**GUILLERMO AFONSO CASTRO**

**LEONARDO GALEANO**

**POOB 2020-1**

**LAB06**

**PROGRAMACIÓN ORIENTADA A OBJETOS**

**I/O**

**2020-01**

**Laboratorio 6/6 [ :) ]**

# OBJETIVOS

1. Completar el código de un proyecto considerando requisitos funcionales.
2. Diseñar y construir los métodos básicos de manejo de archivos: abrir, guardar como, importar y exportar.
3. Controlar las excepciones generadas al trabajar con archivos.
4. Experimentar las prácticas XP :  *Refactor whenever and wherever possible.*

All code must pass all [unit tests](http://www.extremeprogramming.org/rules/unittests.html) before it can be released. -

# ENTREGA

 Incluyan en un archivo .zip los archivos correspondientes al laboratorio. El nombre debe ser los dos apellidos de los miembros del equipo ordenados alfabéticamente.

 Deben publicar el avance al final de la sesión y la versión definitiva en la fecha indicada en los espacios preparados para tal fin.

 En el foro de entrega de avance deben indicar los logros y los problemas pendientes por resolver.

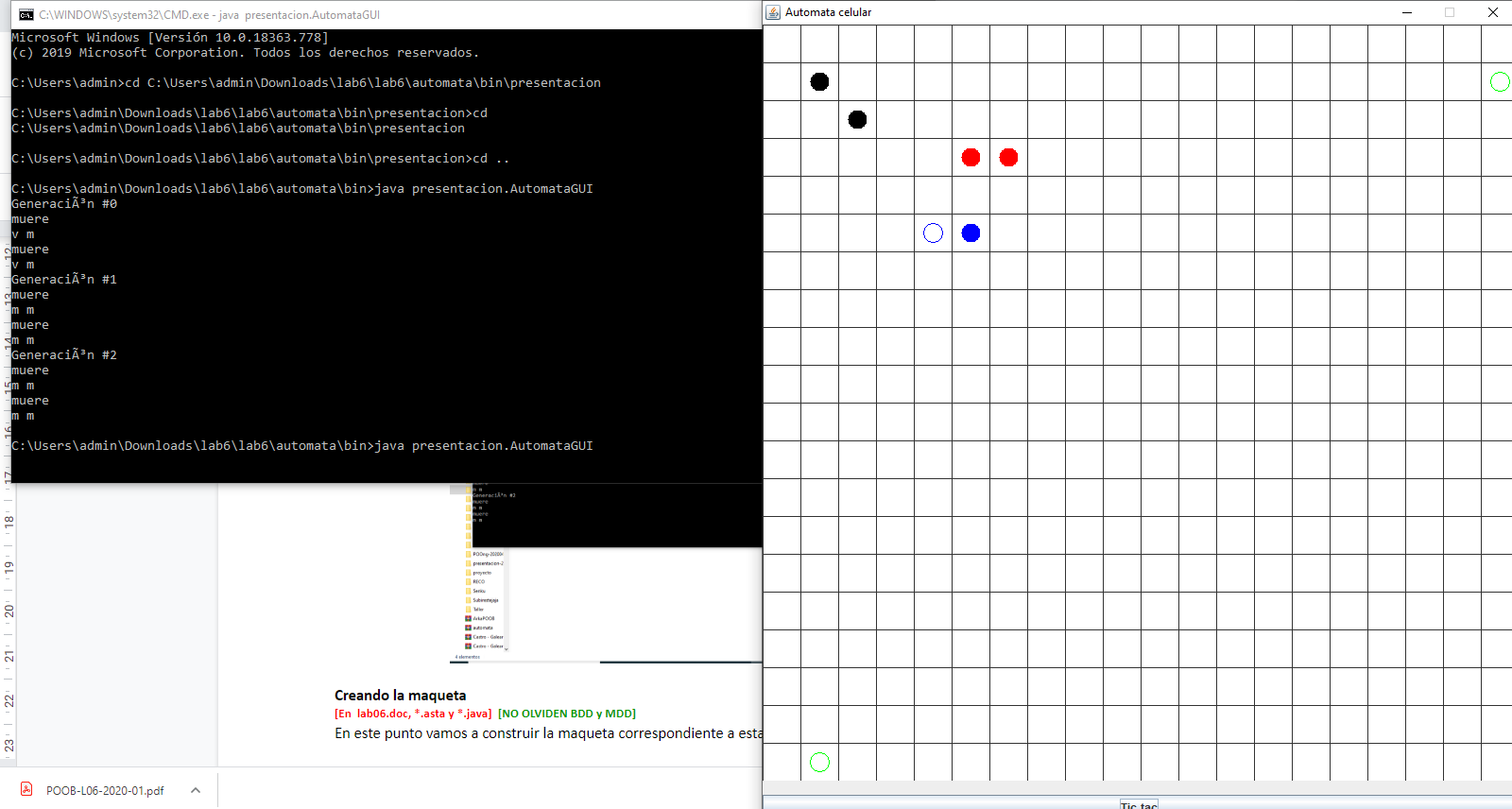
**DESARROLLO**

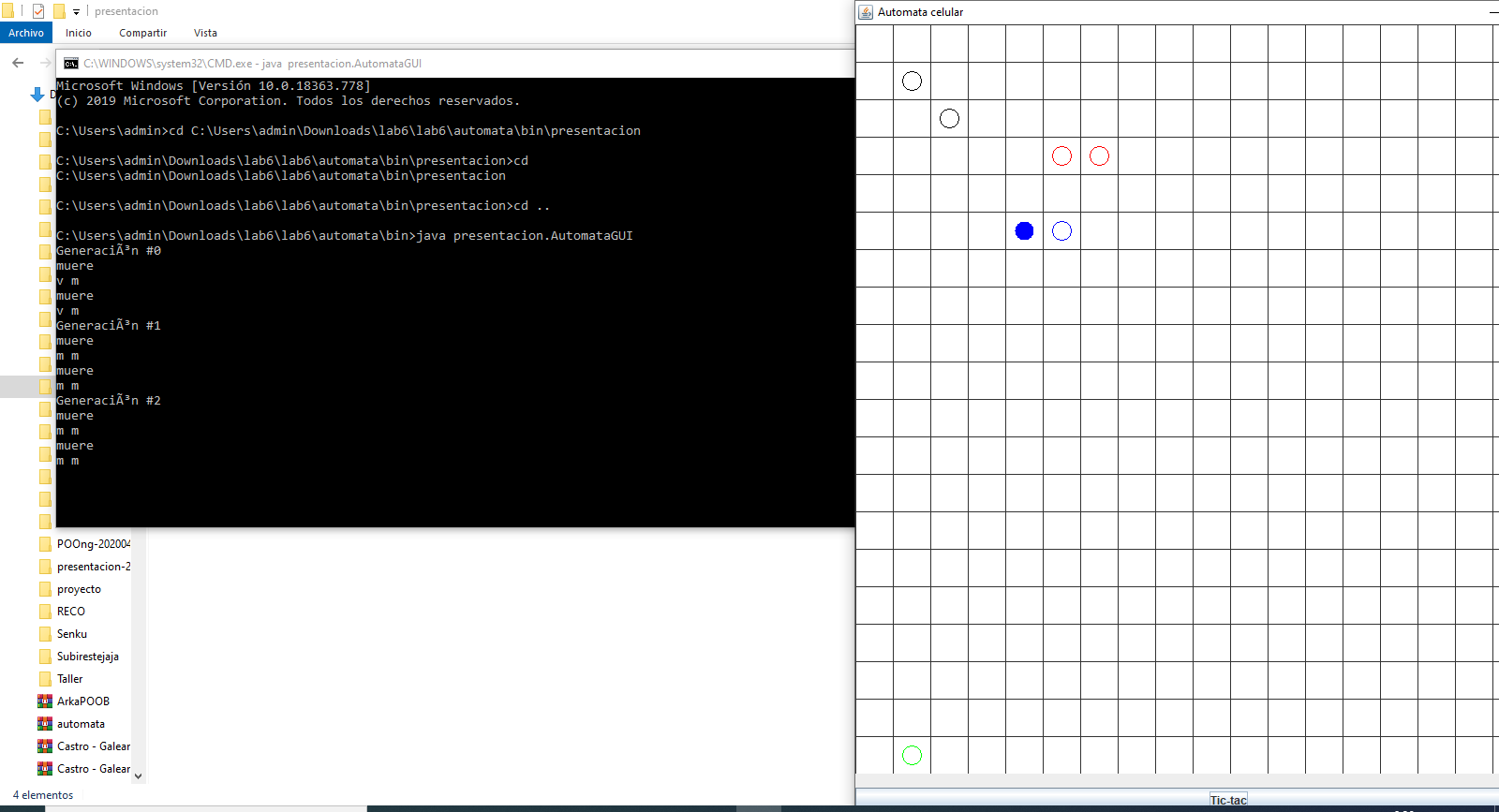
## Preparando

En este laboratorio vamos a extender el proyecto automata 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.

R/: tiene las funcionalidades de tictac ,muestra las células,su regeneración y podemos ver como se maneja la célula.



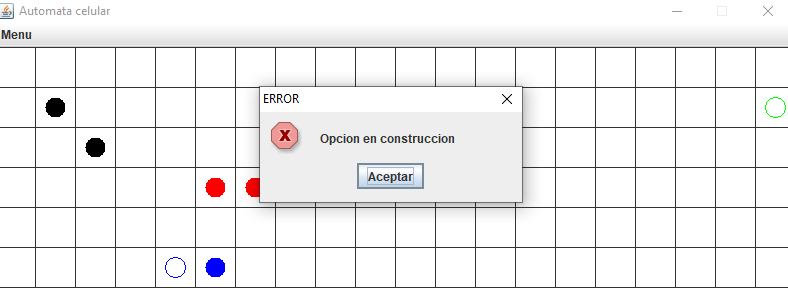


## Creando la maqueta

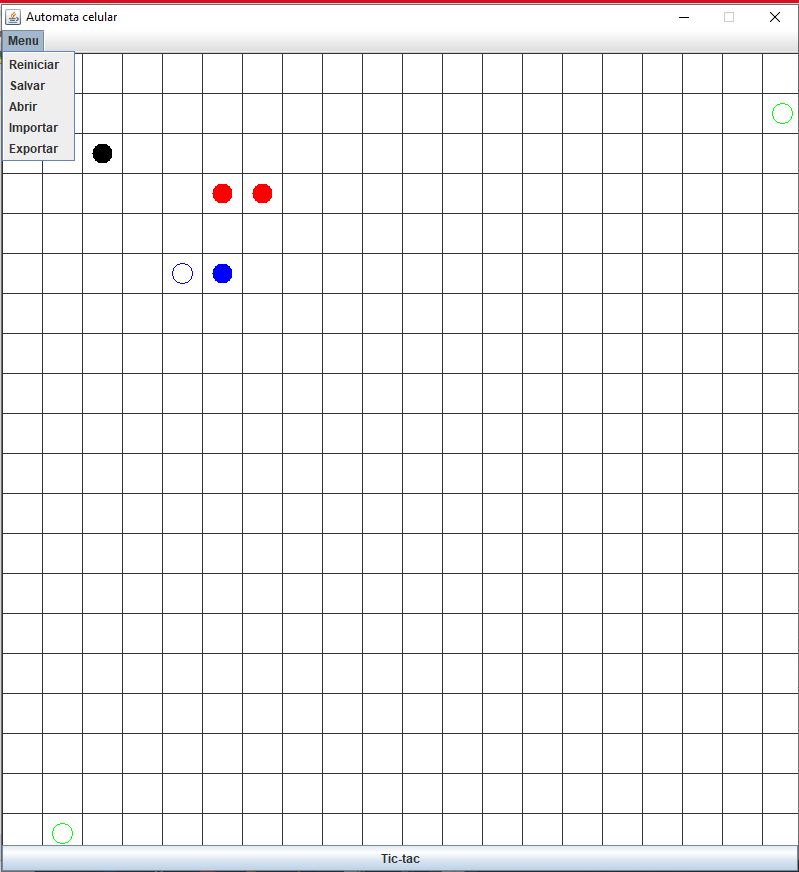
### [En lab06.doc, \*.asta y \*.java] [NO OLVIDEN BDD y MDD]

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 automataExcepcion con el mensaje: “Opción … en construcción”. Los métodos de entrada salida deben tener un parámetro File.

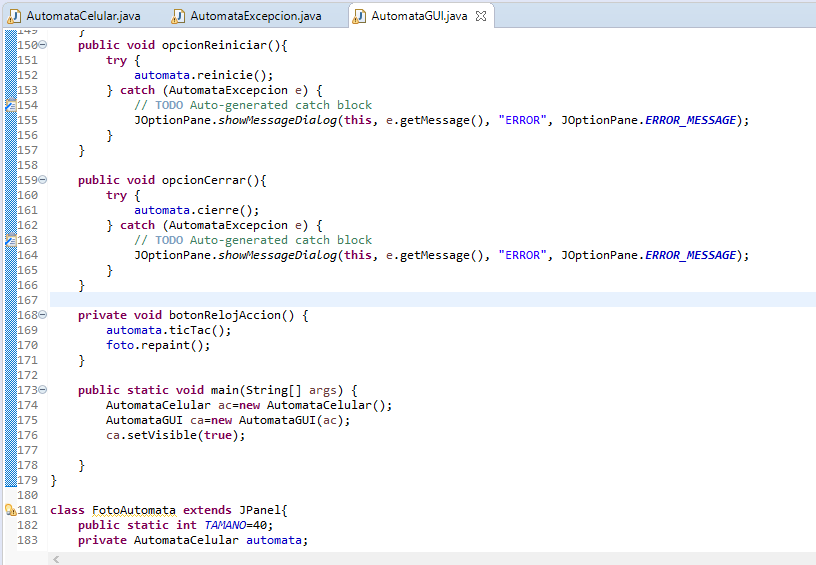


1. **VISTA :** Construyan un menú barra que ofrezca, además de las opciones básicas de entrada-salida, las opciones estándar de iniciar y salir. Para esto creen el método prepareElementosMenu. Capturen la pantalla correspondiente.



1. **CONTROLADOR:** Construyan los controladores correspondientes a estas acciones. Para esto creen el método prepareAccionesMenu y los métodos base del controlador ( opcionSalver,opcionAbir, opcionExportar, opcionImportar, opcionReiniciar, opcionCerrar), Estos métodos, por ahora, **llaman directamente el método correspondiente de la capa de aplicación**. Capturen una pantalla significativa.



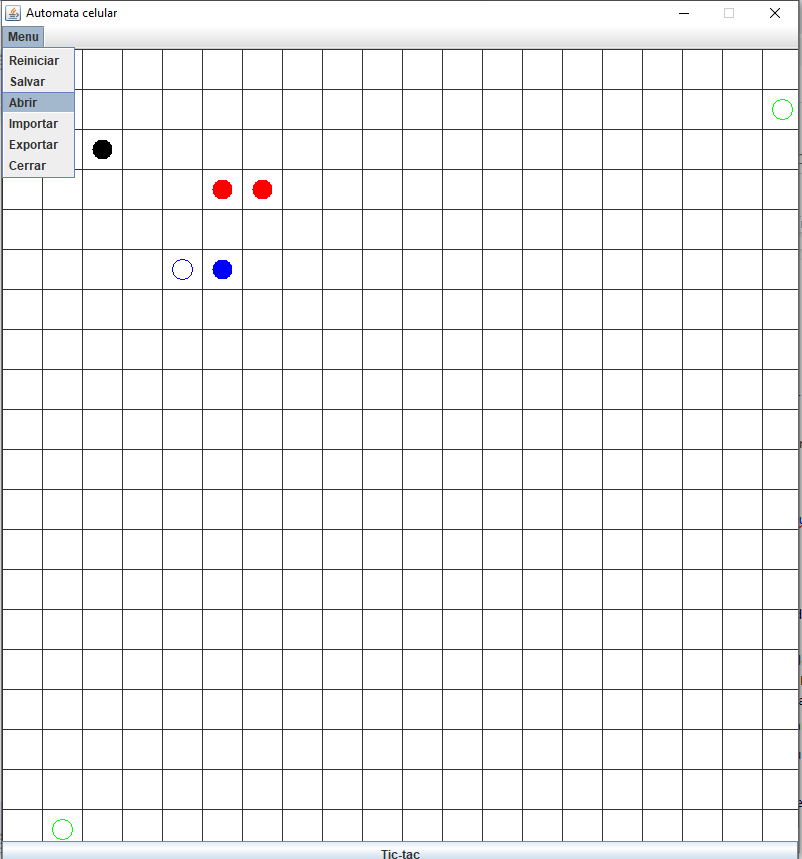


## Implementando salir e iniciar

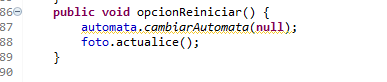
### [En lab06.doc, \*.asta y \*.java] [NO OLVIDEN BDD y MDD]

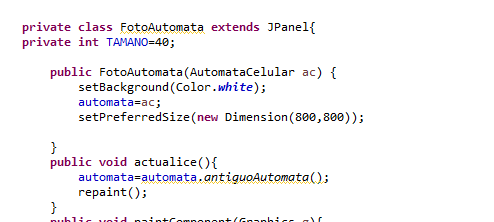
Las opciones salir e iniciar van a ofrecer los dos servicios estándar de las aplicaciones. El primero no requiere ir a capa de aplicación y el segundo sí.

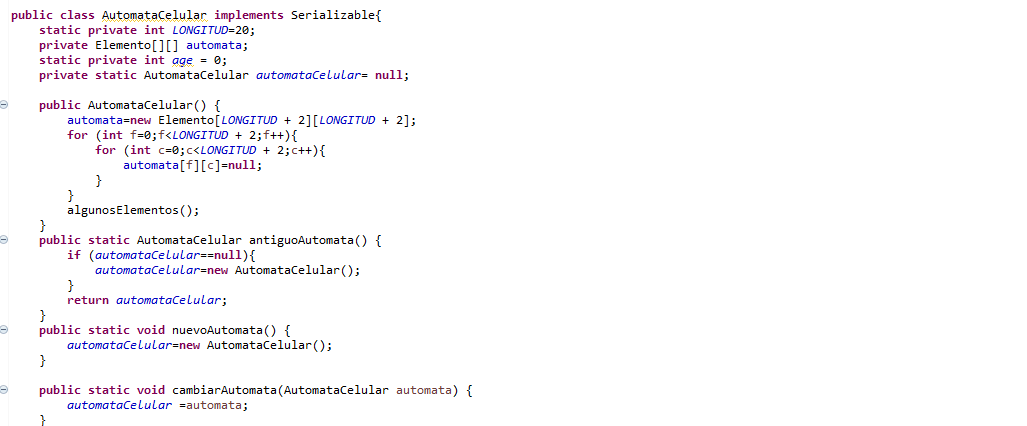
1. Construyan el método opcionSalir que hace que se termine la aplicación. No es necesario incluir confirmación.

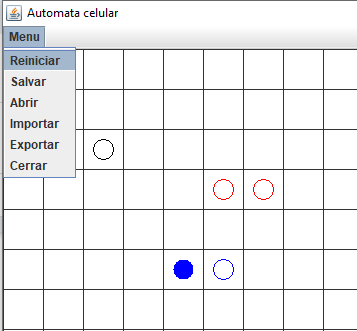


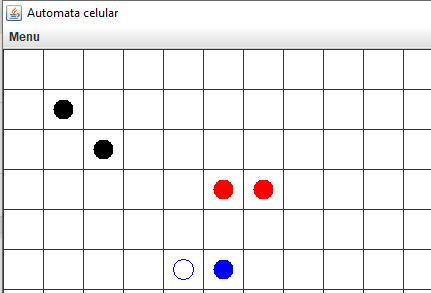
1. Construyan el método opcionIniciar que crea un nuevo automata. Capturen una pantalla significativa.









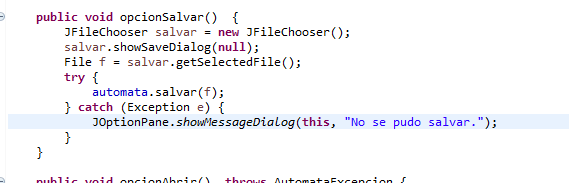


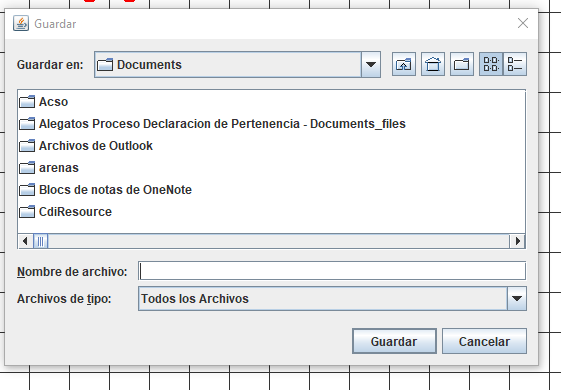
## Implementando salvar y abrir

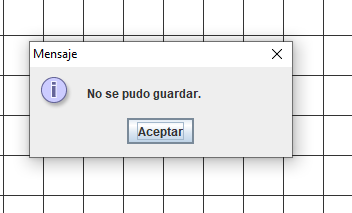
### [En lab06.doc, \*.asta y \*.java] [NO OLVIDEN BDD y MDD]

Las opciones salvar y abrir van a ofrecer servicios de persistencia de un automata 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.





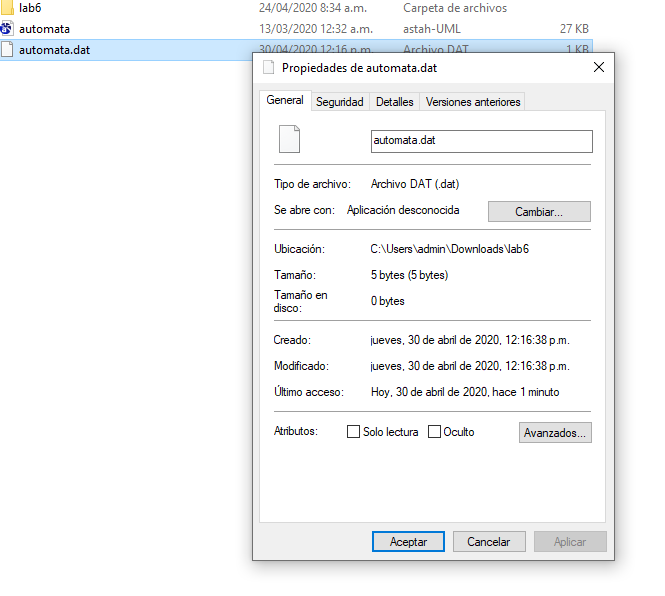


1. Construyan el método salve que ofrece el servicio de guardar en un archivo el estado actual del automata.

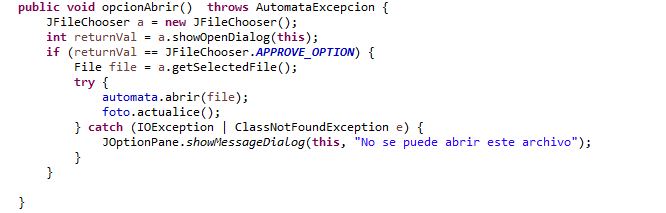
HECHO EN EL PROYECTO

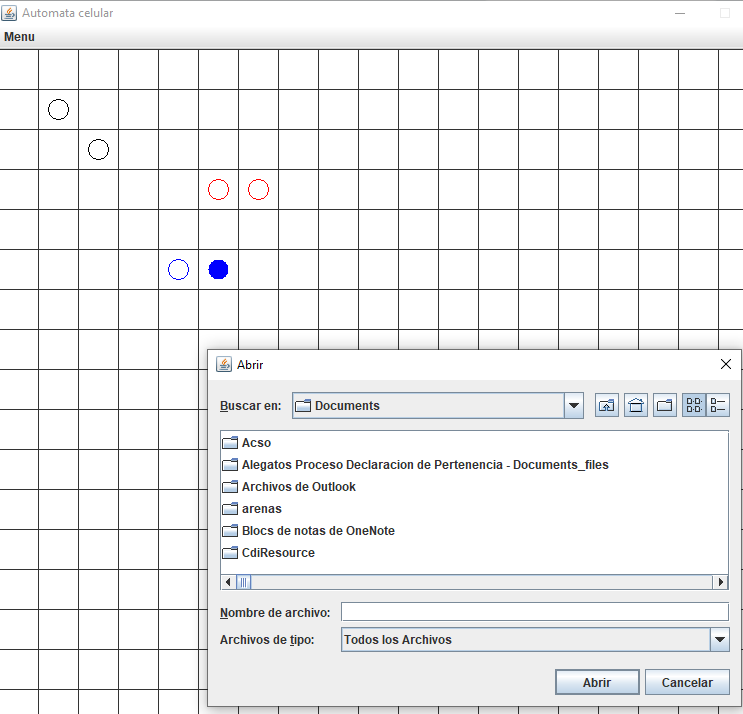
1. Validen este método guardando el automata inicial después de dos clics como unautomata.dat. ¿El archivo se creó en el disco? ¿Cuánto espacio ocupa?

RTA// si se creó en el disco y ocupa 1 kb.

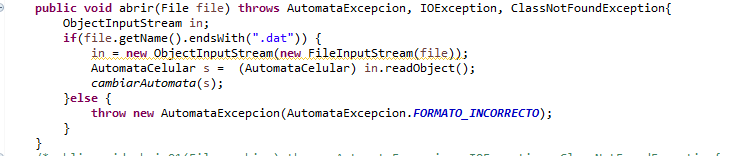


1. Construyan el método opcionAbrir que une de forma adecuada la capa de presentación con la capa de aplicación. Ejecuten la aplicación probando las diferentes opciones del FileChooser y capturen una pantalla significativa.

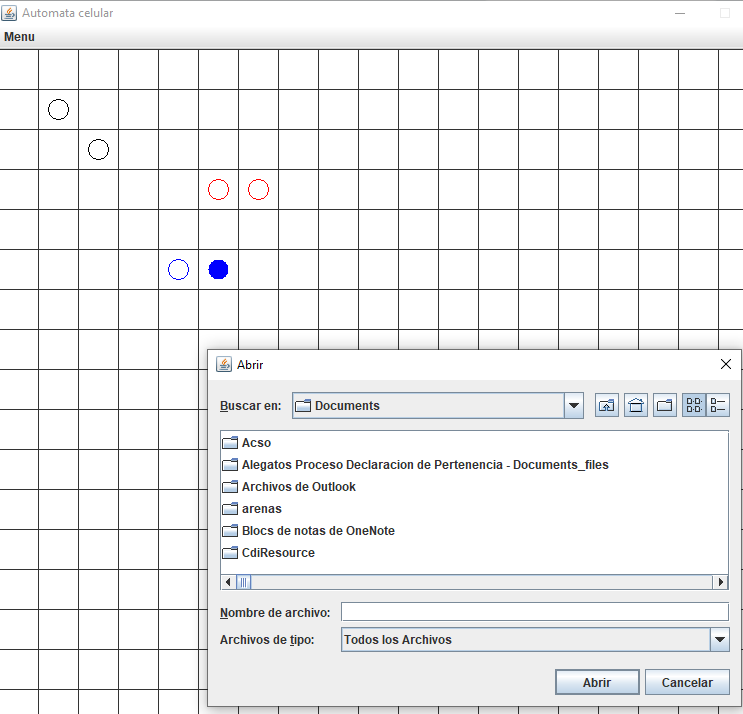


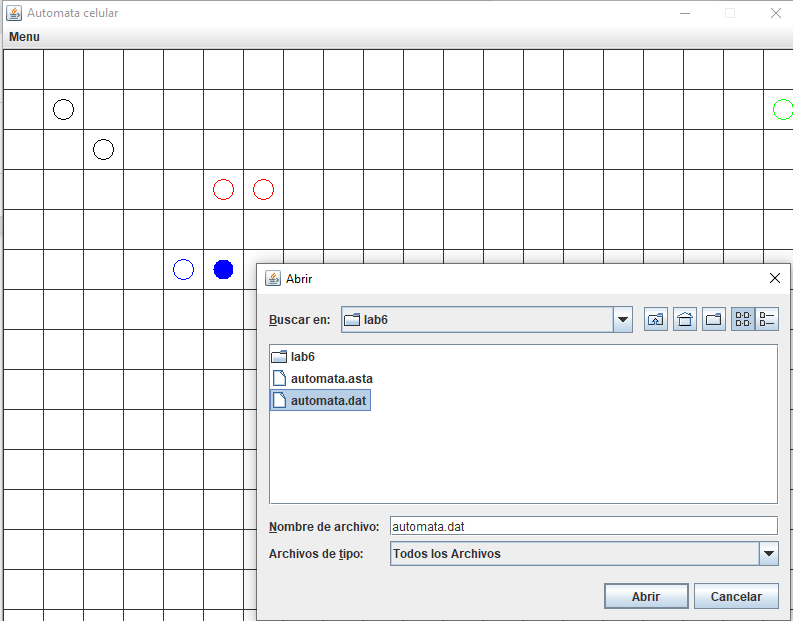


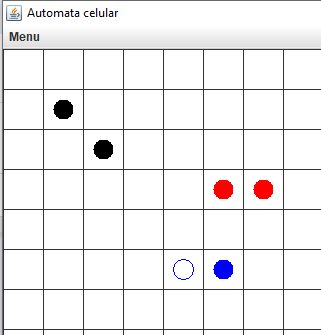
1. Construyan el método abra que ofrece el servicio de leer un automata de un archivo. Por ahora para las excepciones sólo consideren un mensaje de error general.



1. Realicen una prueba de aceptación para este método iniciando la aplicación, creando una nueva situación en el automata y abriendo el archivo unautomata.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 automata desde un archivo de texto y exportarlo. Los nombres de los archivos de texto deben tener como apellido .txt

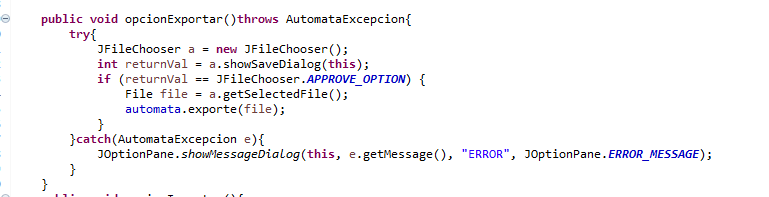
Los archivos texto tienen una línea de texto por cada elemento

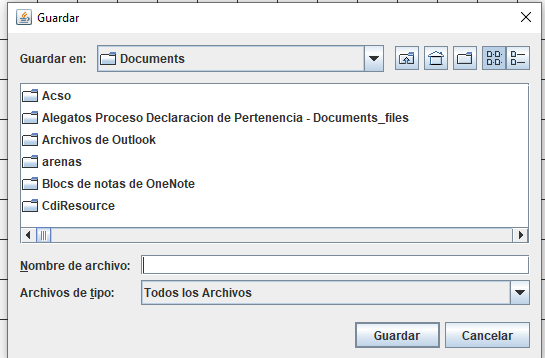
En cada línea asociada un elemento se especifica el tipo y la posición.

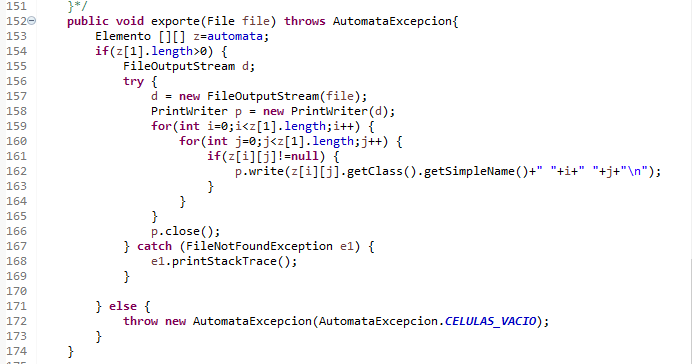
Cedula 20 20

Barrera 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.

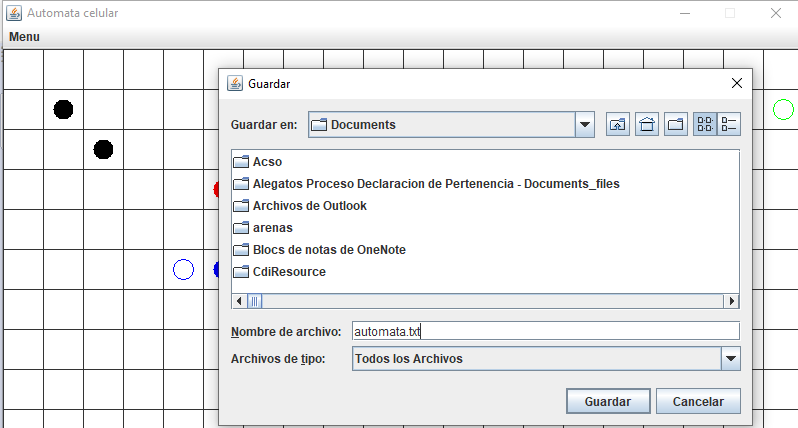


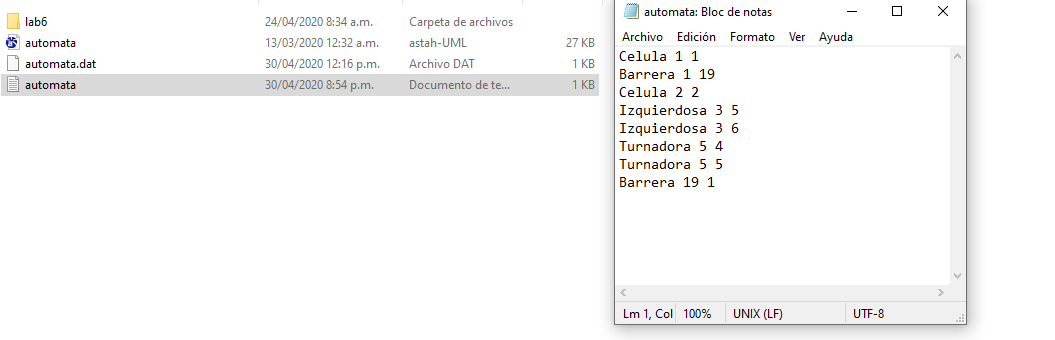


1. Construyan el método exporte que ofrece el servicio de exportar a un archivo texto, con el formato definido, el estado actual.
2. 
3. Realicen una prueba de aceptación de este método: iniciando la aplicación y exportando como unautomata.txt. Editen el archivo y analicen los resultados. ¿Qué pasó?

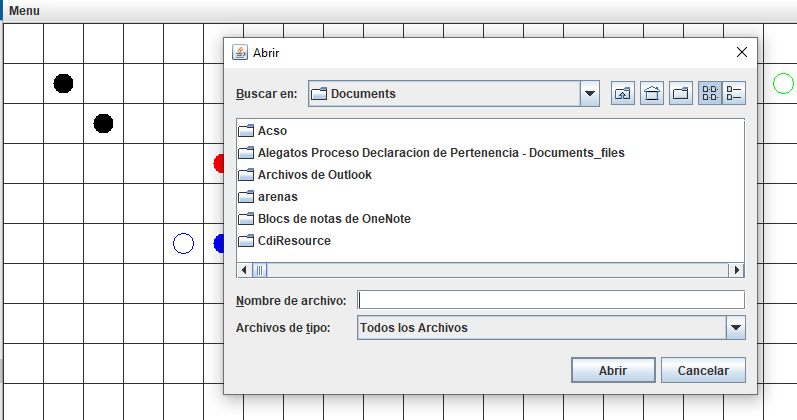
RTA//

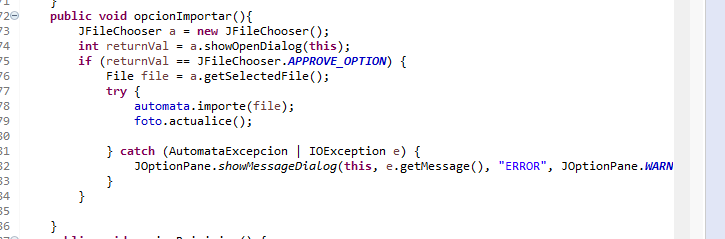
se guardo en el archivo txt toda la informacion del autómata y se creo un archivo el cual contiene las células y las posiciones donde estan en un string como se tenía dispuesto.





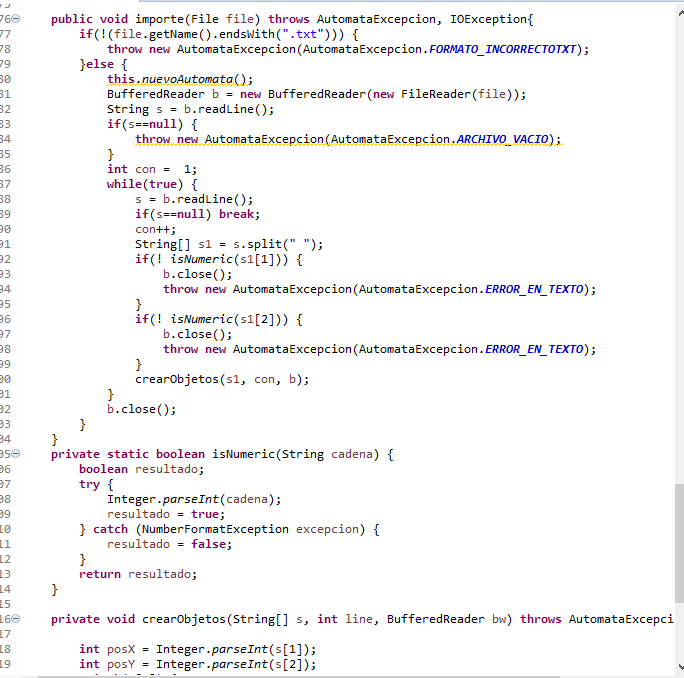
1. Construyan el método opcionImportar que une de forma adecuada la capa de presentación con la capa de aplicación. Ejecuten la aplicación y capturen una pantalla significativa.





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



## 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 ábralo. Describan el comportamiento

RTA//la aplicación queda tal cual como se había guardado la primera vez que se utilizó el guardar.

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

RTA// la aplicación queda con las células en la posición correspondiente , los pone como recién creados los archivos.

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

RTA// al abrir el archivo y al importar en el momento en el que se grafican, las celulas o los puntos son diferentes.

## Perfeccionando salvar y abrir

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

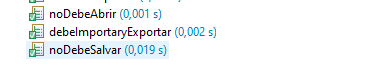
1. Copien las versiones actuales de abra y salve y renómbrenlos como abra01 y salve01

HECHO

1. Perfeccionen el manejo de excepciones de los métodos abra y salve detallando los errores.

HECHO

1. 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 , automataErr.txt \*.java] [NO OLVIDEN BDD y MDD]

1. Copien las versiones actuales de importe y exporte y renómbrenlos como importe01 y exporte01

HECHO

1. Perfeccionen el manejo de excepciones de los métodos importe y exporte detallando los errores.

HECHO

1. 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 , automataErr.txt \*.java] [NO OLVIDEN BDD y MDD]

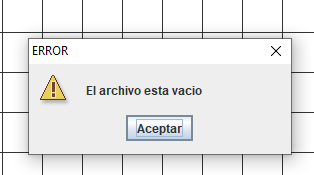
1. Copien las versiones actuales de importe y exporte y renómbrenlos como importe02 y exporte02

HECHO

1. 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.

HECHO

1. Escriban otro archivo con errores, llámelo automataErr.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 , automataFlex.txt \*.java] [NO OLVIDEN BDD y MDD]

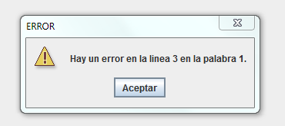
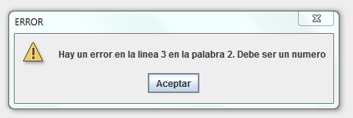
1. Copien las versiones actuales de importe y exporte y renómbrenlos como importe03 y exporte03

HECHO

1. Perfeccionen los métodos importe y exporte para que pueda servir para cualquier tipo de elementoscreados en el futuro

HECHO

1. Escriban otro archivo de pruebas, llámelo automataErrG.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)

RTA// 15 horas por persona

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

RTA// el laboratorio se encuentra completo

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

RTA// la de pruebas de aceptación ya que nos ayudaba a mirar si el laboratorio estaba bien, ademas tambien nos ayudo en la de pares porque así nos entendimos y lo hicimos completo.

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

RTA// hacer el importe y exporte ya que como planteamos el lab 3 se nos complicaba hacer que guardar el .txt en la matriz y no en arreglos como habria sido mas facil.

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

RTA// acordarnos de como funcionaba el laboratorio autómata y verificar que todo funcionara bien para poder implementar las nuevas cosas que debíamos hacer.

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

RTA// se hizo la práctica a par bien y logramos desarrollar el laboratorio completo.