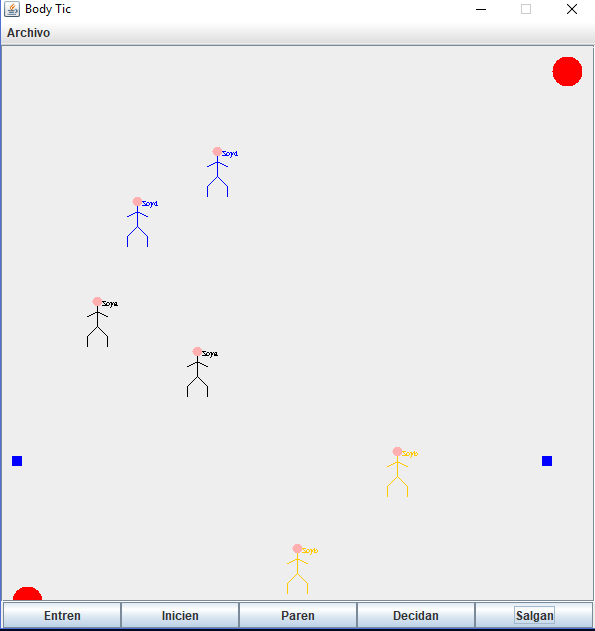
1. **Realicen una prueba de aceptación de este par de métodos: iniciando la aplicación exportando a unbodyTIc.txt. saliendo, entrando, creando un nuevo bodyTIc e importando el archivo otrobodyTIc.txt.. ¿Qué resultado obtuvieron? Capturen la pantalla final.**

Al exportar el archivo otrobodyTIc.txt creado después de importarlo de una situación avanzada de la ejecución el estado del salón actual se actualizo al estado anterior en el que se había guardado.



1. **Realicen otra prueba de aceptación de este método escribiendo un archivo de texto correcto en unbodyTIc.txt. e importe este archivo. ¿Qué resultado obtuvieron? Capturen la pantalla.**

Se crearon los objetos con las especificaciones que se escribieron en el archivo unbodyTIc.txt



**Analizando comportamiento**

**[En lab06.doc, \*.asta y \*.java] [NO OLVIDEN BDD y MDD]**

1. **Ejecuten la aplicación, tres clics, salven a un archivo cualquiera y ábranlo. ¿Describa el comportamiento?**

Al salvar el salon y luego al volverlo a abrir, este se guarda tal cual y continua con su funcionamiento normal.

1. **Ejecuten la aplicación, tres clics, exporten a un archivo cualquiera e importen. ¿Describa el comportamiento?**

Al exportar el salón y luego volverlo a abrir, los estados de las células se reinician, solo se conserva el tipo y su posición.

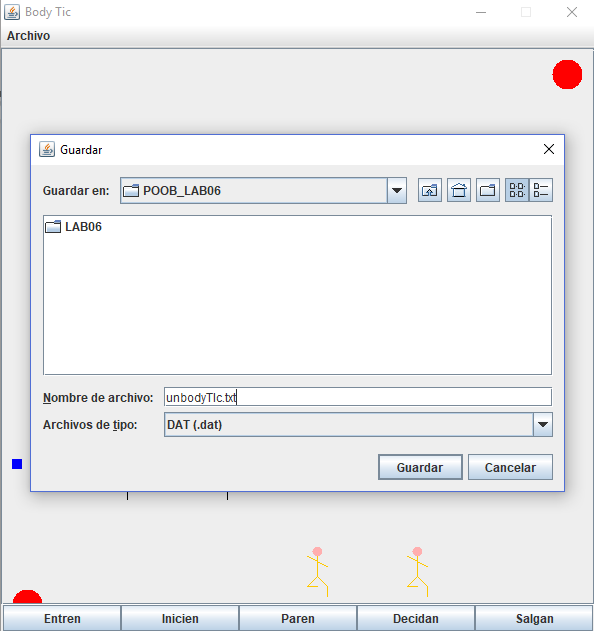
1. **¿Qué diferencias ven el comportamiento 1. y 2.? Expliquen los resultados.**

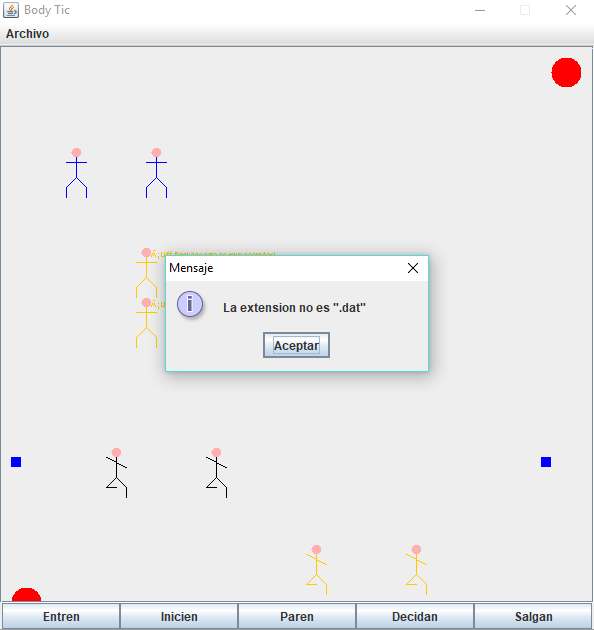
Al salvar, se guarda todo el objeto como tal, esto incluye sus atributos que a su vez pueden ser otros objetos, mientras que al exportar solo se está guardando le tipo de deportista y sus posiciones más no sus atributos.

**Perfeccionando salvar y abrir**

**[En lab06.doc, \*.asta y \*.java] [NO OLVIDEN BDD y MDD]**

1. **Copien las versiones actuales de abra y salve y renómbrenlos como abra01 y salve01**
2. **Perfeccionen el manejo de excepciones de los métodos abra y salve.**
3. **Realicen una prueba de aceptación para validar cada una de los nuevos mensajes diseñados, ejecútenla y capturen la pantalla final.**

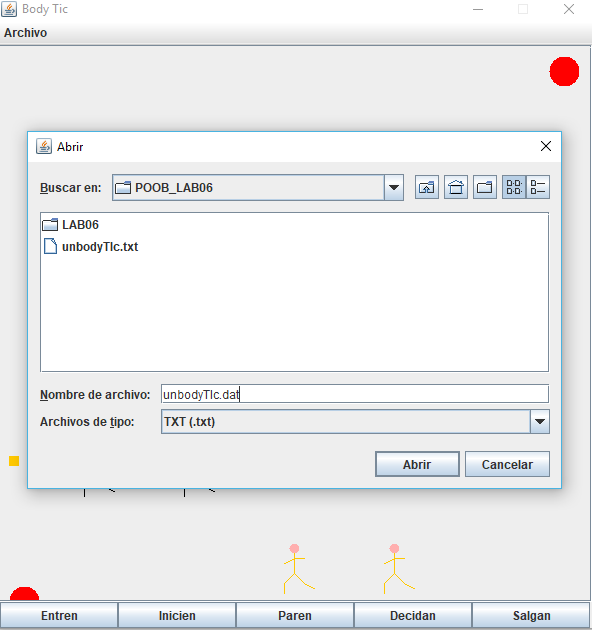
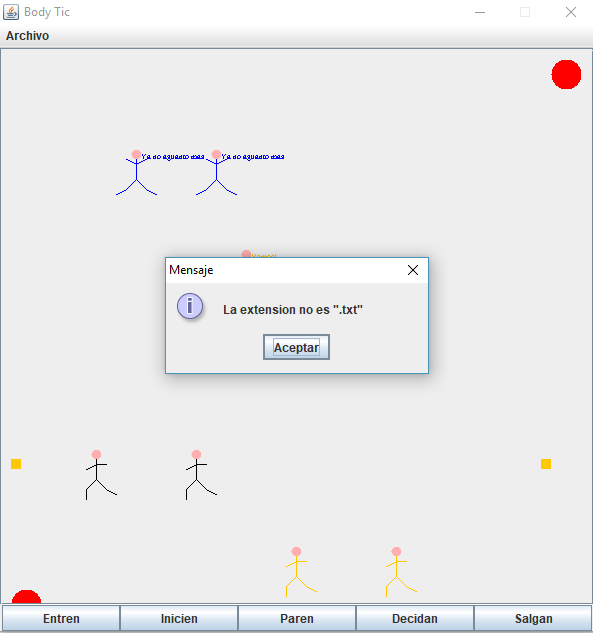




**Perfeccionando importar y exportar.**

**[En lab06.doc, \*.asta , bodyTIcErr.txt \*.java] [NO OLVIDEN BDD y MDD]**

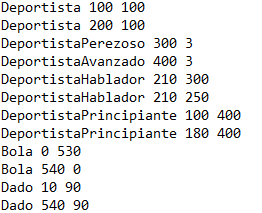
1. **Copien las versiones actuales de importe y exporte y renómbrenlos como importe01 y exporte01**
2. **Perfeccionen el manejo de excepciones de los métodos importe y exporte.**
3. **Realicen una prueba de aceptación para validar cada una de los nuevos mensajes diseñados, ejecútenla y capturen la pantalla final.**

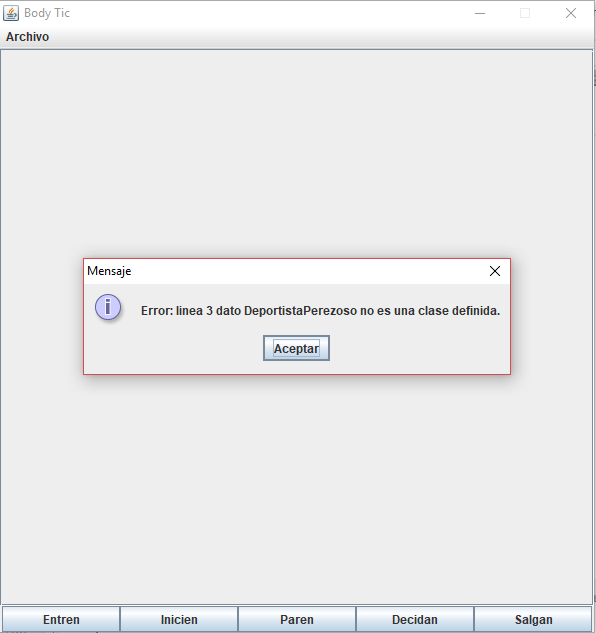
 

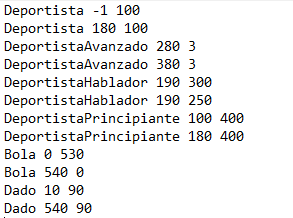
**Perfeccionando importar. Hacia un minicompilador.**

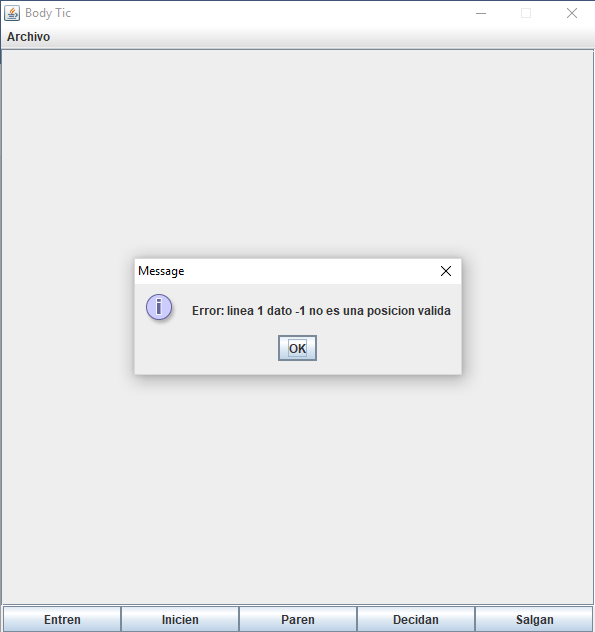
**[En lab06.doc, \*.asta , bodyTIcErr.txt \*.java] [NO OLVIDEN BDD y MDD]**

1. **Copien las versiones actuales de importe y exporte y renómbrenlos como importe02 y exporte02**
2. **Perfeccionen el método importe para que, además de los errores generales, en las excepciones indique el detalle de los errores encontrados en el archivo (como un compilador) : número de línea donde se encontró el error, palabra que tiene el error y causa de error.**
3. **Escriban otro archivo con errores, llámelo bodyTIcErr.txt, para ir arreglándolo con ayuda de su “importador”. Presente las pantallas que contengan los errores.**





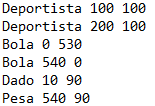


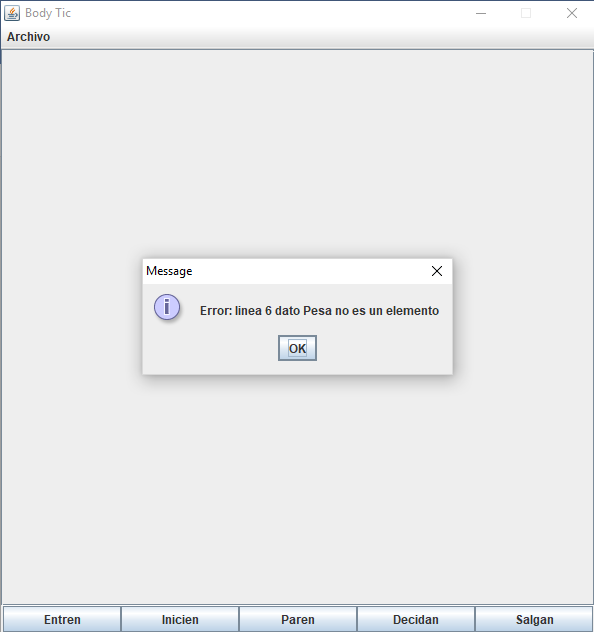


**Perfeccionando importar. Hacia un minicompilador flexible.**

**[En lab06.doc, \*.asta , bodyTIcFlex.txt \*.java] [NO OLVIDEN BDD y MDD]**

1. **Copien las versiones actuales de importe y exporte y renómbrenlos como importe03 y exporte03**
2. **Perfeccionen los métodos importe y exporte para que pueda servir para cualquier tipo de elementos creados en el futuro (Investiguen cómo crear un objeto de una clase dado su nombre)**
3. **Escriban otro archivo de pruebas, llámelo bodyTIcErrG.txt, para probar la flexibilidad. Presente las pantallas que contenga un error significativo.**





**RETROSPECTIVA**

1. **¿Cuál fue el tiempo total invertido en el laboratorio por cada uno de ustedes? (Horas/Hombre)**

15/Carlos Murillo

1. **¿Cuál es el estado actual del laboratorio? ¿Por qué?**

Incompleto, falto la realización de las pruebas debido a que no me alcanzo el tiempo.

1. **Considerando las prácticas XP del laboratorio. ¿cuál fue la más útil? ¿por qué?**

Para este laboratorio considero que la práctica que fue más útil es la de refactorización, debido a que toco realizar algunos cambios en el orden de algunos parámetros de algunas clases para que todo funcionara correctamente.

1. **¿Cuál consideran fue el mayor logro? ¿Por qué?**

Haber llegado hasta este punto de completitud del laboratorio ya que para esta entrega me toco realizarlo de manera individual.

1. **¿Cuál consideran que fue el mayor problema técnico? ¿Qué hicieron para resolverlo?**

El mayor problema que se me presento fue lograr perfeccionar el método de importar para lograr que sea un minicompilador flexible

1. **¿Qué hicieron bien como equipo? ¿Qué se comprometen a hacer para mejorar los resultados?**

Nada, hice el laboratorio de manera individual. Me comprometo a distribuir de mejor manera mi tiempo destinado para el desarrollo de estos trabajos.