Angie Medina – Jose Pérez

POOB - 1

2020 - 2

Laboratorio 6/6 I/O [ :) ]

**DESARROLLO**

**Preparando**

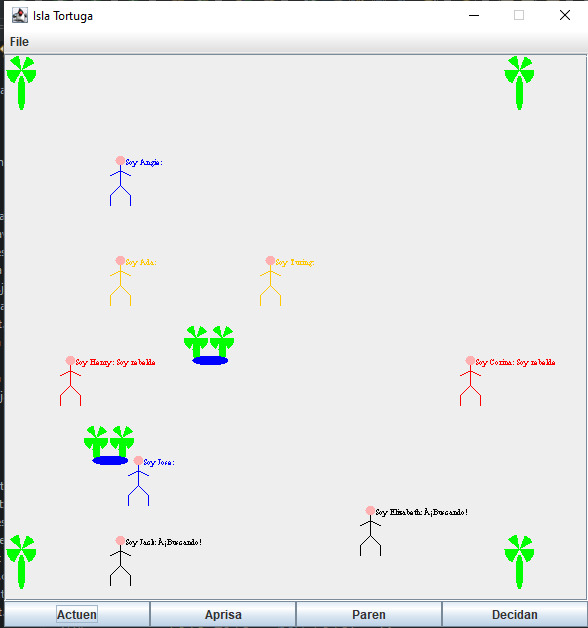
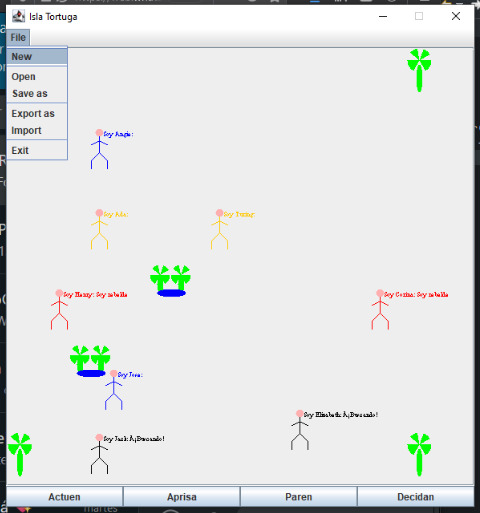
En este laboratorio vamos a extender el proyecto *islaTortuga* adicionando un menú barra con las opciones básicas de entrada-salida y las opciones estándar nuevo y salir.

1. En su directorio descarguen la versión del proyecto realizado por ustedes para el laboratorio 03 y preparen el ambiente para trabajar desde **CONSOLA**
2. Ejecuten el programa, revisen la funcionalidad.

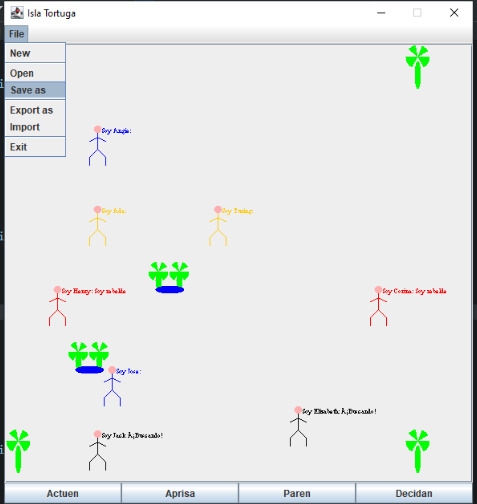
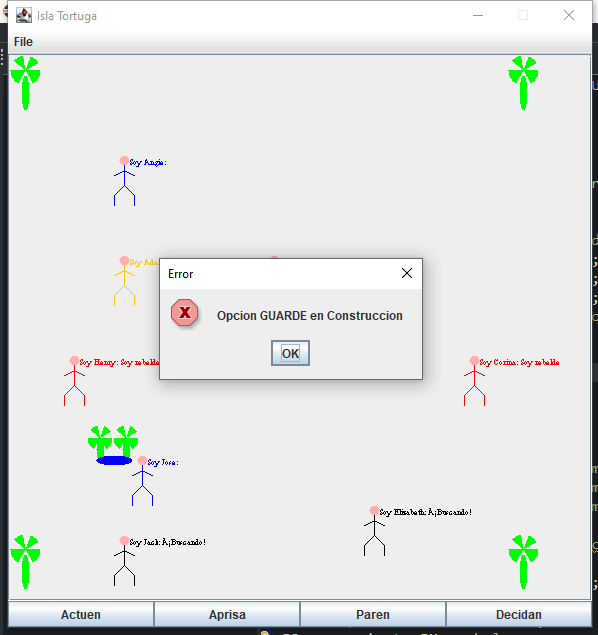
**Creando la maqueta**

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

1. **MODELO:** Preparen en la clase fachada del modelo los métodos correspondientes a las cuatro opciones básicas de entrada-salida (abra, guarde, importe y exporte). Los métodos deben simplemente propagar una *islaTortugaExcepcion* con el mensaje: “Opción ... en construcción”. Los métodos 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 nuevo y salir (Nuevo, Abrir, Guardar como, Importar, Exportar como, Salir). No olviden incluir los separadores. Para esto creen el método *prepareElementosMenu*. Capturen la pantalla correspondiente.

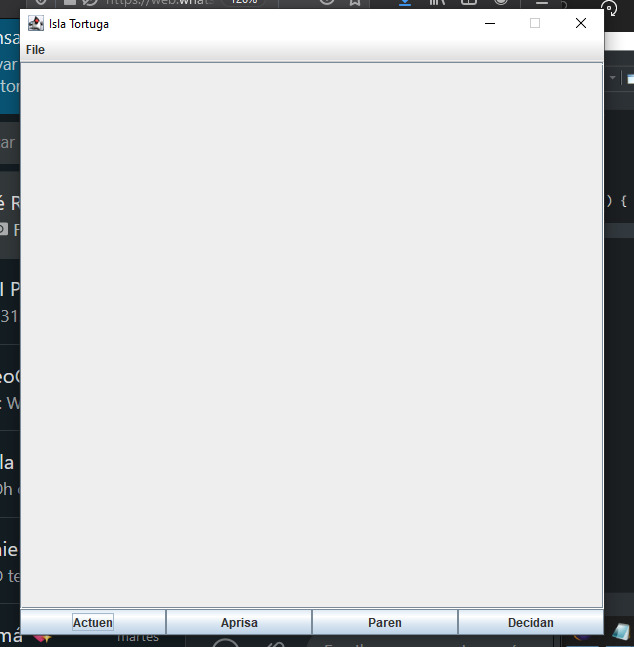
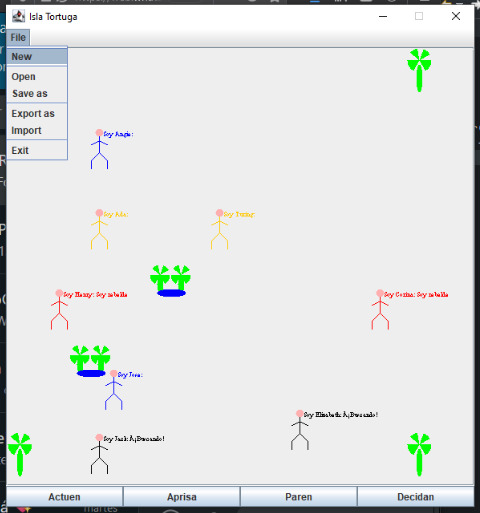
1. **CONTROLADOR:** Construyan los oyentes correspondientes a las seis opciones. Para esto creen el método *prepareAccionesMenu* y los métodos base del controlador (opcionAbir, opcionGuardar, opcionExportar, opcionImportar, opcionNuevo, opcionSalir), Estos métodos, por ahora, llaman directamente el método correspondiente de la capa de aplicación. No incluyan todavía el *FileChooser*. Capturen una pantalla significativa.

**Implementando salir y nuevo**

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.

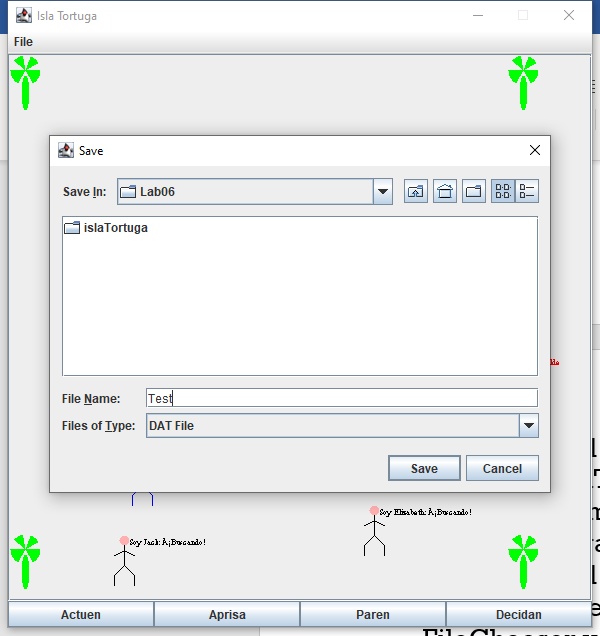
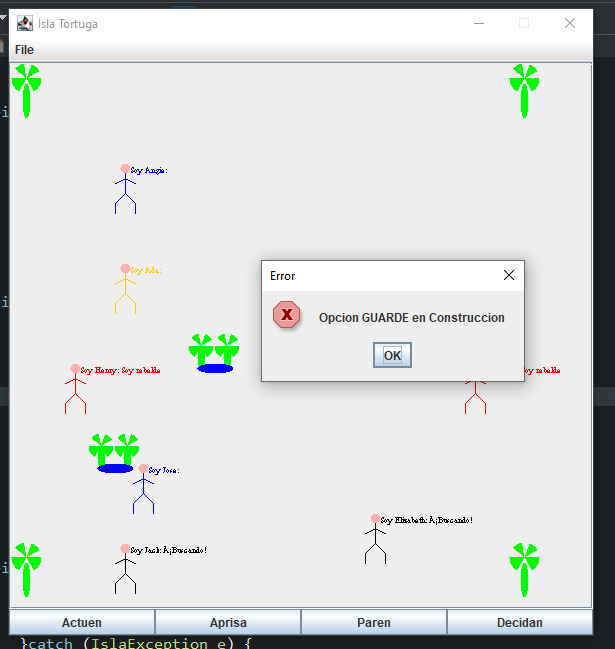
1. Construyan el método opcionSalir que hace que se termine la aplicación. No es necesario incluir confirmación.
2. Construyan el método opcionNuevo que crea una nueva islaTortuga. Capturen una pantalla significativa.



**Implementando salvar y abrir**

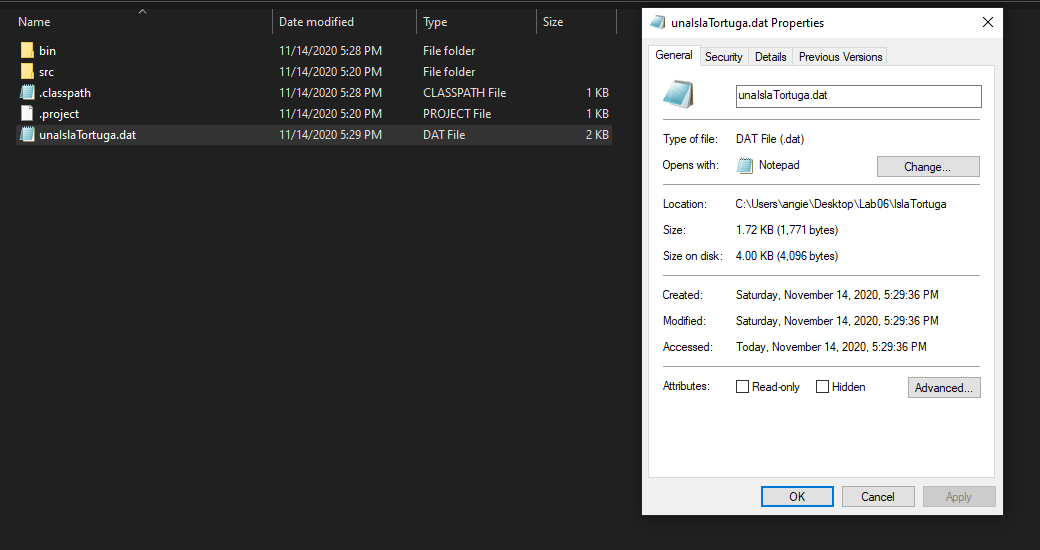
Las opciones salvar y abrir van a ofrecer servicios de persistencia de una islaTortuga como objeto. Los nombres de los archivos deben tener como extensión .dat.

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

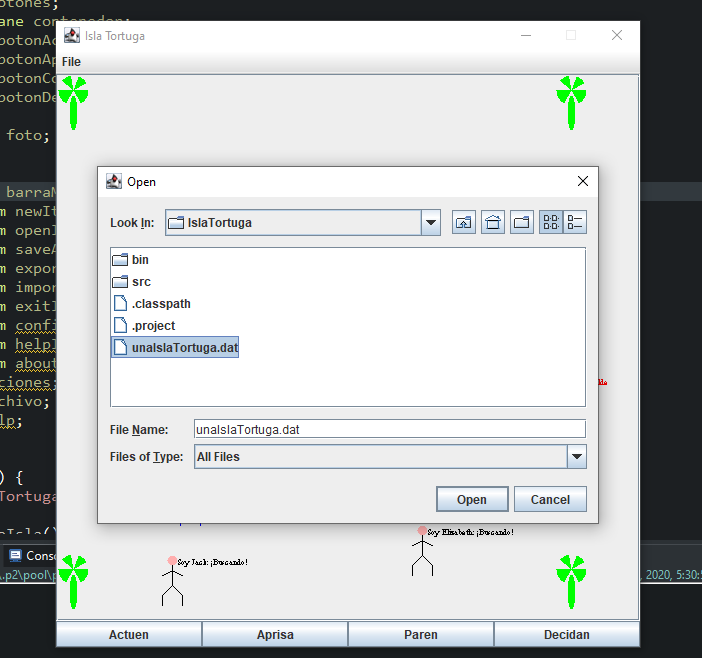
 

1. Construyan el método guarde que ofrece el servicio de guardar en un archivo el estado actual del islaTortuga.
2. Validen este método guardando la isla inicial después de dos clics como *unaIslaTortuga.dat*. ¿El archivo se creó en el disco? ¿Cuánto espacio ocupa?

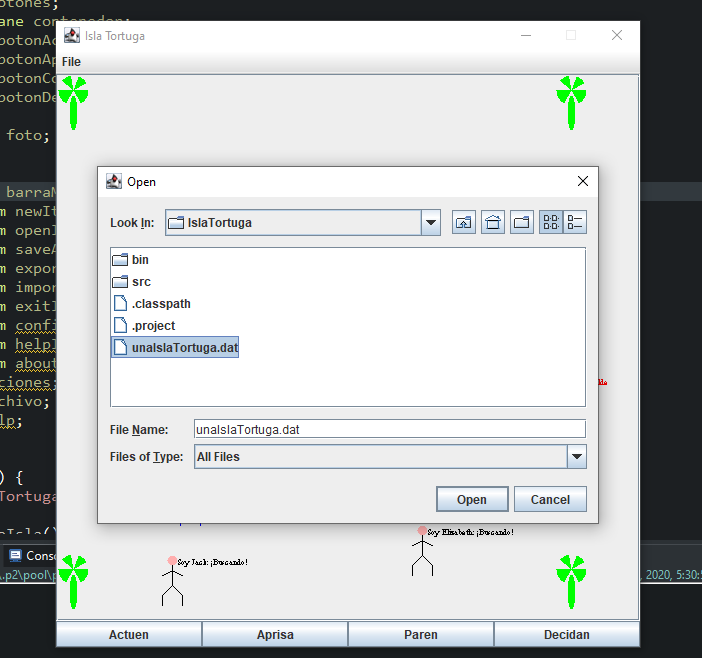
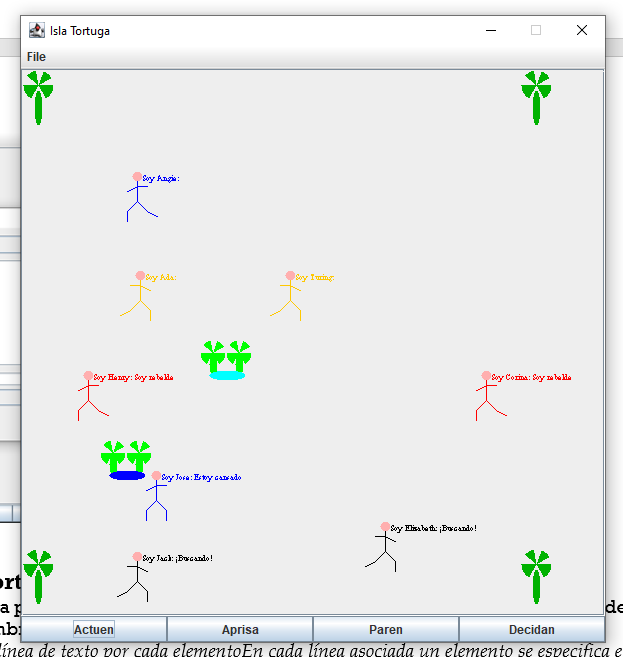
Si, el archivo se creó en el directorio, ocupa 1.71KB



1. Construyan el método *opcionAbrir* que une de forma adecuada la capa de presentación con la capa de dominio. 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 *islaTortuga* de un archivo. Por ahora para las excepciones sólo consideren un mensaje de error general.
2. Realicen una prueba de aceptación para este método iniciando la aplicación, creando una nueva situación en el *islaTortuga* y abriendo el archivo *unaIslaTortuga.dat*. Capturen imágenes significativas de estos resultados

**Implementando importar y exportar**

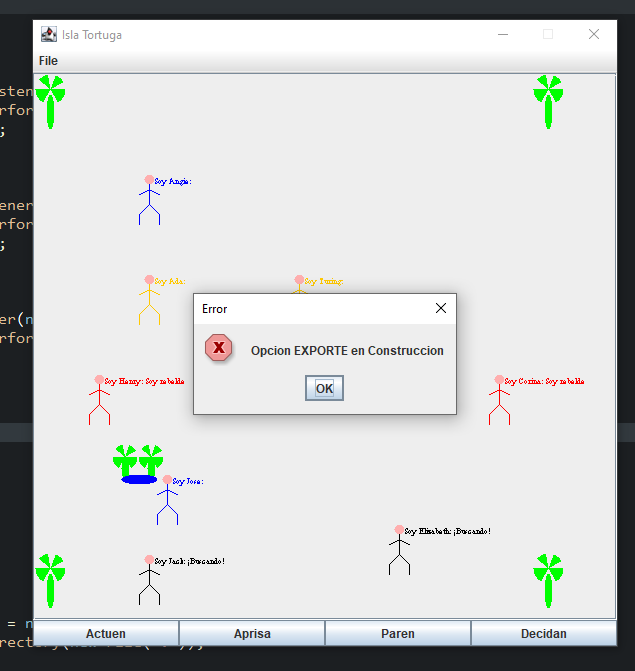
Estas operaciones nos van a permitir importar información de una *islaTortuga* desde un archivo de texto y exportarlo. Los nombres de los archivos de texto deben tener como extensión.txt

*Los archivos texto tienen una línea de texto por cada elemento. En cada línea asociada un elemento se especifica el tipo y la posición.*

*Pirata jack 20 20*

*Palmera 50 50*

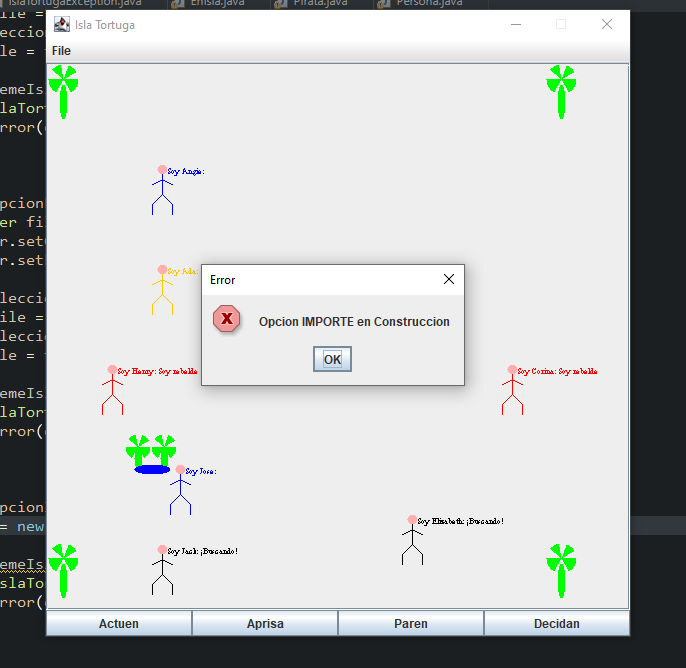
1. Construyan el método *opcionExportar* que une de forma adecuada la capa de presentación con la capa de aplicación. Ejecuten la aplicación y capturen una pantalla significativa.



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

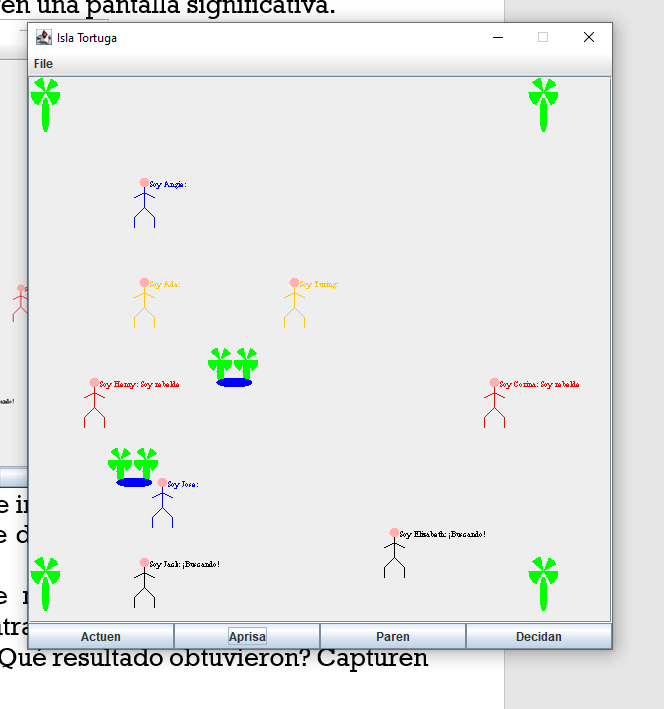
En el archivo quedo escrita la información de la isla

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



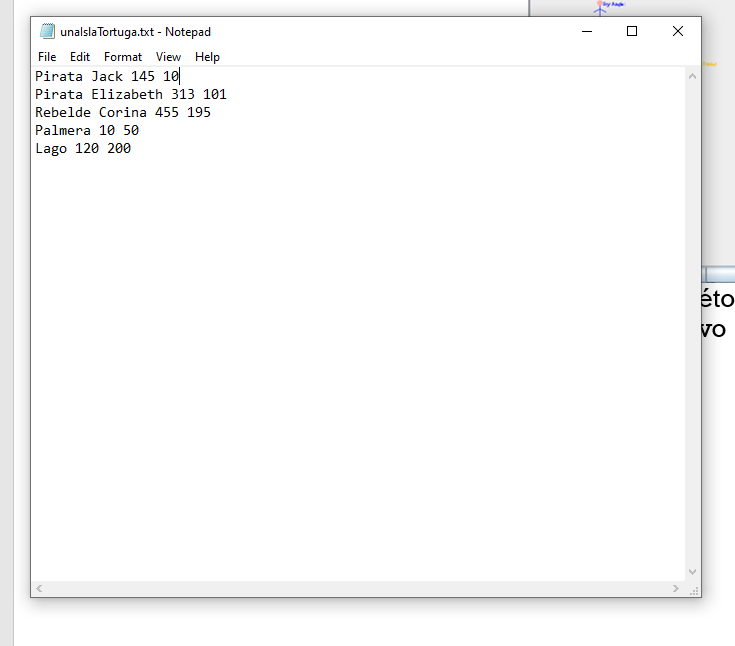
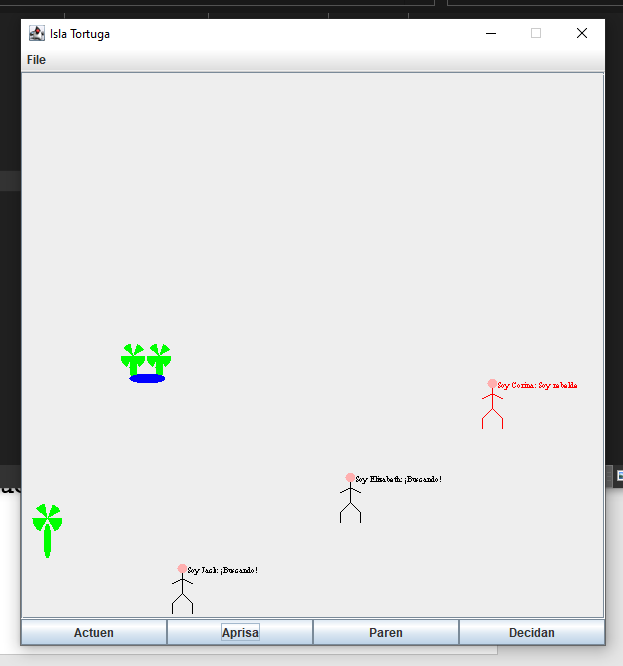
1. Construyan el método importe que ofrece el servicio de importar de un archivo texto con el formato definido. Por ahora sólo considere un mensaje de error general. (Consulten en la clase *String* los métodos *trim* y *split*)
2. Realicen una prueba de aceptación de este par de métodos: iniciando la aplicación exportando a *otraIslaTortuga.txt*. saliendo (actúen(?)), entrando(aprisa(?)), creando un nuevo islaTortuga e importando el archivo *otraIslaTortuga.txt.* ¿Qué resultado obtuvieron? Capturen la pantalla final.

Se obtuvo la isla antes de realizar las acciones propuestas



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

El resultado que se obtuvo fue el esperado, los elementos reflejados en la pantalla

**Analizando comportamiento**

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

Los elementos automáticamente vuelven al estado exacto que tenían al momento en el que se guardó la partida.

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

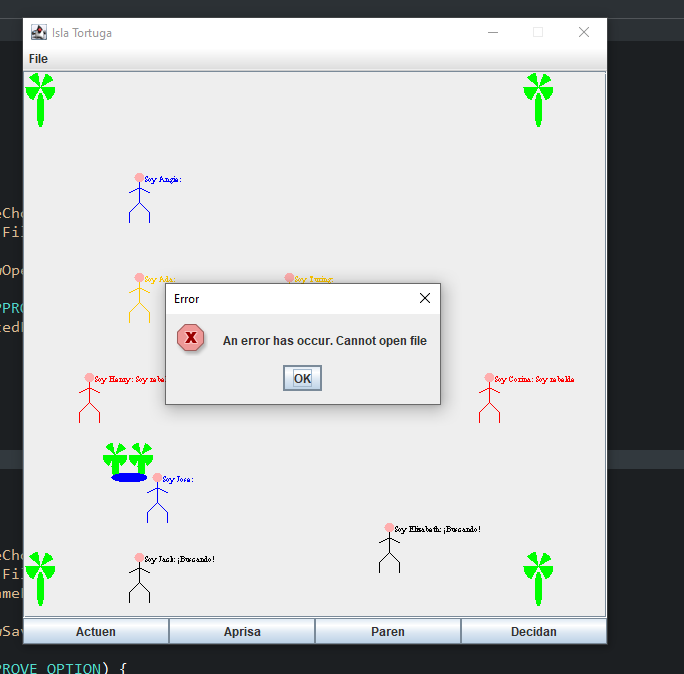
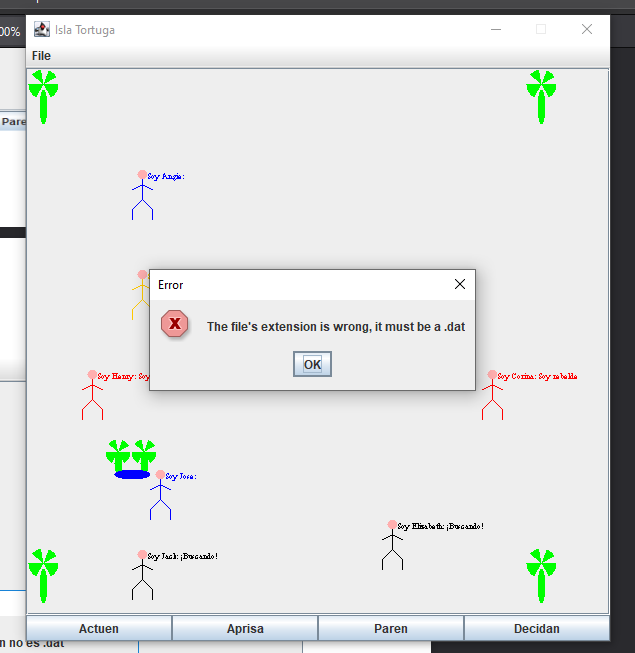
Los elementos vuelven a las posiciones que tenían al momento en el que se guardo la partida, sin embargo, los colores y el estado de sus extremidades están en su estado inicial.

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

La diferencia entre estos es que con el primer comportamiento obtenemos todos los elementos tal cual estaban al momento en el que se guardó la partida y en el segundo comportamiento los elementos aparecen con sus nuevas posiciones, pero no con otras características que tenían al momento de guardar la partida, como por ejemplo el color, las palabras y las extremidades (pues esta información no se guardó en el txt)

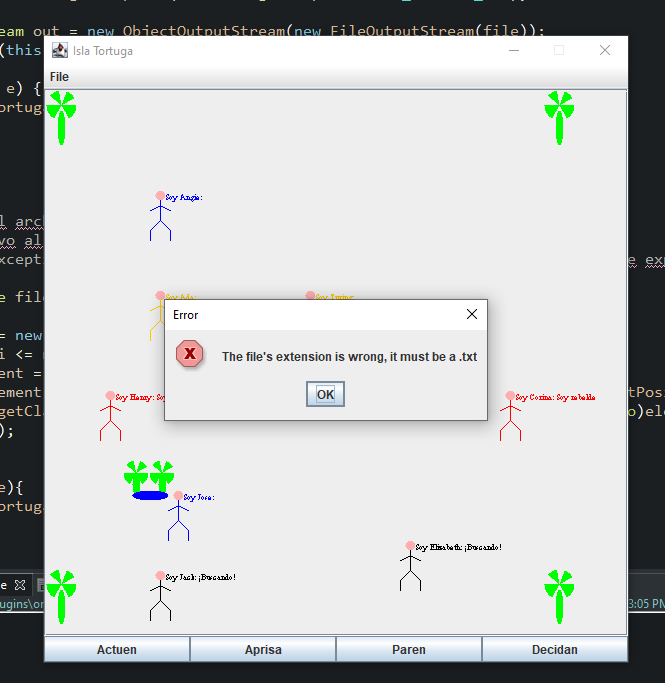
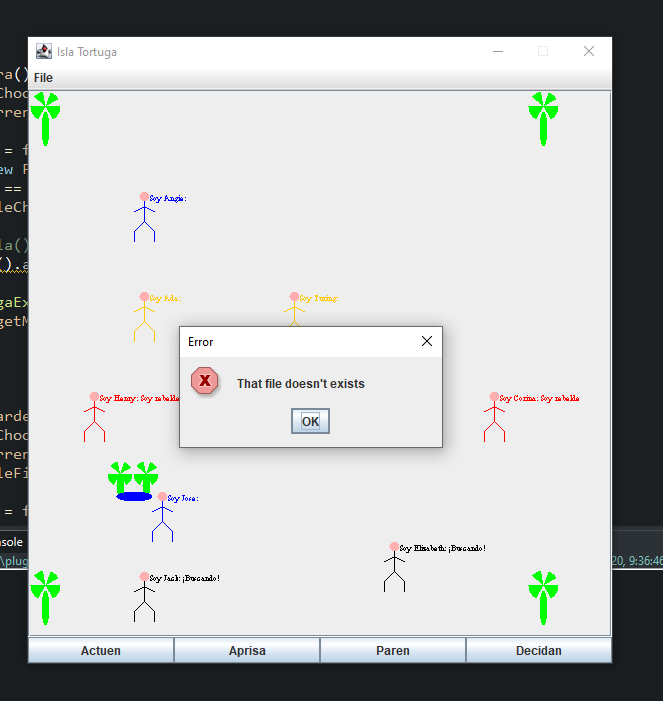
**Perfeccionando salvar y abrir**

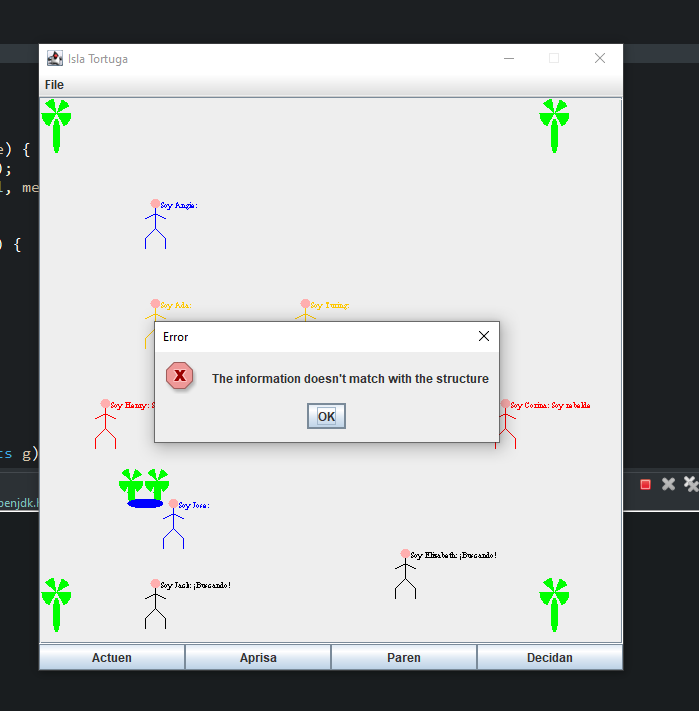
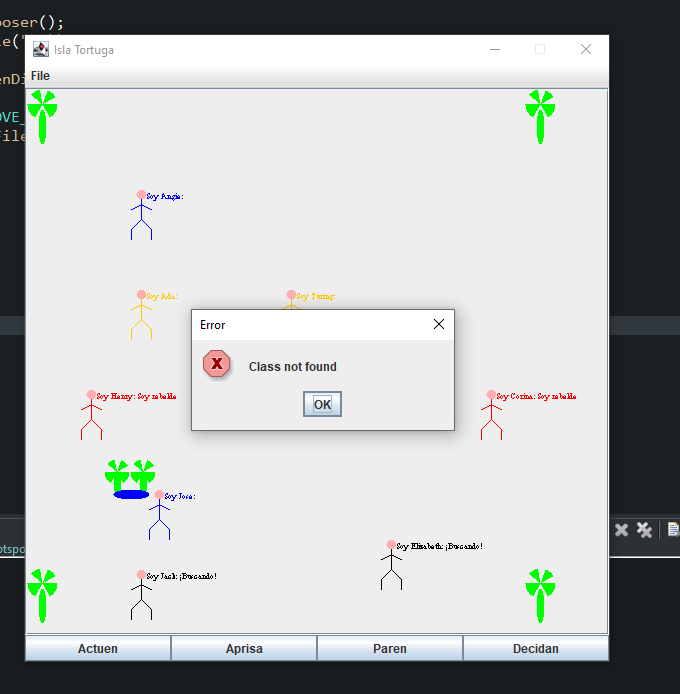
1. Copien las versiones actuales de abra y guarde y renómbrenlos como abra01 y guarde01
2. Perfeccionen el manejo de excepciones de los métodos abra y guarde detallando los errores.
3. Realicen una prueba de aceptación para validar cada uno de los nuevos mensajes diseñados, ejecútenla y capturen la pantalla final



**Perfeccionando importar y exportar**

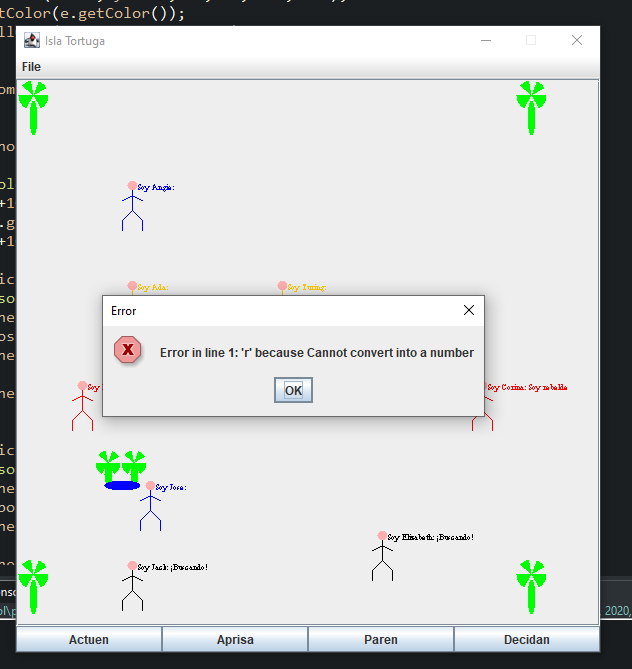
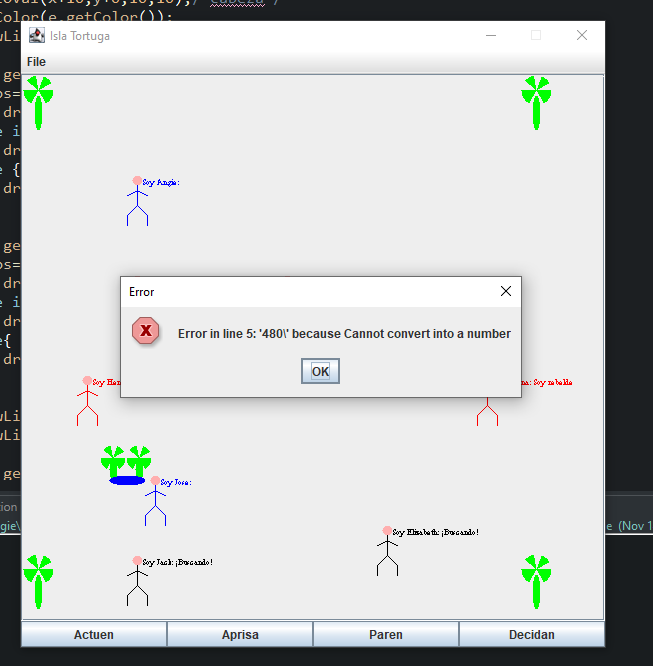
1. Copien las versiones actuales de importe y exporte y renómbrenlos como importe01 y exporte01
2. Perfeccionen el manejo de excepciones de los métodos importe y exporte detallando los errores.
3. Realicen una prueba de aceptación para validar cada uno de los nuevos mensajes diseñados, ejecútenla y capturen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

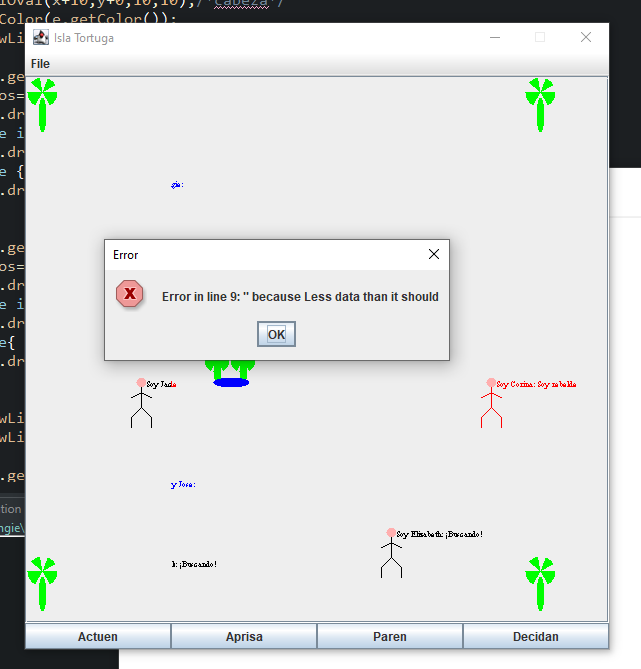
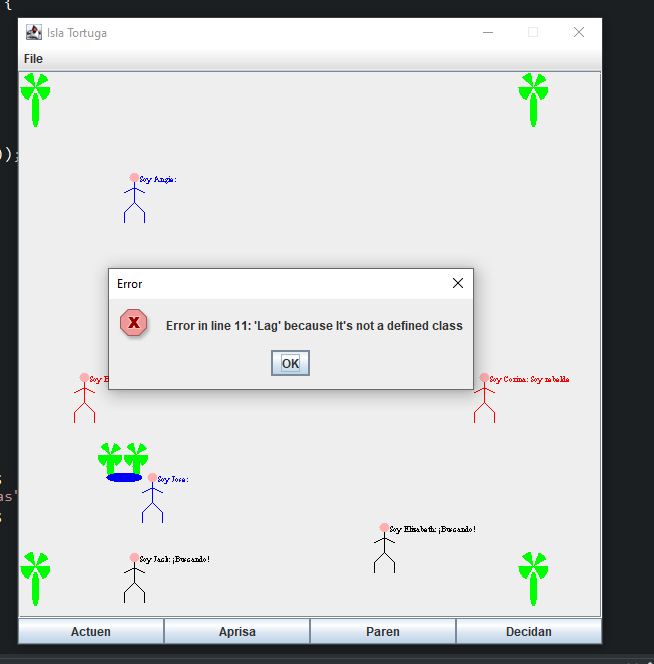
 

**Perfeccionando importar, Hacia un minicompilador**

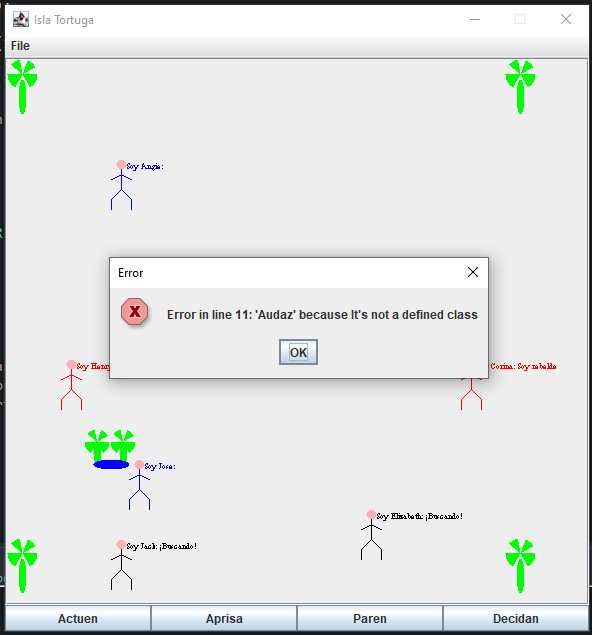
1. Copien las versiones actuales de importe y exporte y renómbrenlos como importe02 y exporte02
2. Perfeccionen el método importe para que, además de los errores generales, en las excepciones indique el detalle de los errores encontrados en el archivo (como un compilador) : número de línea donde se encontró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

**Perfeccionando importar, Hacia un minicompilador flexible**

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 islaTortugaErrG.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/Nombre)

Cada uno invirtió aproximadamente 9 horas. 9 hr / Angie Medina – Jose Perez

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

Se logro terminar el laboratorio, porque le invertimos el tiempo necesario

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

Creemos que ambas practicas son muy importantes cuando se está realizando alguna actividad, pero si escogemos una seria la de *All code must pass all unit tests before it can be released* porque lo básico es que una aplicación funcione

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

Implementar el último punto, porque no habíamos escuchado de *reflection* antes

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

No se presentó ningún problema técnico

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

Consideramos que nos comunicamos bien como equipo y así mismo tratamos de implementar las practicas xp, pero igual nos comprometemos a mejorar en la implementación de las practicas ya vistas en pasadas entregas junto con las nuevas vistas en este laboratorio.