**Eduard Arias**

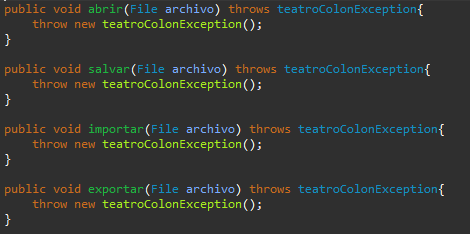
**LABORATORIO 6**

1. **Creando la maqueta**

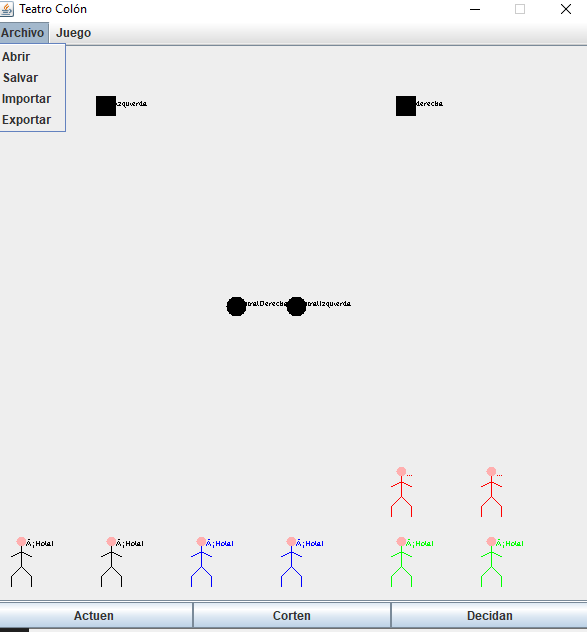
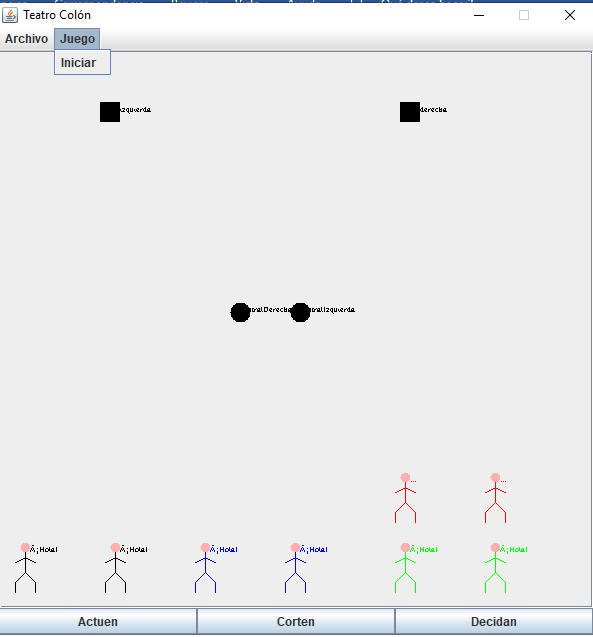
[En lab06.doc, \*.asta y \*.java] [NO OLVIDEN BDD y MDD]

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

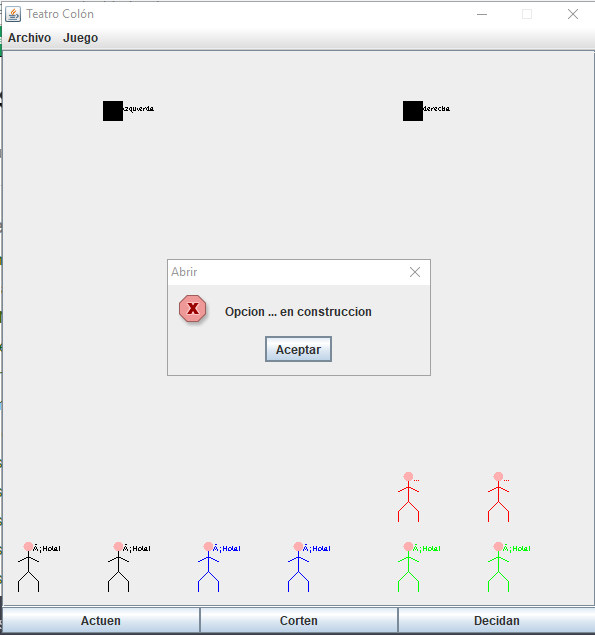
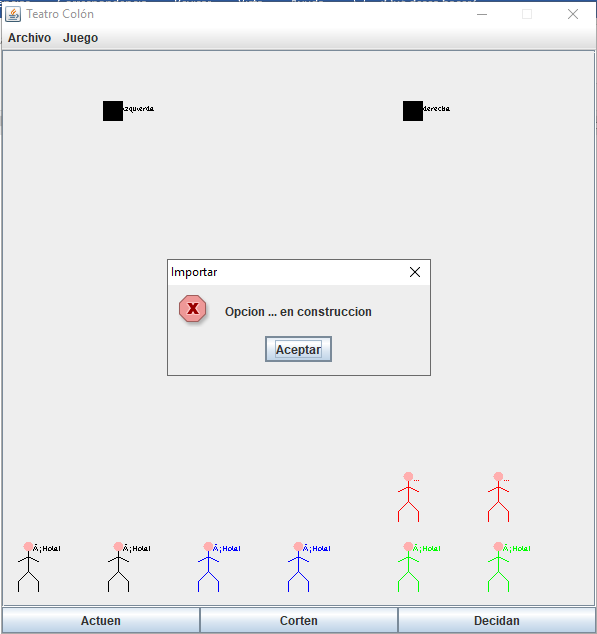
* 1. **MODELO:** Preparen los métodos correspondientes a reiniciar y a las cuatro opciones básicas de entrada-salida (salve,abra,exporte,importe). Los métodos deben simplemente propagar una teatroColonExcepcion con el mensaje: “Opción … en construcción”. Los métodos de entrada salida deben tener un parámetro File.



* 1. **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 ( opcionSalvar, opcionAbir, opcionExportar, opcionImportar, opcionIniciar, opcionCerrar). Estos últimos métodos, por ahora, sólo deben llaman directamente el método correspondiente de la capa de aplicación. Capturen una pantalla significativa.

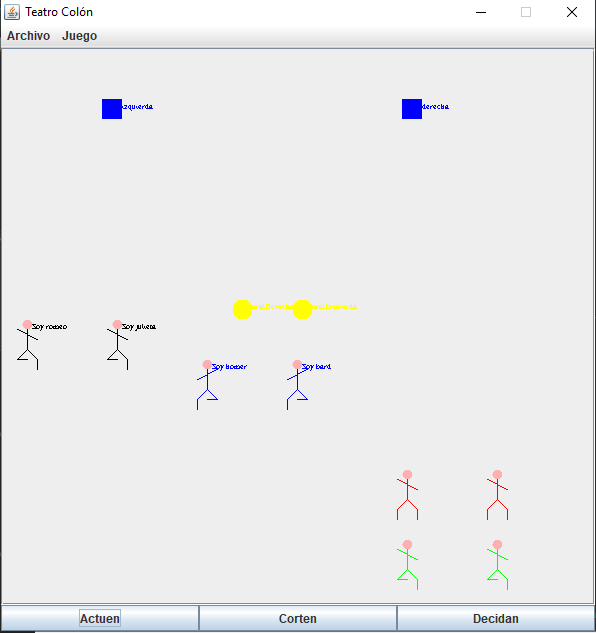
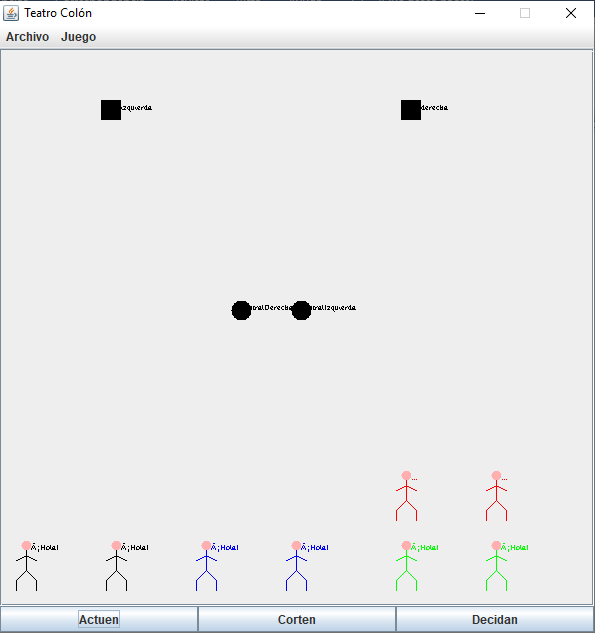
1. **Implementando salir e iniciar**

[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 teatroColon. Capturen una pantalla significativa.

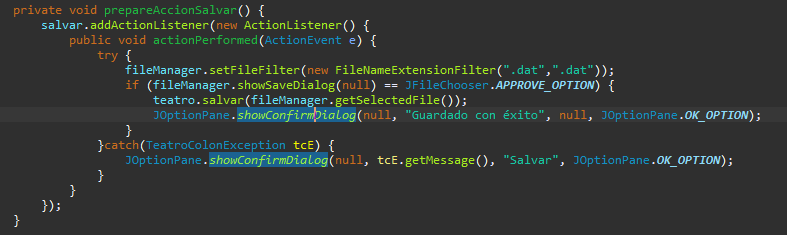
 

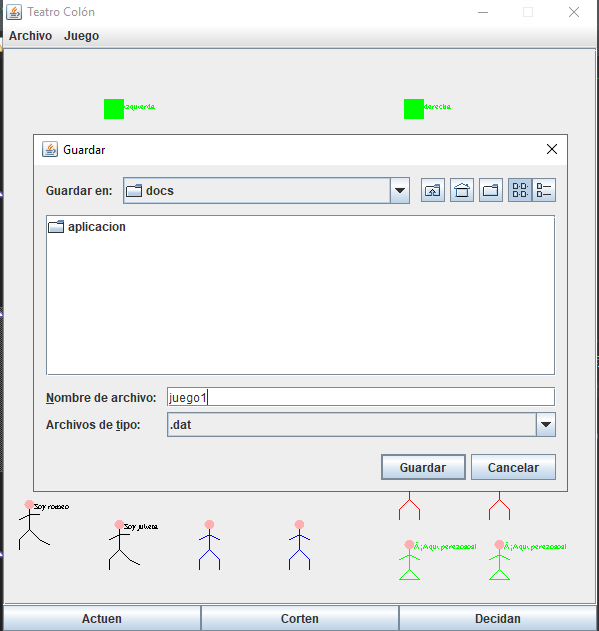
1. **Implementando salvar y abrir**

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

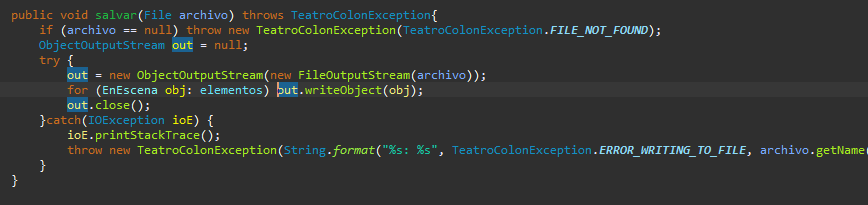
Las opciones salvar y abrir van a ofrecer servicios de persistencia de un teatroColon como objeto. Los nombres de los archivos deben tener como apellido .dat.

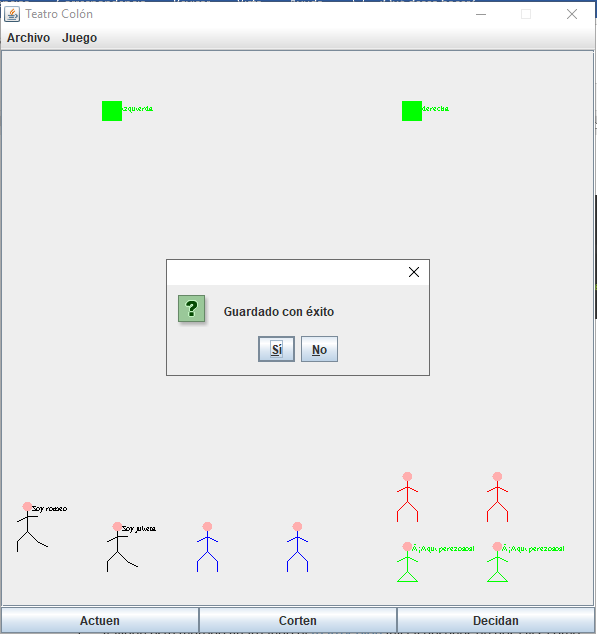
* 1. Construyan el método opcionSalvar que une de forma adecuada la capa de presentación con la capa de aplicación. Usen un FileChooser y atiendan la excepción. Ejecuten la aplicación probando las diferentes opciones del FileChooser y capturen una pantalla significativa.





* 1. Construyan el método salve que ofrece el servicio de guardar en un archivo el estado actual del teatroColon.

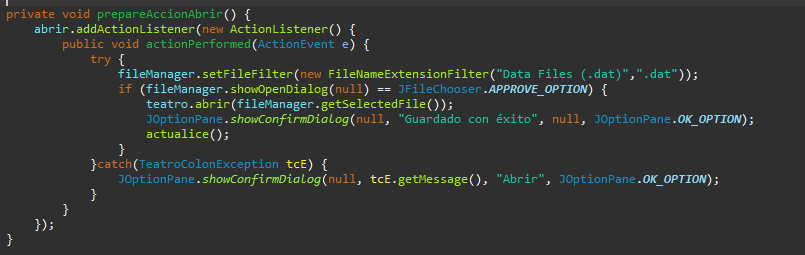


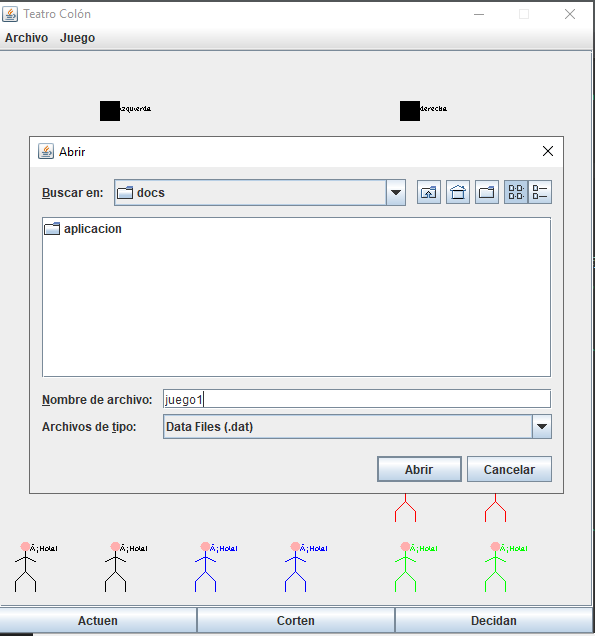


* 1. Validen este método guardando el teatroColon inicial después de dos clics como unteatroColon.dat. ¿El archivo se creó en el disco? ¿Cuánto espacio ocupa?

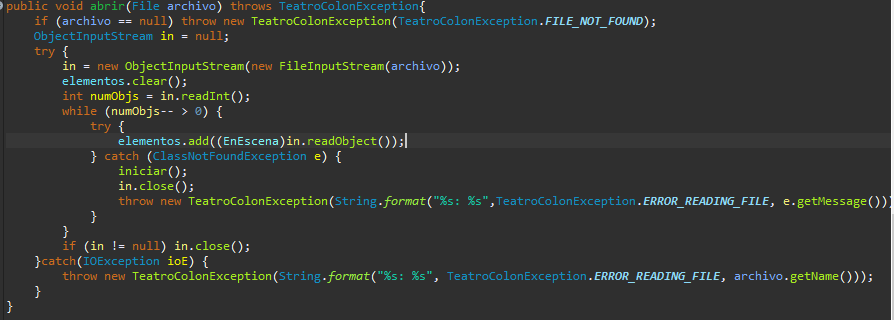
El archivo se creó en la carpeta en donde le indicamos al FileChooser (docs -> en el laboratorio) y pesa 2Kb. Se le indicó el numero de objetos escritos con un numer al inicio del archivo.

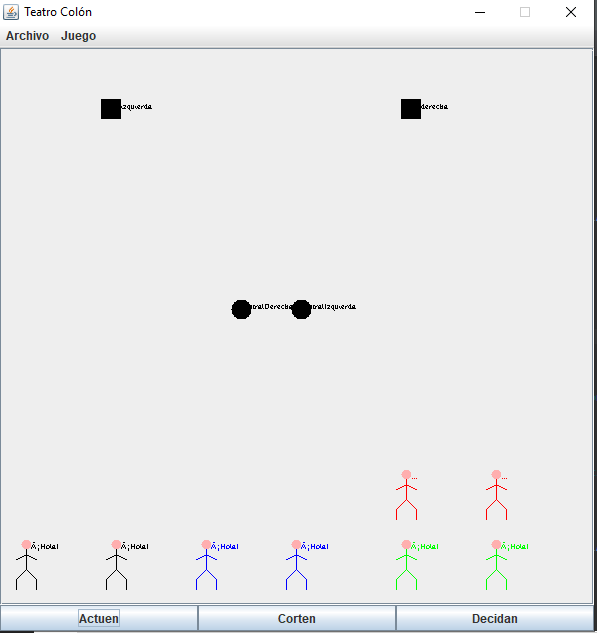
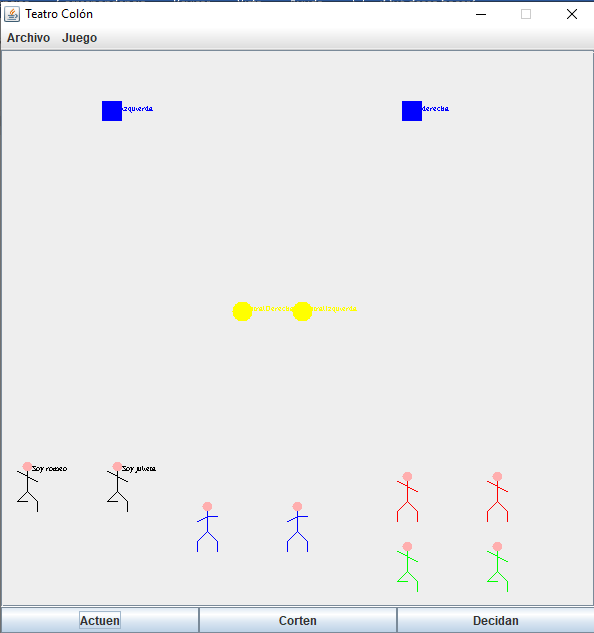
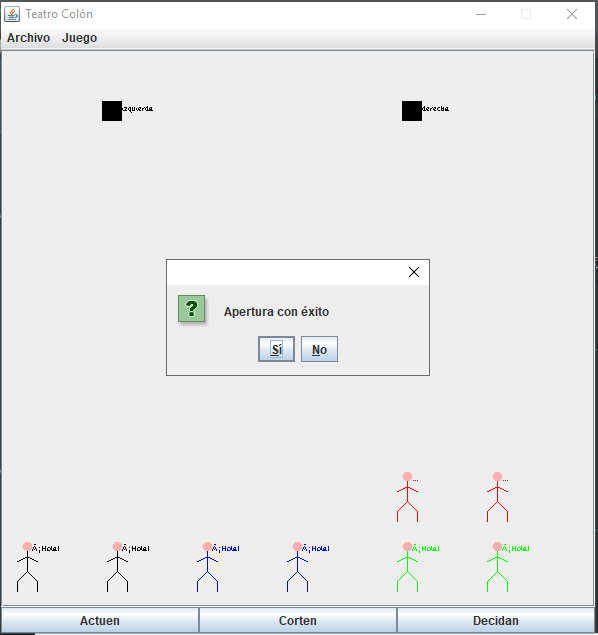
* 1. Construyan el método opcionAbrir que une de forma adecuada la capa de presentación con la capa de aplicación. Ejecuten la aplicación probando las diferentes opciones del FileChooser y capturen una pantalla significativa.





* 1. Construyan el método abra que ofrece el servicio de leer un teatroColon 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 teatroColon y abriendo el archivo unteatroColon.dat. Capturen imágenes significativas de estos resultados.

Falta Prueba

1. **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 teatroColon desde un archivo de texto y exportarlo. Los nombres de los archivos de texto deben tener como apellido .txt

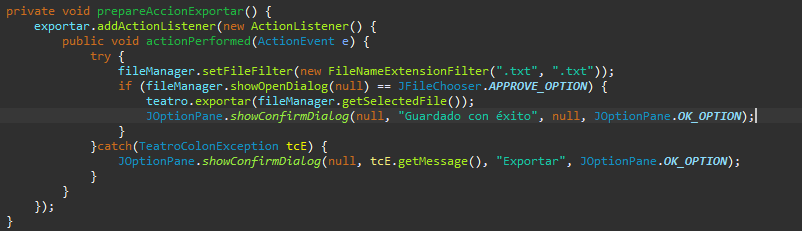
Los archivos texto tienen una línea de texto por cada elemento

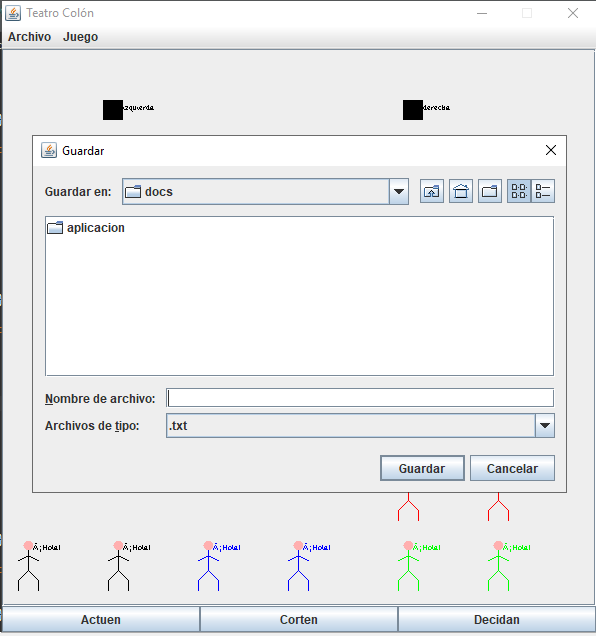
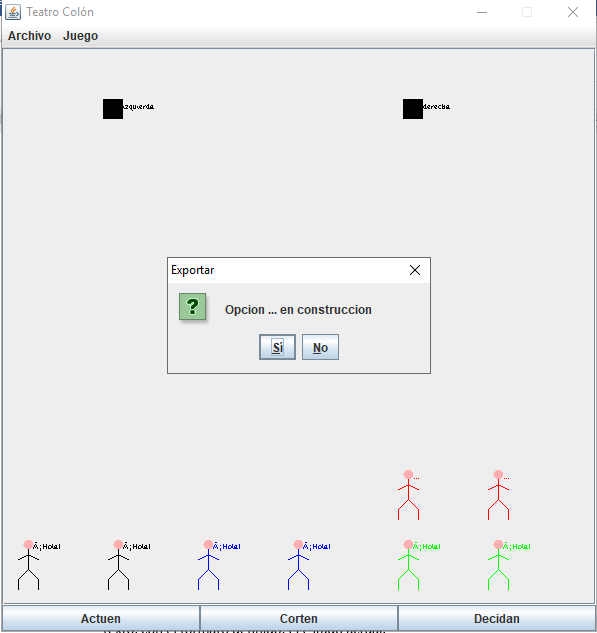
En cada línea asociada un elemento se especifica el tipo y la posición.

Normal 20 20 Romeo

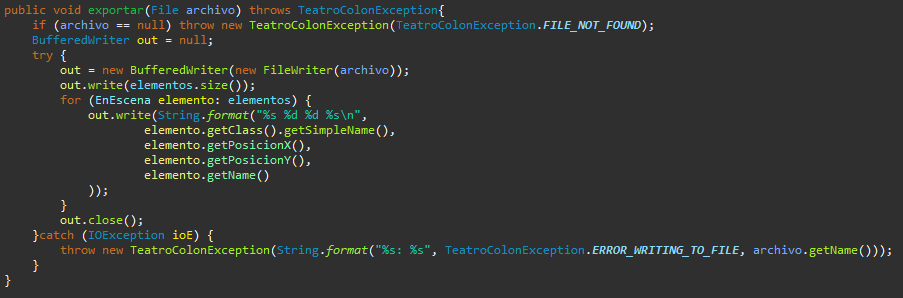
Luz 50 50

* 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.



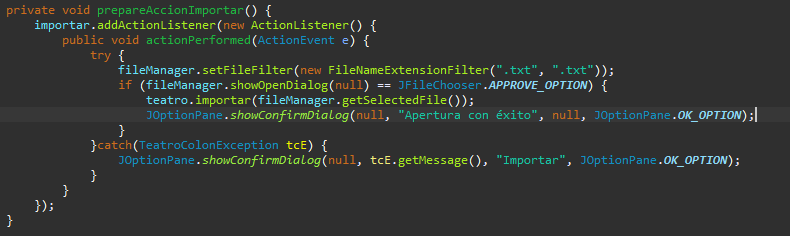
* 1. Construyan el método exporte que ofrece el servicio de exportar a un archivo texto, con el formato definido, el estado actual.

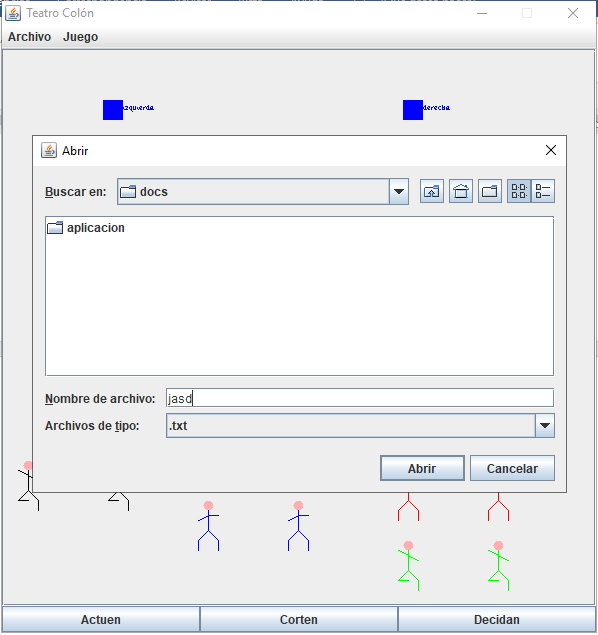
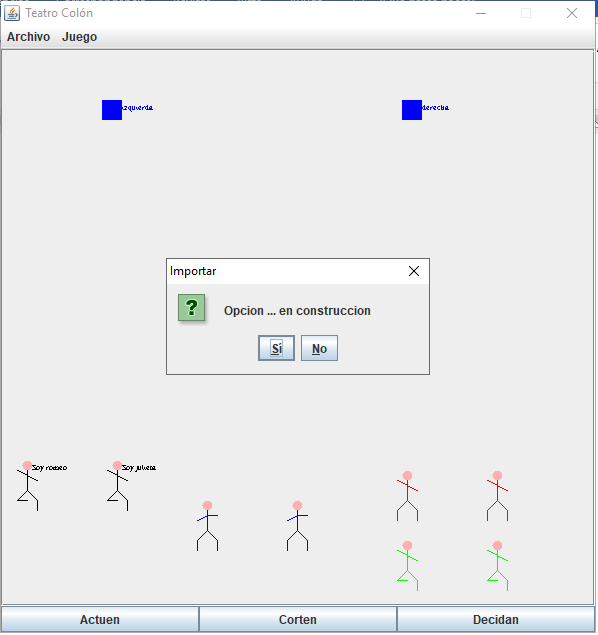


* 1. Realicen una prueba de aceptación de este método: iniciando la aplicación y exportando como unteatroColon.txt. Editen el archivo y analicen los resultados. ¿Qué pasó?

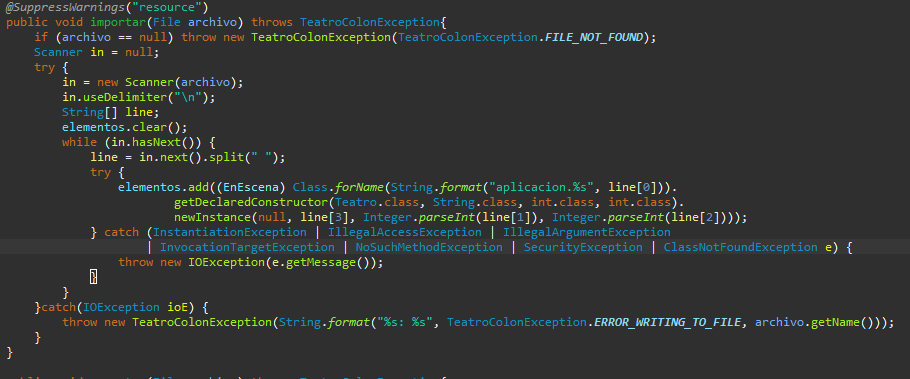
La información contenida en el archivo ya no corresponde con la del juego. No ocurrió ningún error de sistema.

* 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.

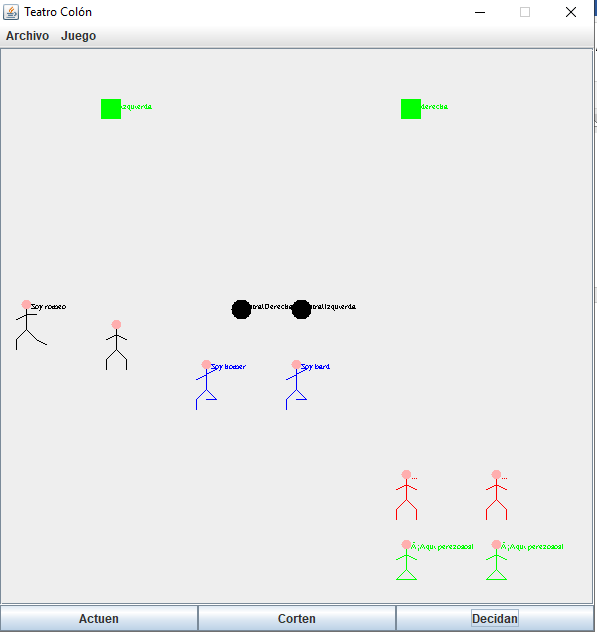


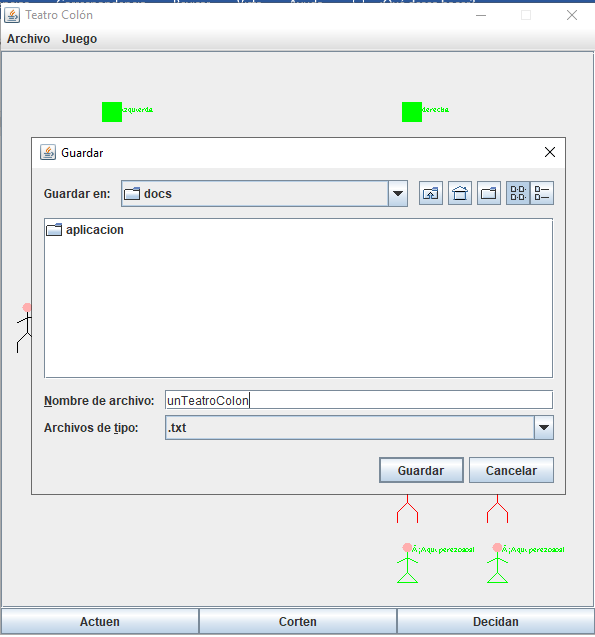
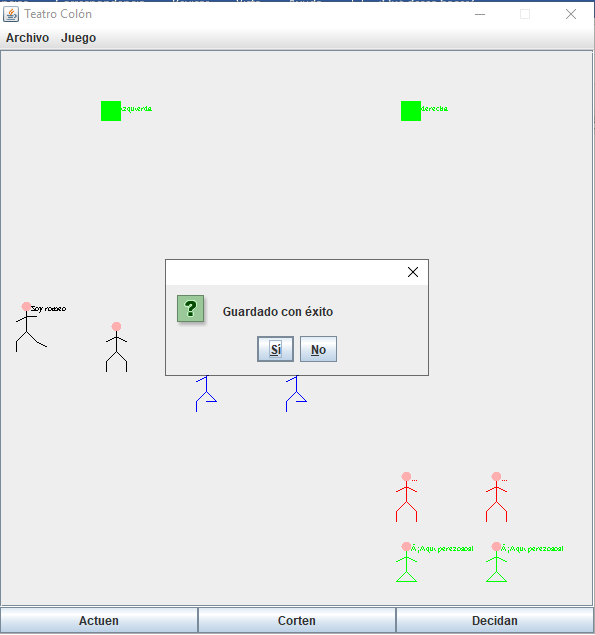
* 1. Construyan el método importe que ofrece el servicio de importar de un archivo texto con el formato definido. Por ahora sólo considere un mensaje de error general. (Consulten en la clase String los métodos trim y split)



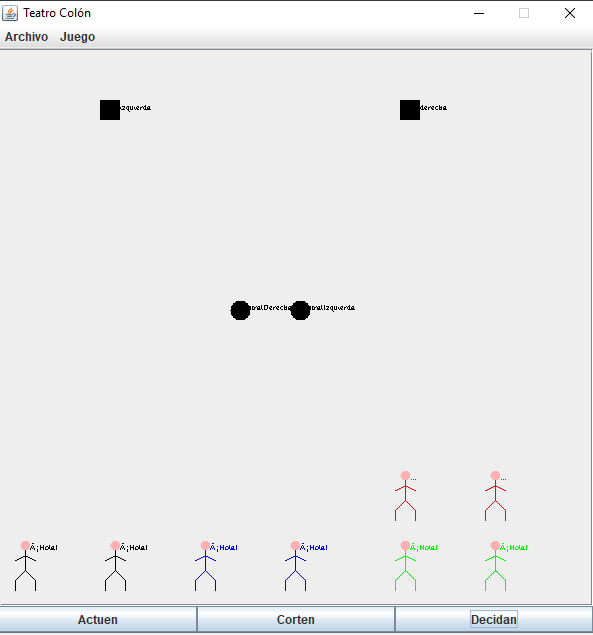
* 1. Realicen una prueba de aceptación de este par de métodos: iniciando la aplicación exportando a unteatroColon.txt saliendo, entrando, creando un nuevo teatroColon e importando el archivo otroteatroColon.txt. ¿Qué resultado obtuvieron? Capturen la pantalla final.
* Estado inicial despues de unos cliks



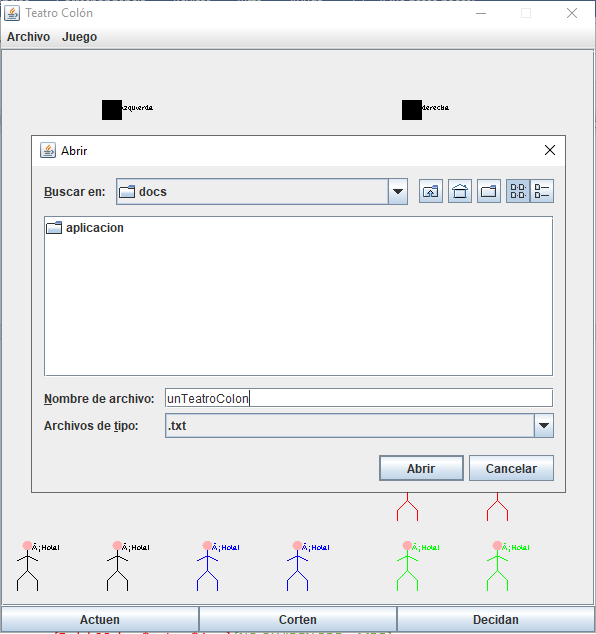
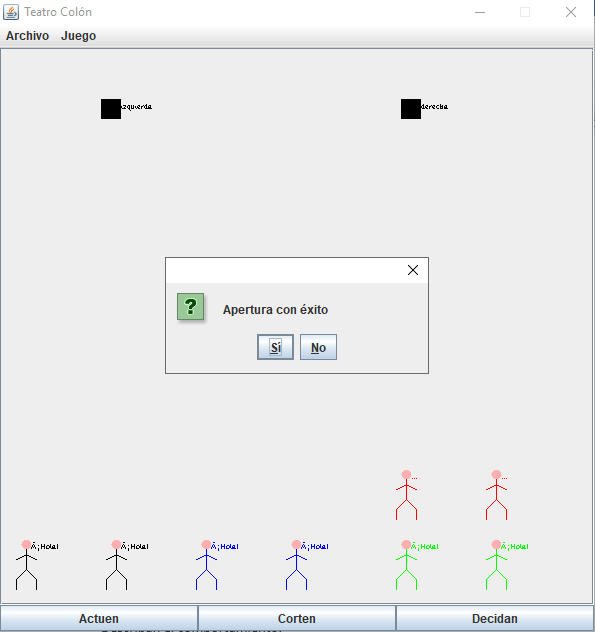
* Exportando a “unTeatroColon.txt”

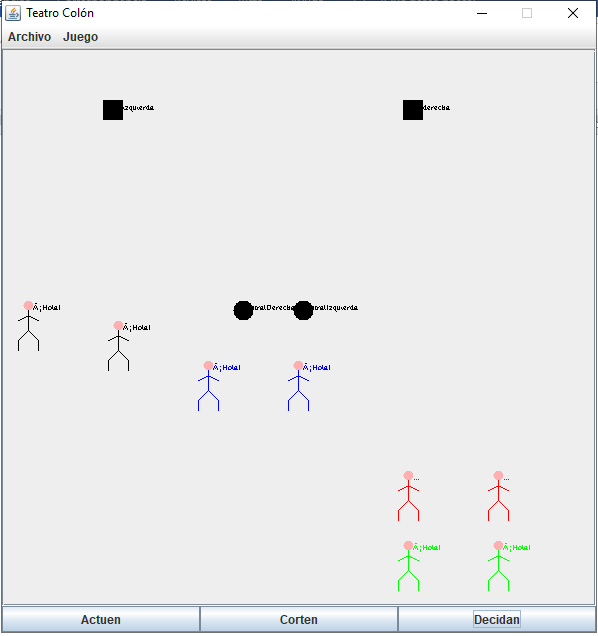
* Iniciando un nuevo juego



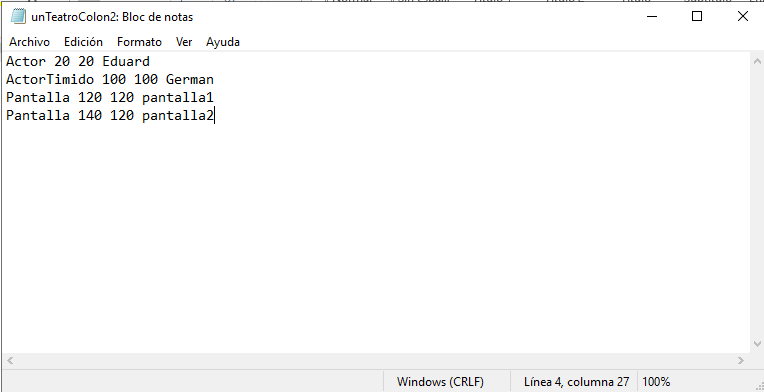
* Importando “unTeatroColon.txt”

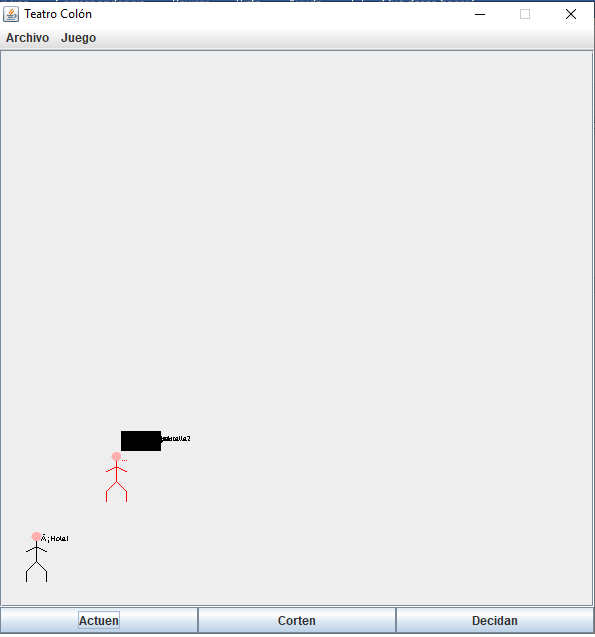
* Estado final



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



* Despues de importar



1. **Analizando comportamiento**

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

* 1. Ejecuten la aplicación, den tres clics, salven a un archivo cualquiera y ábranlo. Describan el comportamiento.

Al abrir el archivo .dat guardado (como “teatExmpl.dat” en est caso) todos los objetos cargados tal y como estaban anteriormente.

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

Al abrir el archivo .txt guardado (como “teatExmpl.txt” en este caso) los elementos aparecieron completos y en sus posiciones pero “por defecto”; es decir, como si recién hubiesen sido creados.

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

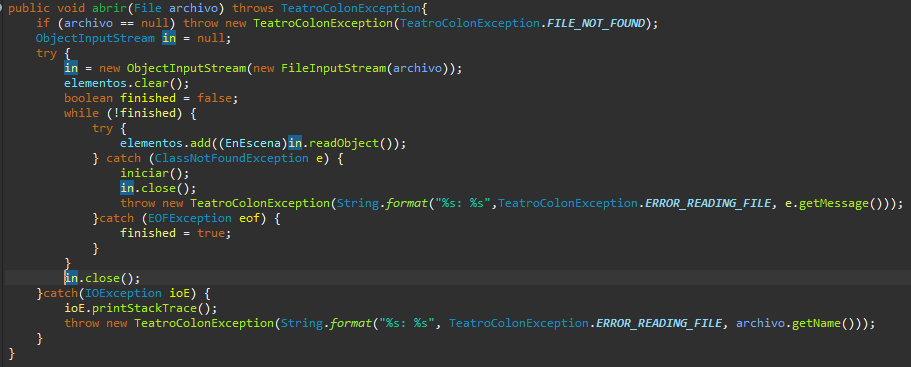
En el primer caso no hubo diferencia alguna entre el guardado y la apertura de la información ya que el ObjectOutputStream persiste toda información relacionada al objeto, en este caso incluye cómo están sus brazos, el mesnaje que estaban diciendo y su color.

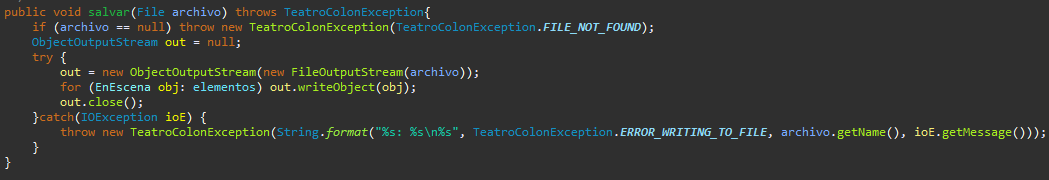
En el segundo caso, por el contrario, solo persistió la información que directamente se escribió en el archivo: El tipo de objeto, la posición en pantalla y el mensaje que tenían. No se almacenó nada más, por eso parecieran recién creados.

1. **Perfeccionando salvar y abrir**

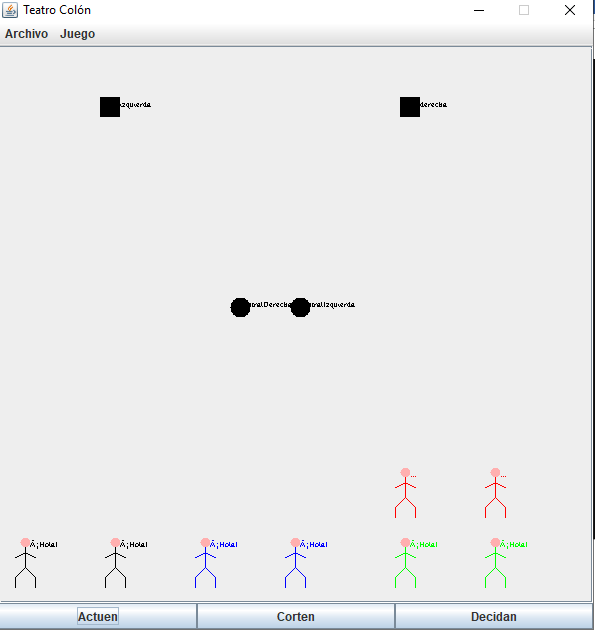
[En lab06.doc, \*.asta y \*.java] [NO OLVIDEN BDD y MDD]

* 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 detallando los errores.

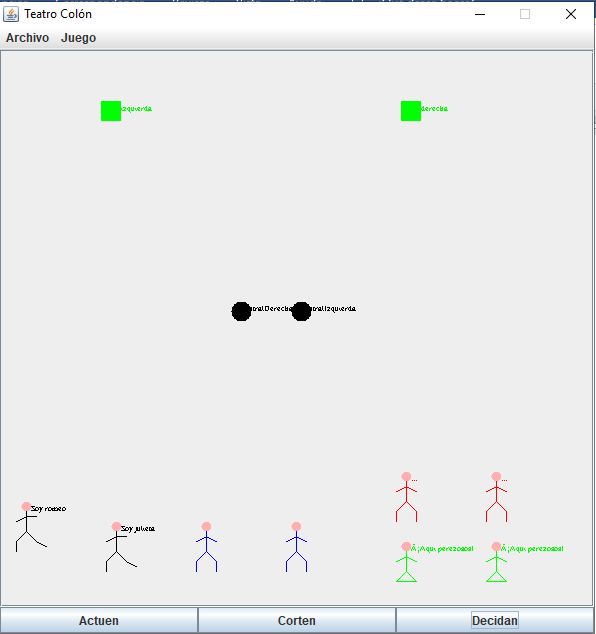




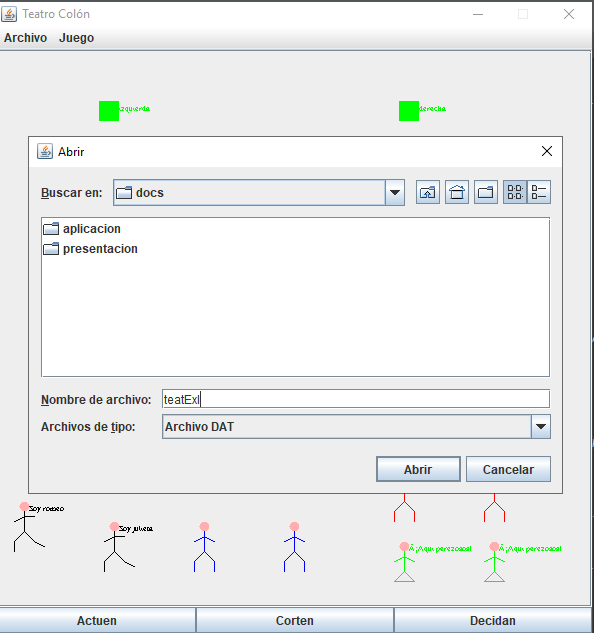
* 1. Realicen una prueba de aceptación para validar cada una de los nuevos mensajes diseñados, ejecútenla y capturen la pantalla final.
* Iniciando un juego nuevo

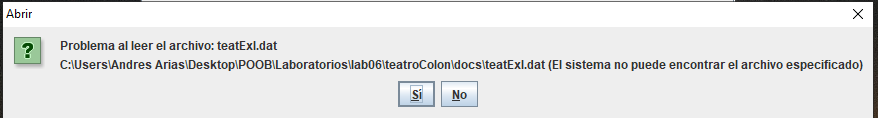


* Despues de 3 clicks en “decidan” (para garantizar aletoriedad)

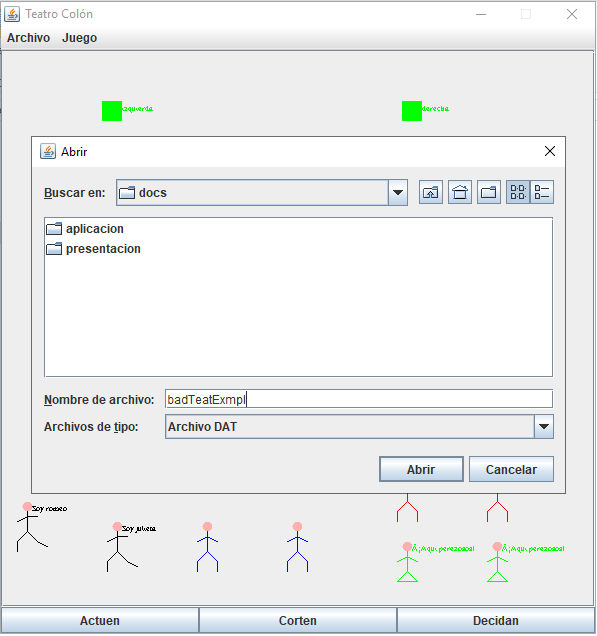
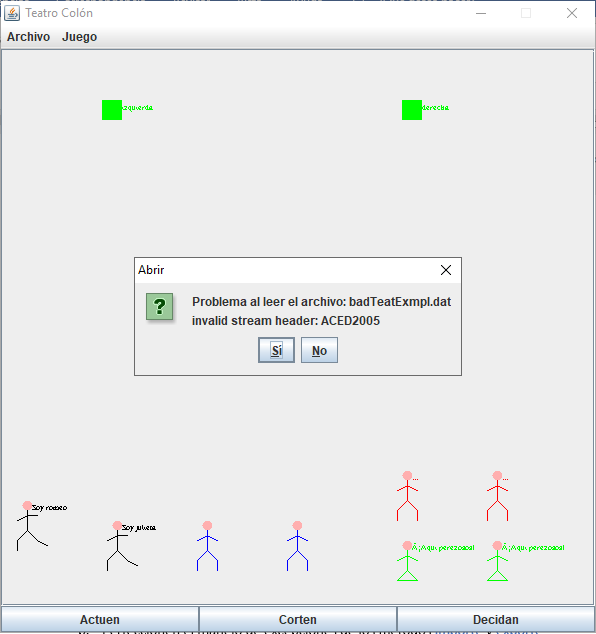


* Intentando abrir el archivo “teatExl2.dat”





* Leyendo un archivo mal escrito (“badTeatExmpl.dat”)

1. **Perfeccionando importar y exportar.**

[En lab06.doc, \*.asta, \*.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 detallando los errores.

Se realizó desde la primera versión de los métodos.

* 1. Realicen una prueba de aceptación para validar cada una de los nuevos mensajes diseñados, ejecútenla y capturen la pantalla final.

1. **Perfeccionando importar. Hacia un minicompilador.**

[En lab06.doc, \*.asta, teatroColonErr.txt y \*.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 teatroColonErr.txt, para ir arreglándolo con ayuda de su “importador”. Presente las pantallas que contengan los errores.

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

[En lab06.doc, \*.asta, teatroColonFlex.txt y \*.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 teatroColonErrG.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 ?

Eduard : 15 horas

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

El laboratorio está casi completo. La parte del minicompilador, aunque se entendió no alcanzó el tiempo destinado en la elaboración de este lab.

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

La práctica xp que más nos sirvió fue **Designing**[[refactor](http://www.extremeprogramming.org/rules/refactor.html) Refactor](http://www.extremeprogramming.org/rules/refactor.html) whenever and wherever possible. debido a que fue esto lo que nos permitió extender mejor el comportamiento de la aplicación que ya se tenía.

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

Lograr vencer todos los obtáculos al persistir datos en archivos, debido a que esto ayudó a mejorar nuestro conocimiento en el lenguaje y a reforzar lo que vimos en clase.

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

Hubo muchas complicaciones al persistir datos, en especial cómo leerlos de un archivo de texto plano. Se solucionó al leeer con mayor profundidad la documentación del lenguaje.

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

Mejorar la distribución de tiempo en proyectos futuros para no reducir el alcance.