**Laboratorio 06. POOB-02**

Jefer Alexis González Romero  
Angel Nicolas Cuervo Naranjo

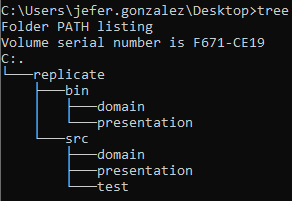
**DESARROLLO**

**Preparando**

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

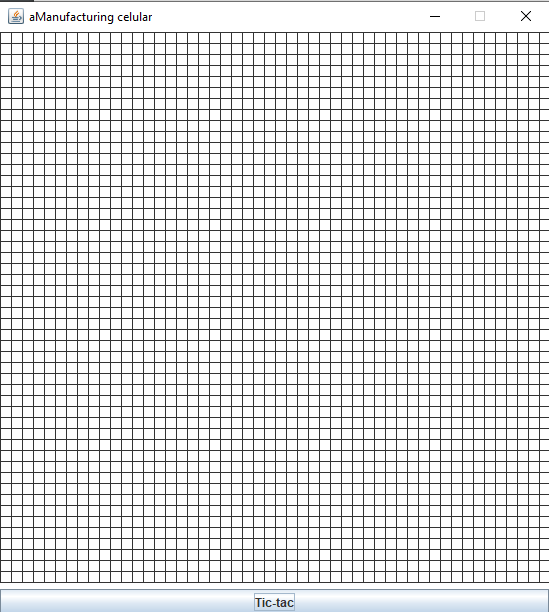
**1.** En su directorio descarguen la versión del proyecto realizado por ustedes para el laboratorio 03 y preparen el ambiente para trabajar desde CONSOLA.

Se descargó el proyecto, se creó los directorios correspondientes del proyecto, y se copiló.



**2.** Ejecuten el programa, revisen la funcionalidad.

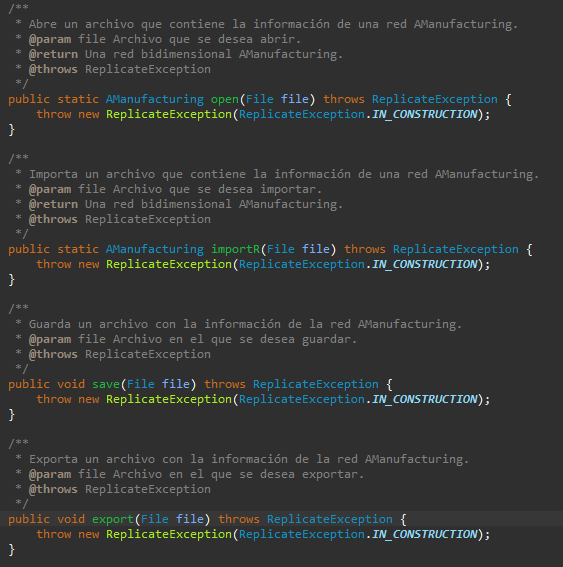




**Creando la maqueta**

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 ReplicateException con el mensaje genérico de “Opción en construcción”. Los métodos deben tener un parámetro File



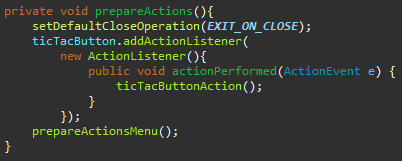
**2. VISTA:** Construyan un menú barra que ofrezca, además de las opciones básicas de entrada-salida, las opciones estándar de nuevo y salir (Nuevo, Abrir, Guardar como, Importar, Exportar como, Salir). No olviden incluir los separadores. Para esto creen el método prepareElementosMenu. Únicamente debe funcionar la vista. Capturen la pantalla correspondiente.

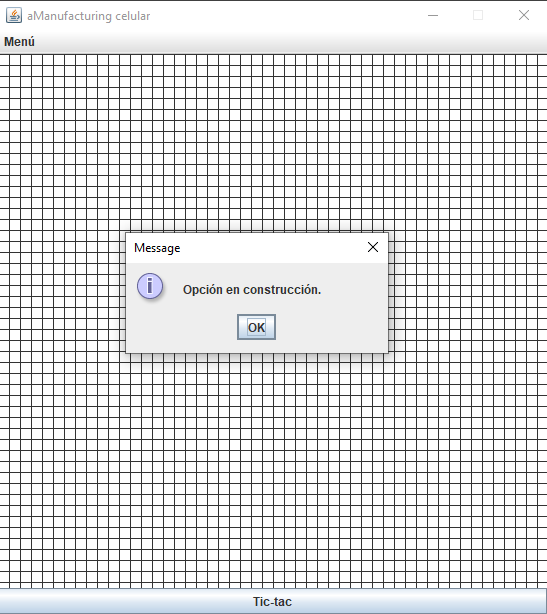
Texto

Descripción generada automáticamente Imagen que contiene Patrón de fondo

Descripción generada automáticamente

**3. 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 dominio (usen nulo como parámetro) . No incluyan todavía el FileChooser. Capturen una pantalla significativa.

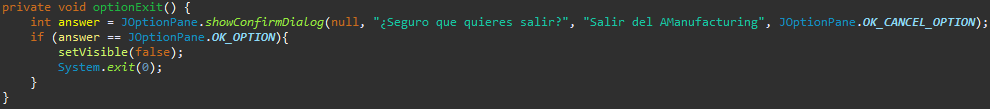
Texto

Descripción generada automáticamente

**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 dominio y el segundo sí.

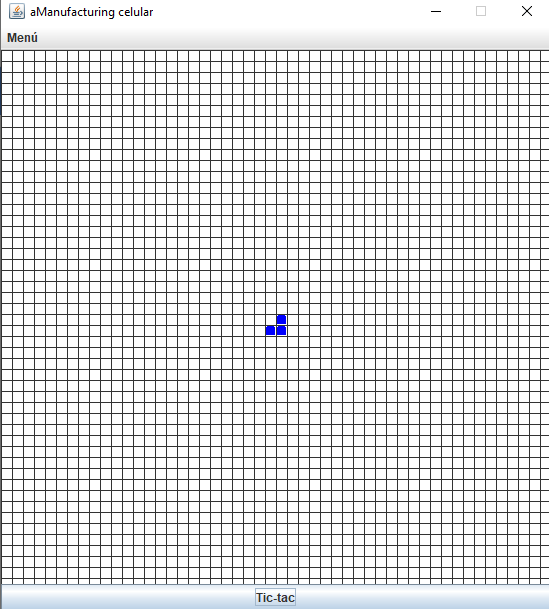
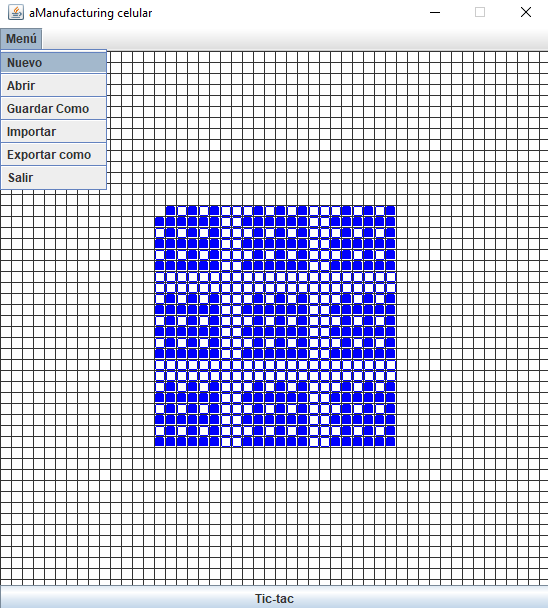
**1.** Construyan el método opcionSalir que hace que se termine la aplicación. No es necesario incluir confirmación.



**2.** Construyan el método opcionNuevo que crea un nuevo replicate. Capturen una pantalla significativa.

Texto

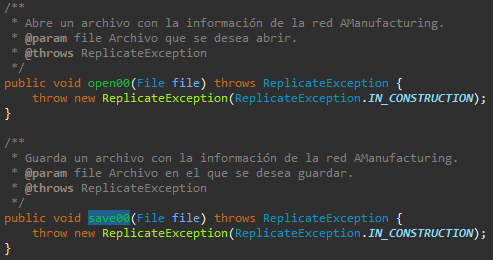
Descripción generada automáticamente



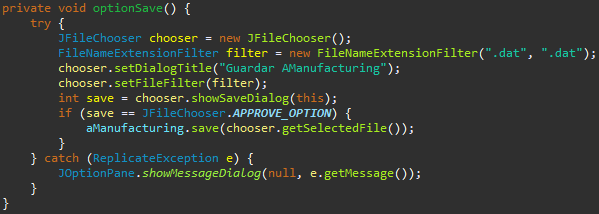
**Implementando salvar y abrir**

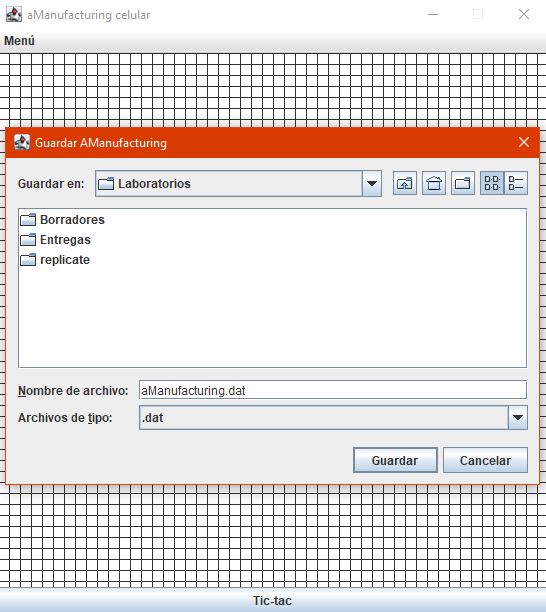
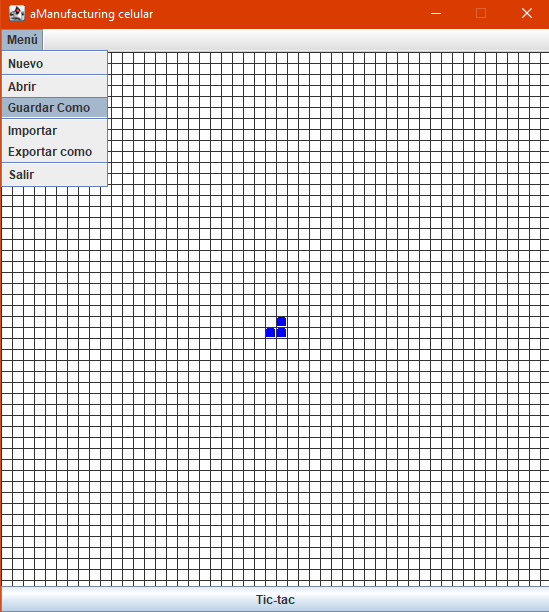
Las opciones salvar y abrir van a ofrecer servicios de persistencia del replicate como objeto. Los nombres de los archivos deben tener como extensión .dat.

**1.** Copien las versiones actuales de abra y guarde y renómbrenlos como abra00 y guarde00.



**2.** 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.



Tabla

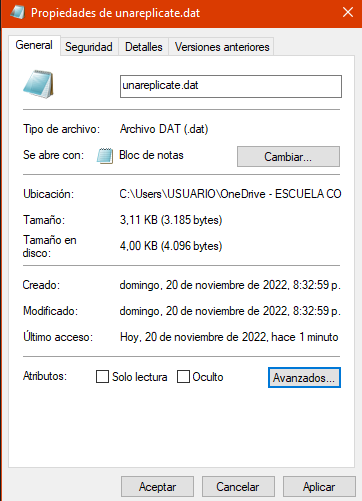
Descripción generada automáticamente

**3.** Construyan el método guarde que ofrece el servicio de guardar en un archivo el estado actual del replicate.

Texto

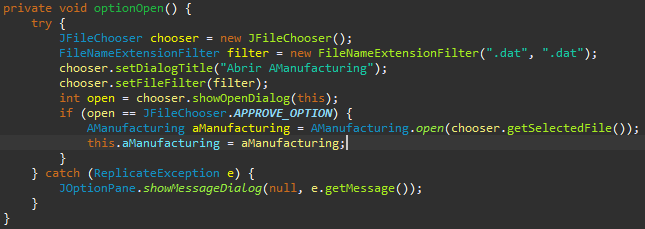
Descripción generada automáticamente

**4.** Validen este método guardando el estado obtenido después de dos clics como unareplicate.dat. ¿El archivo se creó en el disco? ¿Cuánto espacio ocupa?

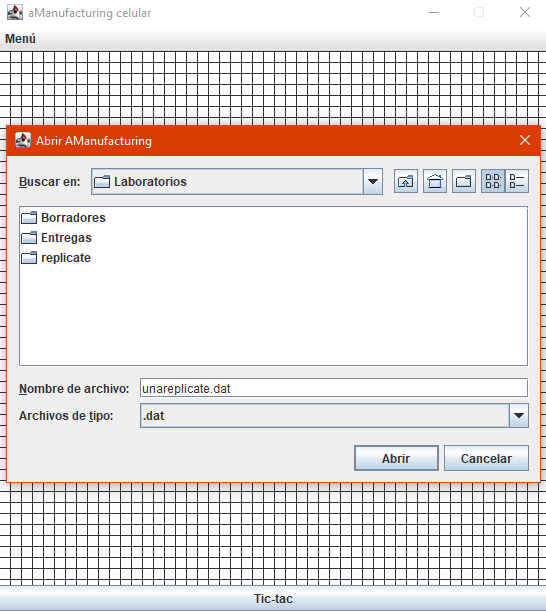


Si se creó en el disco y ocupa 4,096 bytes.

**5.** 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.

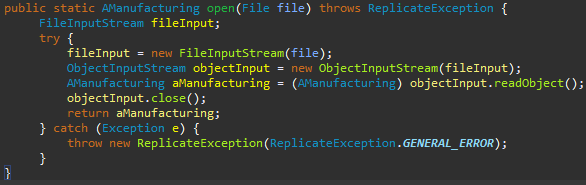


Tabla

Descripción generada automáticamenteGráfico

Descripción generada automáticamente con confianza media

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



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

**1.** Se inicia la aplicación con una red *AManufacturing* que tenga tres celdas *Rfplicbte* activas ubicadas en el centro como se mostró en la figura del laboratorio 3.

Imagen que contiene Tabla

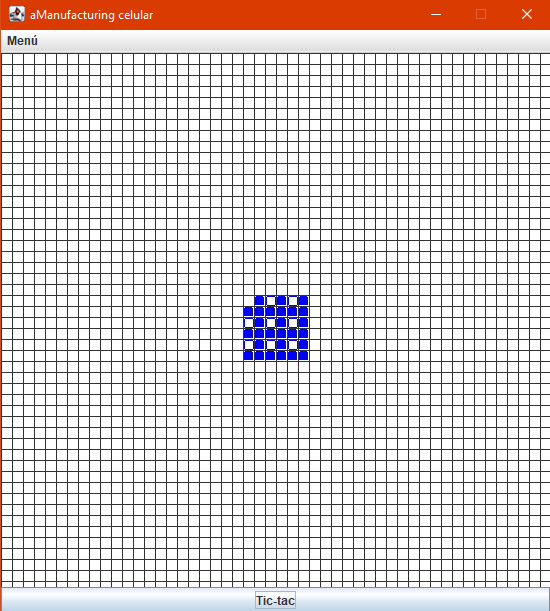
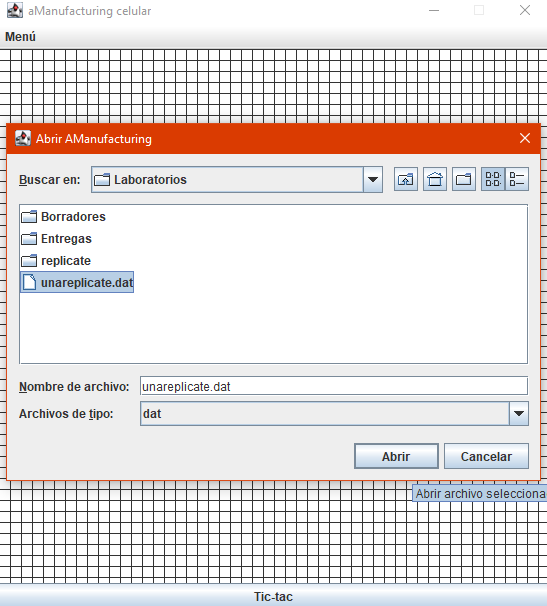
Descripción generada automáticamente

**2.** Se hacen 5 *Tic-Tac.*

*Imagen que contiene Tabla

Descripción generada automáticamente*

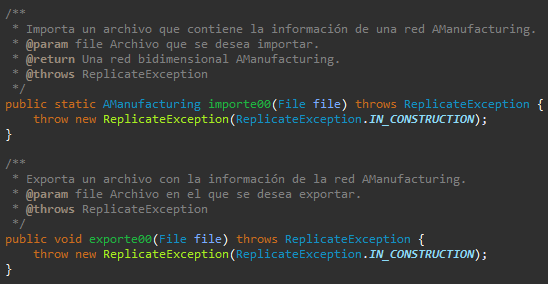
**3.** Se abre el menú y con la opción *Abrir* se abre el archivo *unareplicate.dat* que fue guardado en el inciso 4.

**

**Implementando importar y exportar**

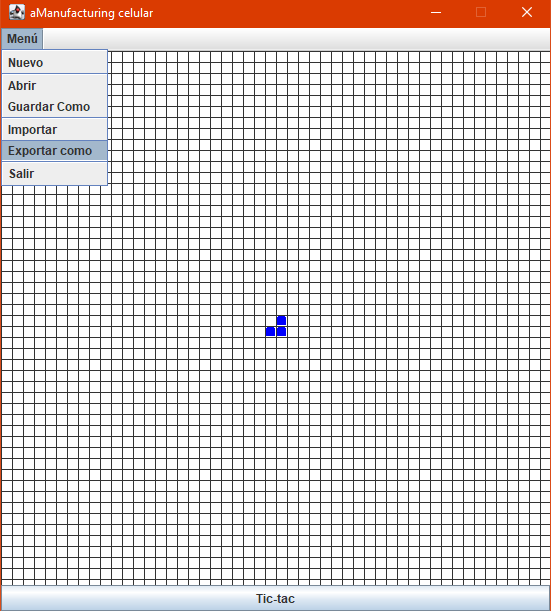
Estas operaciones nos van a permitir importar información del replicate desde un archivo de texto y exportarlo. Los nombres de los archivos de texto deben tener como extensión .txt Los archivos texto tienen una línea de texto por cada elemento En cada línea asociada un elemento se especifica el tipo y la posición. Inquieta 100 100 Bombillo 500 500

**1.** Copien las versiones actuales de importe y exporte y renómbrenlos como importe00 y exporte00.



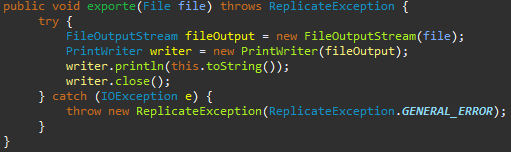
**2.** Construyan el método opcionExportar que une de forma adecuada la capa de presentación con la capa de dominio. Ejecuten la aplicación y capturen una pantalla significativa.

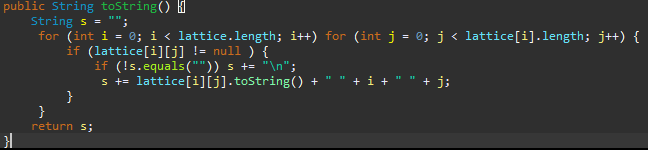


Gráfico

Descripción generada automáticamente

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





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

1. Se inicia la aplicación con una red *AManufacturing* que tenga tres celdas *Rfplicbte* activas ubicadas en el centro como se mostró en la figura del laboratorio 3.

Imagen que contiene Tabla

Descripción generada automáticamente

1. Se hacen 3 *Tic-tac.*

Imagen que contiene Gráfico

Descripción generada automáticamente

1. Se abre el menú y con la opción *Exportar como* se exporta con el nombre de *unareplicat.txt*

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto

Descripción generada automáticamente

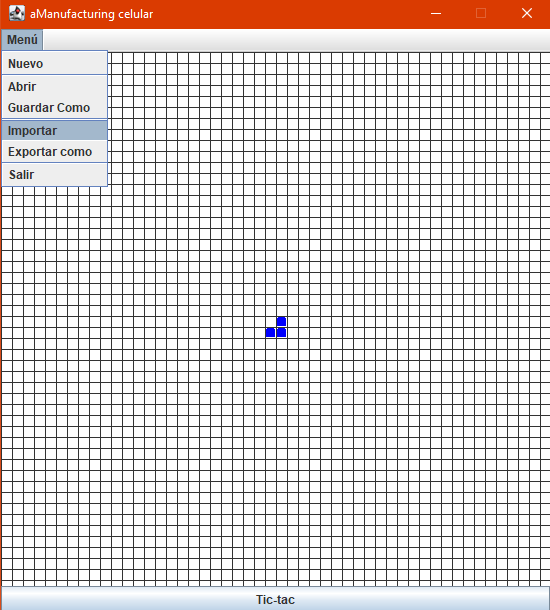
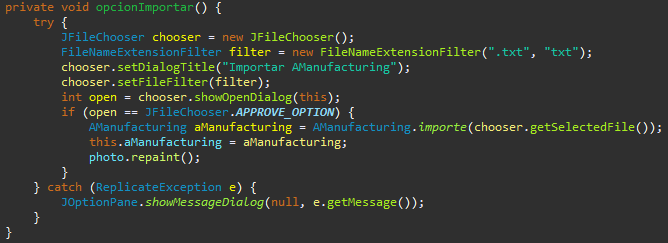
1. Se edita el archivo incluyendo la siguiente línea: *Kriptonita 21 21*

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

Las 63 células obtenidas después de los 3 Tic-tac se guardan en el archivo cada una con su respectivo tipo que es *Rfplicbte,* con su fila y columna separados por un espacio.

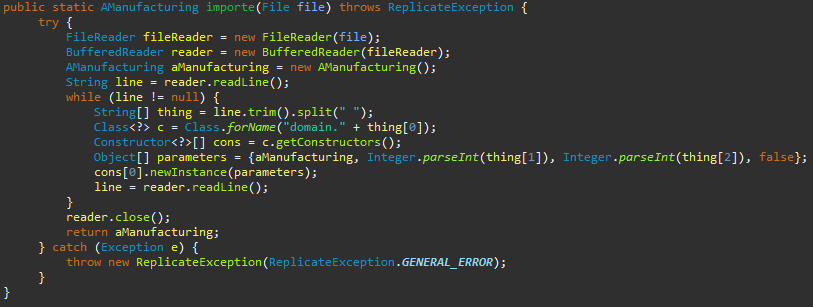
**5.** Construyan el método opcionImportar que une de forma adecuada la capa de presentación con la capa de dominio. Ejecuten la aplicación y capturen una pantalla significativa.

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamenteGráfico

Descripción generada automáticamente con confianza media

**6.** 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)



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

1. Se realiza la misma prueba de aceptación propuesta en el inciso 4.
2. Se inicia la aplicación con una red *AManufacturing* que tenga tres celdas *Rfplicbte* activas ubicadas en el centro como se mostró en la figura del laboratorio 3.

Imagen que contiene Tabla

Descripción generada automáticamente

1. Se abre el menú y con la opción *Importar* se importa el archivo de texto *unareplicat.txt*

Tabla

Descripción generada automáticamenteInterfaz de usuario gráfica, Tabla

Descripción generada automáticamenteImagen de la pantalla de una jaula

Descripción generada automáticamente con confianza baja

Se obtiene el *AManufacturing* que se había exportado junto con la edición que se le hizo al archivo, que le añadió una célula *Kriptonita* en la posición (21, 21). Pero sin los estados que tenía cada célula al momento de exportarlo.

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

1. Se crea un archivo de texto llamado *otrareplicate.txt* con la siguiente información:

Kriptonita 24 24, Rfplicbte 25 25, Rfplicbte 25 24, Rfplicbte 24 25.

**Texto, Tabla

Descripción generada automáticamente**

1. Se inicia la aplicación con una red *AManufacturing* que no tenga ningún *Thing* y Se abre el menú y con la opción *Importar* se importa el archivo de texto *otrareplicat.txt*

Tabla

Descripción generada automáticamenteAplicación, Tabla

Descripción generada automáticamenteImagen de la pantalla de un celular

Descripción generada automáticamente con confianza baja

Se obtienen las células escritas en el archivo de texto, en las posiciones dadas y con estados inactivos.

**Analizando comportamiento**

Imagen que contiene Gráfico

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

Al hacer esto se obtiene todas las células que se habían obtenido tras los 3 *Tic-tac* con sus respectivos estados al momento de guardar la red *AManufacturing.*

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

Imagen que contiene biombo, edificio, jaula

Descripción generada automáticamente

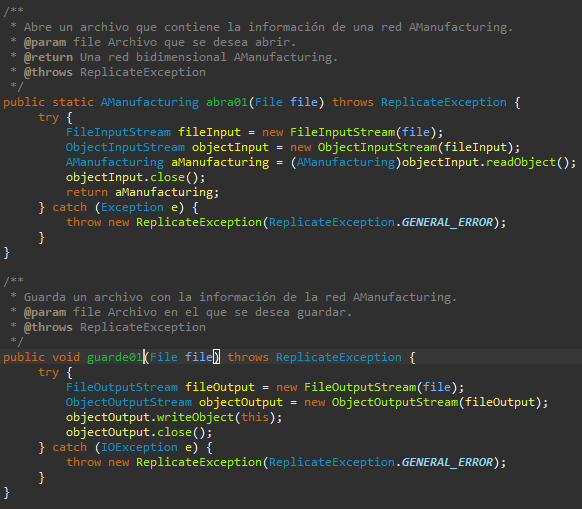
La red resultante de exportar e importar el archivo tiene las mismas células de la red original y en las mismas posiciones, pero no con el mismo estado de estas, ya que todas están inactivas y en la que se exportó había activas.

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

La diferencia es que con guardar y abrir se guardan los objetos con su estado, mientras que en exportar e importar solo se está teniendo el nombre del *Thing* y la posición de este, dejando de lado otros aspectos como el estado en el que se encontraba al momento de exportar.

**Perfeccionando salvar y abrir**

**1.** Copien las versiones actuales de abra y guarde y renómbrenlos como abra01 y guarde0.



**2.** Perfeccionen el manejo de excepciones de los métodos abra y guarde detallando los errores.

Captura de pantalla de computadora

Descripción generada automáticamente

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

1. Se crea un archivo .dat vacío y lo guardamos.

Tabla

Descripción generada automáticamente con confianza media

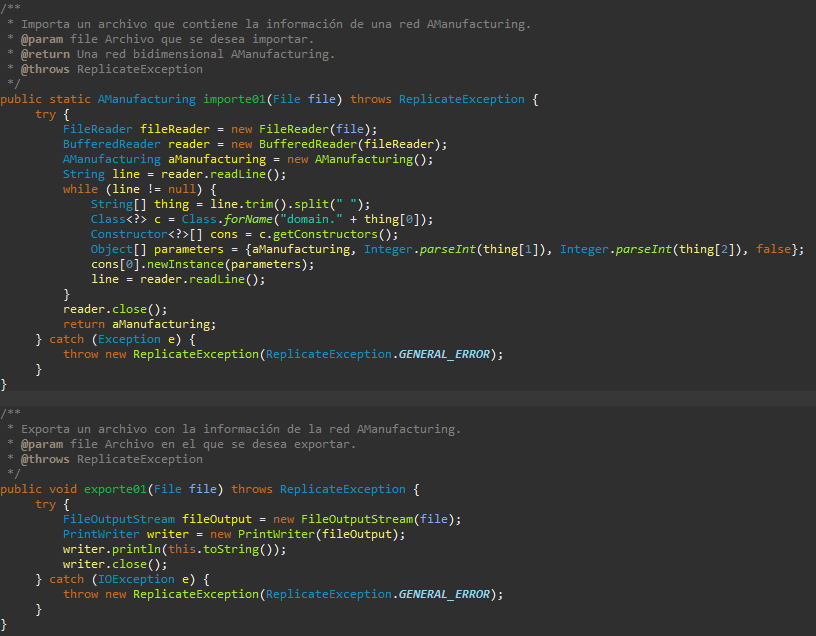
1. Se inicia la aplicación, se abre el menú y con la opción *Abrir* se abre el archivo .dat que está vacío, aparecerá un mensaje indicando el error.

Interfaz de usuario gráfica, Aplicación, Sitio web

Descripción generada automáticamente

**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.

Texto

Descripción generada automáticamente**Texto

Descripción generada automáticamente**

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

1. Se crea un archivo .txt con la siguiente línea “Kriptonita 25 doce” y lo guardamos.

Tabla

Descripción generada automáticamente

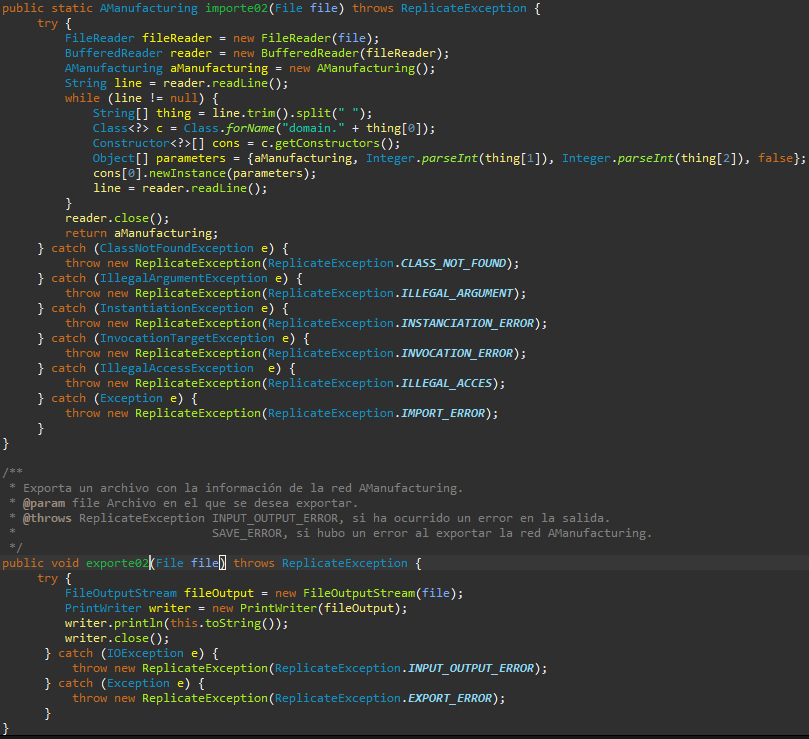
1. Se inicia la aplicación, se abre el menú y con la opción *Importar s*e importa el archivo .txt que se creó anteriormente, aparecerá un mensaje indicando el error.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**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.

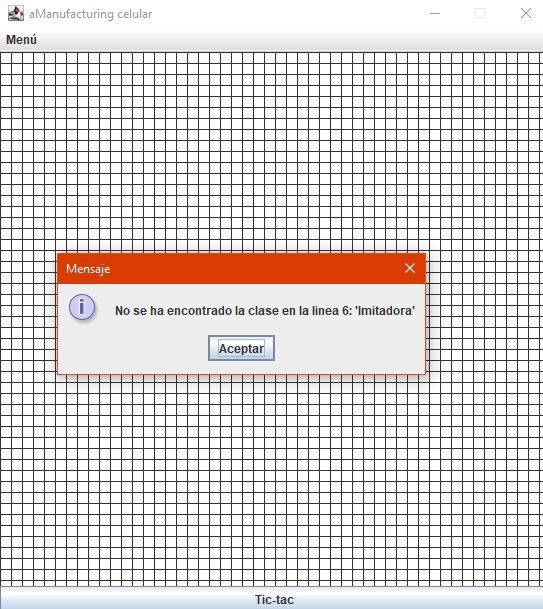
Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamenteTexto

Descripción generada automáticamente

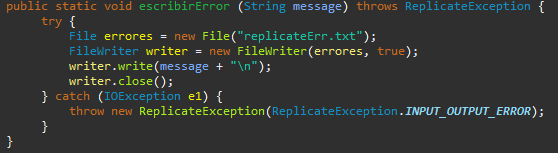
Texto

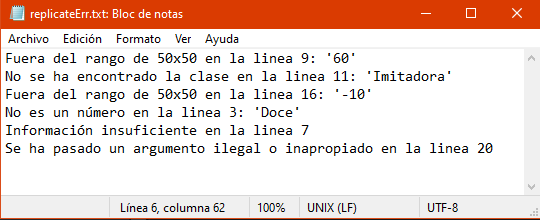
Descripción generada automáticamente

**3.** Escriban otro archivo con errores, llámelo replicateErr.txt, para ir arreglándolo con ayuda de su “importador”. Presente las pantallas que contengan los errores.

Texto

Descripción generada automáticamente





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

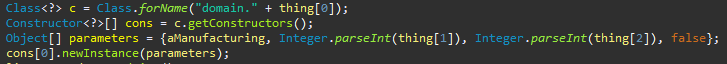
**1.** Copien las versiones actuales de importe y exporte y renómbrenlos como importe03 y exporte03

Texto

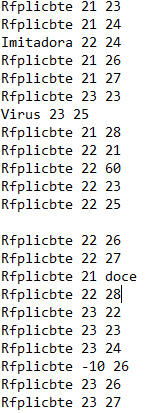
Descripción generada automáticamente

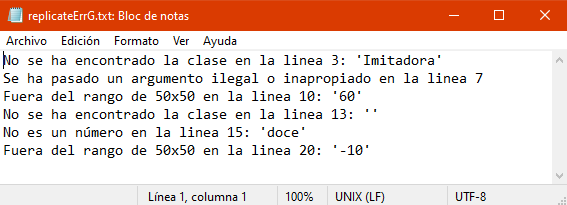
**2.** Perfeccionen los métodos importe y exporte para que pueda servir para cualquier tipo de elementos creados en el futuro (Investiguen cómo crear un objeto de una clase dado su nombre)

Ya se había hecho desde un principio el uso del método forName de Class.



**3.** Escriban otro archivo de pruebas, llámelo replicateErrG.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)

10 horas por cada uno.

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

Completo.

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

Fue más útil *refactor* ya que eliminamos la redundancia, eliminamos la funcionalidad no utilizada ayudándonos a ahorrar tiempo y aumentar la calidad.

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

El mayor logro fue el minicopilador, ya que su implementación fue complicada.

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

El mayor problema técnico fue instanciar un objeto dado el nombre de la clase, para resolverlo investigamos e intentamos de diferentes formas el uso del método *forName* de *Class.*

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

Destinar el tiempo suficiente, nos comprometemos a mejorar la comunicación