Escuela Colombiana de Ingeniería Julio Garavito

Andrés Cardozo

Tulio Riaño

Laboratorio 05

Programacion Orientada a Objetos

FECHA:

26/04/2025

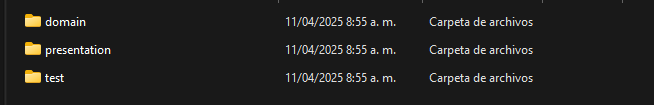
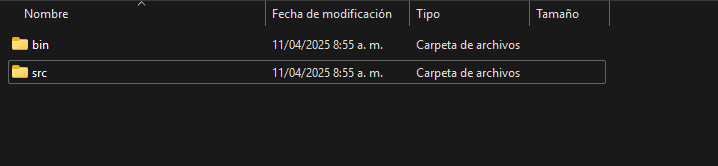
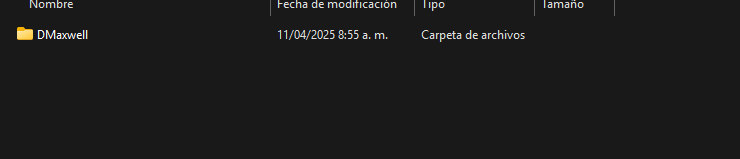
# **DESARROLLO**

Directorios

El objetivo de este punto es construir un primer esquema para el juego DMaxwell

1. Preparen un directorio llamado DMaxwell con los directorios src y bin y los subdirectorios para presentación, dominio y pruebas de unidad. Capturen un pantalla con la estructura.

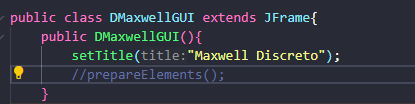
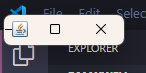
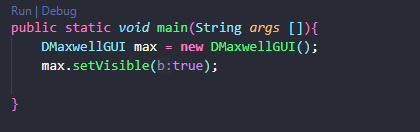
mkdir “nombre”



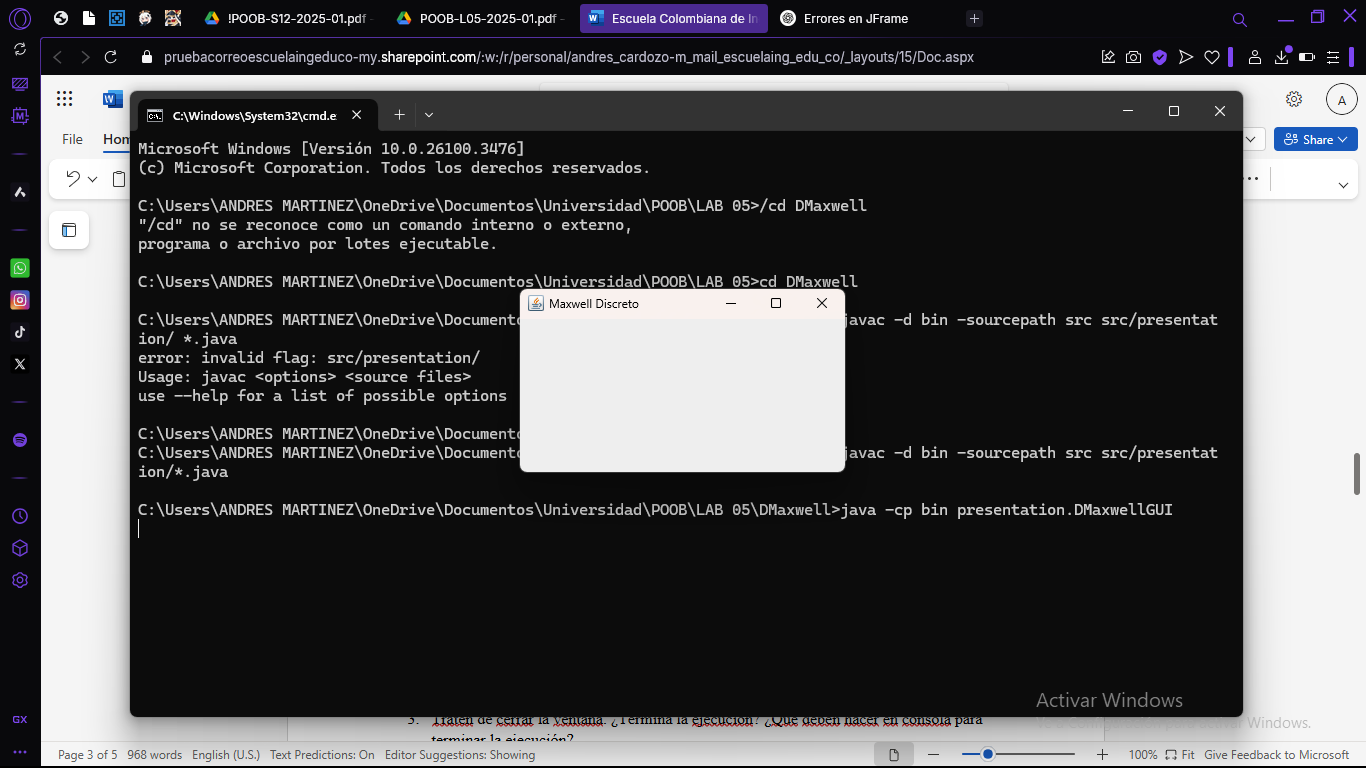
# **Ciclo 0: Ventana vacía – Salir**

El objetivo es implementar la ventana principal de DMaxwell con un final adecuado desde el icono de cerrar. Utilizar el esquema de prepareElements-prepareActions.

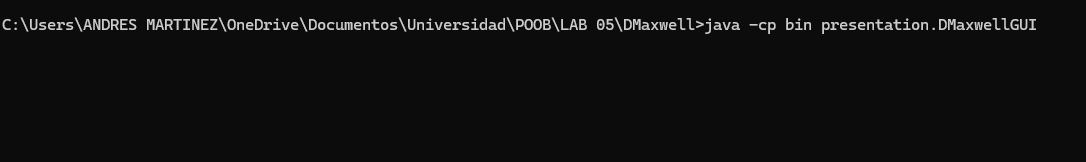
1. Construyan el primer esquema de la ventana de DMaxwell únicamente con el título “Maxwell Discreto”. Para esto cree la clase DMaxwellGUI como un JFrame con su creador (que sólo coloca el título) y el método main que crea un objeto DMaxwellGUI y lo hace visible. Ejecútenlo. Capturen la pantalla. (Si la ventana principal no es la inicial en su diseño, después deberán mover el main al componente visual correspondiente)

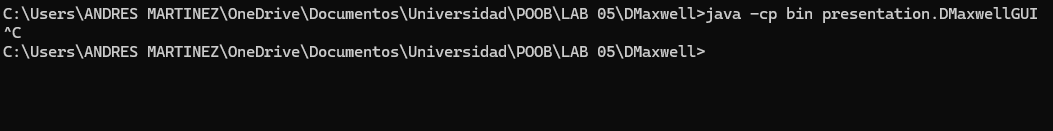
1. Modifiquen el tamaño de la ventana para que ocupe un cuarto de la pantalla y ubíquenla en el centro. Para eso inicien la codificación del método prepareElements. Capturen esa pantalla.



1. Traten de cerrar la ventana. ¿Termina la ejecución? ¿Qué deben hacer en consola para terminar la ejecución?



No termina la ejecucion ya que estamos trabajando con multihilos: y el hilo de eventos no se cierra cuando cerramos la ventana. Por lo que debemos poner control c para que se termine la ejecución en la consola.



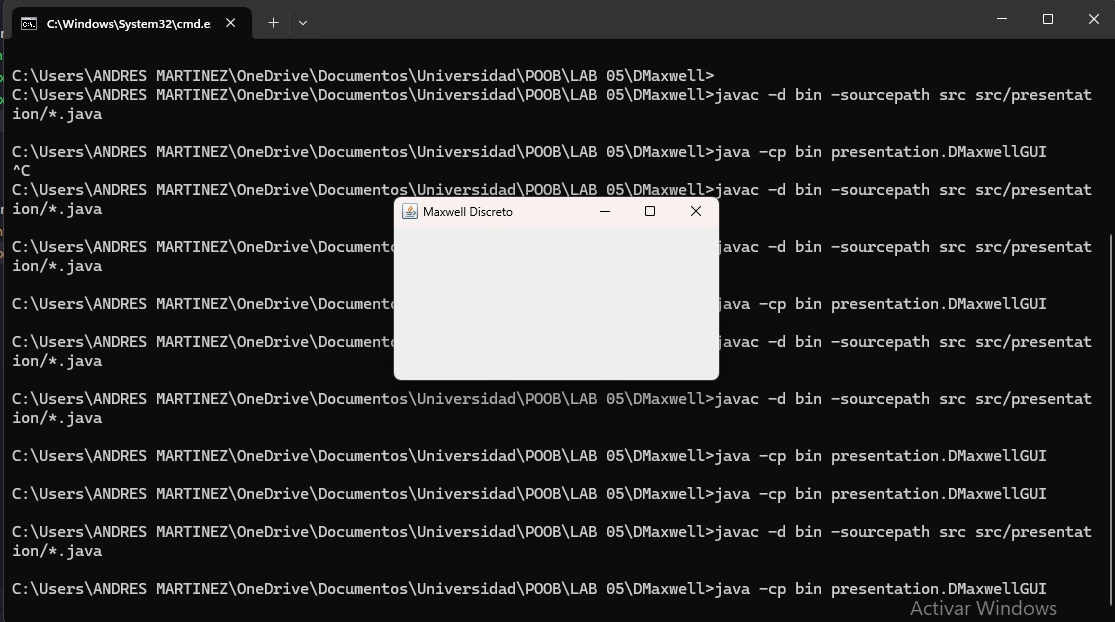
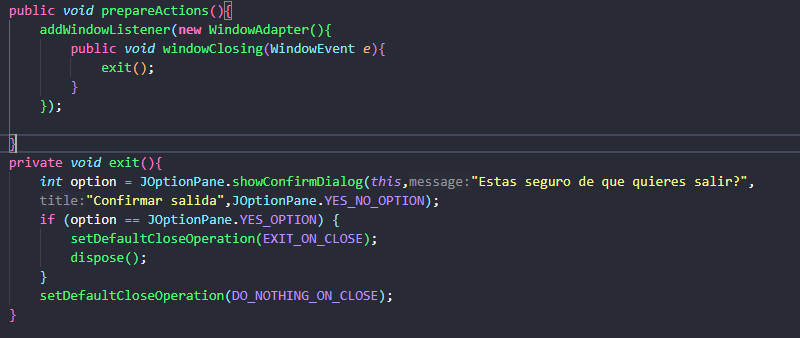
1. Estudien en JFrame el método setDefaultCloseOperation. ¿Para qué sirve? ¿Cómo lo usarían si queremos confirmar el cierre de la aplicación? ¿Cómo lo usarían si queremos simplemente cerrar la aplicación?

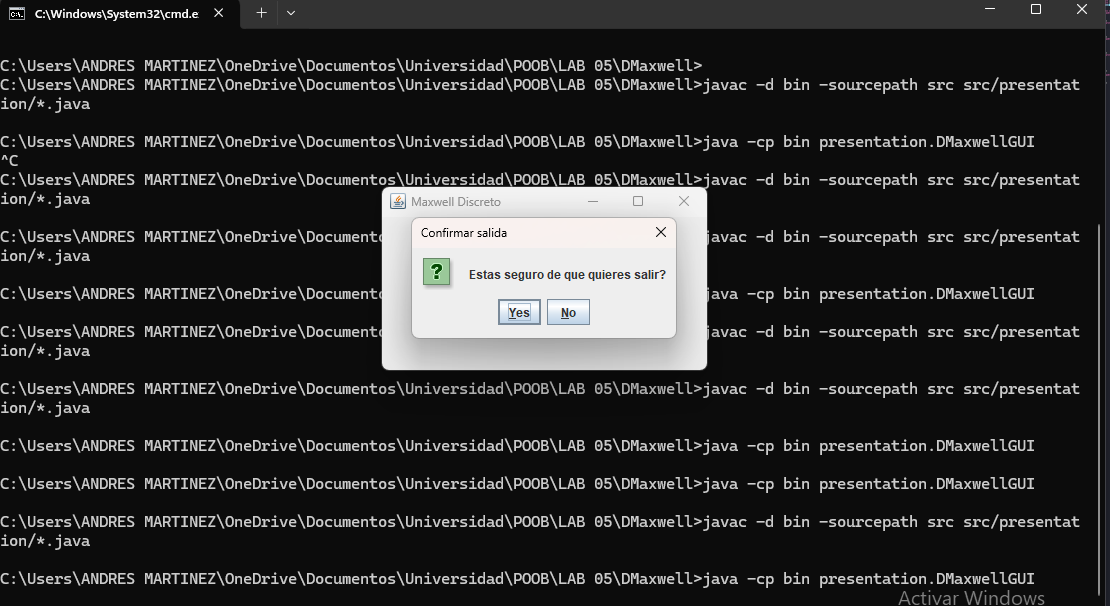
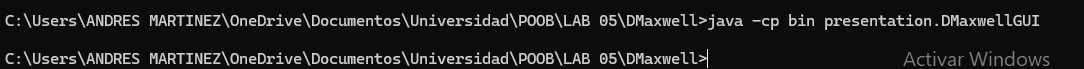
“Unlike a Frame, a JFrame has some notion of how to respond when the user attempts to close the window. The default behavior is to simply hide the JFrame when the user closes the window. To change the default behavior, you invoke the method setDefaultCloseOperation(int). To make the JFrame behave the same as a Frame instance, use setDefaultCloseOperation(WindowConstants.DO\_NOTHING\_ON\_CLOSE).”

Termina la aplicacion cuando se cierra la ventana, por lo que debemos definir qué hacer cuando se cierre la ventana (por default esta HIDE\_ON\_CLOSE ):  
Inserting image...  
Asi que para cerrar la aplicacion cuando se cierre la vertana ponermos EXIT\_ON\_CLOSE.

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE)

1. Preparen el “oyente” correspondiente al icono cerrar que le pida al usuario que confirme su selección. Para eso inicien la codificación del método prepareActions y el método asociado a la acción (exit). Ejecuten el programa y cierren el programa. Capturen las pantallas.



Y cuando le damos Yes se acaba:

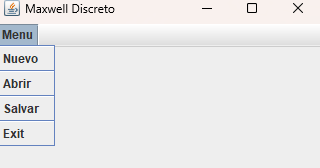
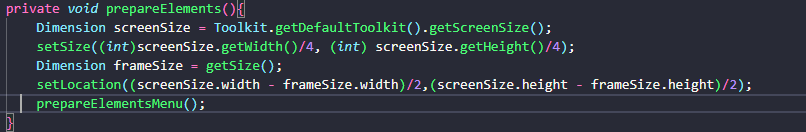
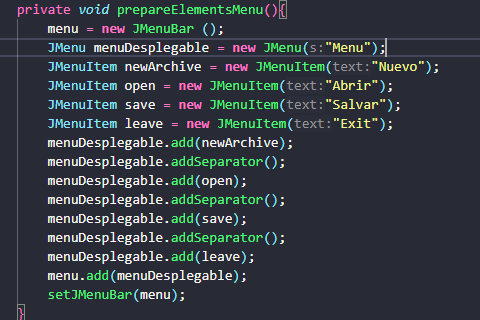
# **Ciclo 1: Ventana con menú – Salir**

El objetivo es implementar un menú clásico para la aplicación con un final adecuado desde la opción del menú para salir. El menú debe ofrecer mínimo las siguientes opciones: Nuevo, Abrir – Salvar y Salir . Incluyan los separadores de opciones.

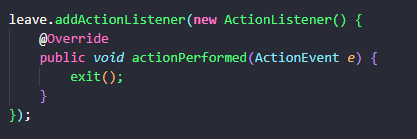
1. Expliquen los componentes visuales necesarios para este menú. ¿Cuáles serían atributos y cuáles podrían ser varibles del método prepareElements? Justifique.

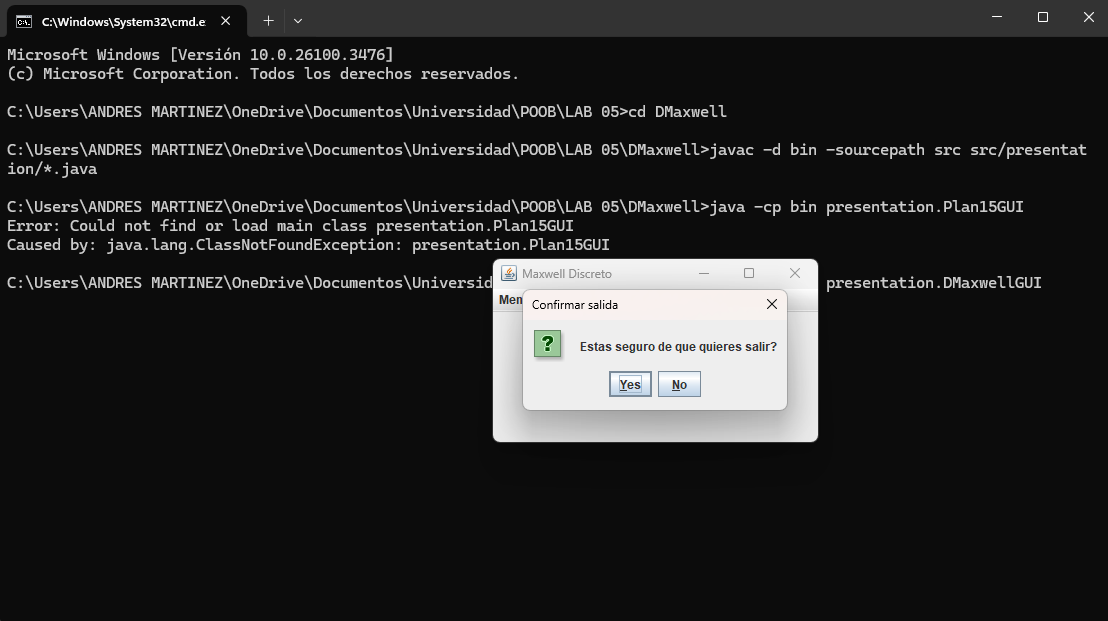
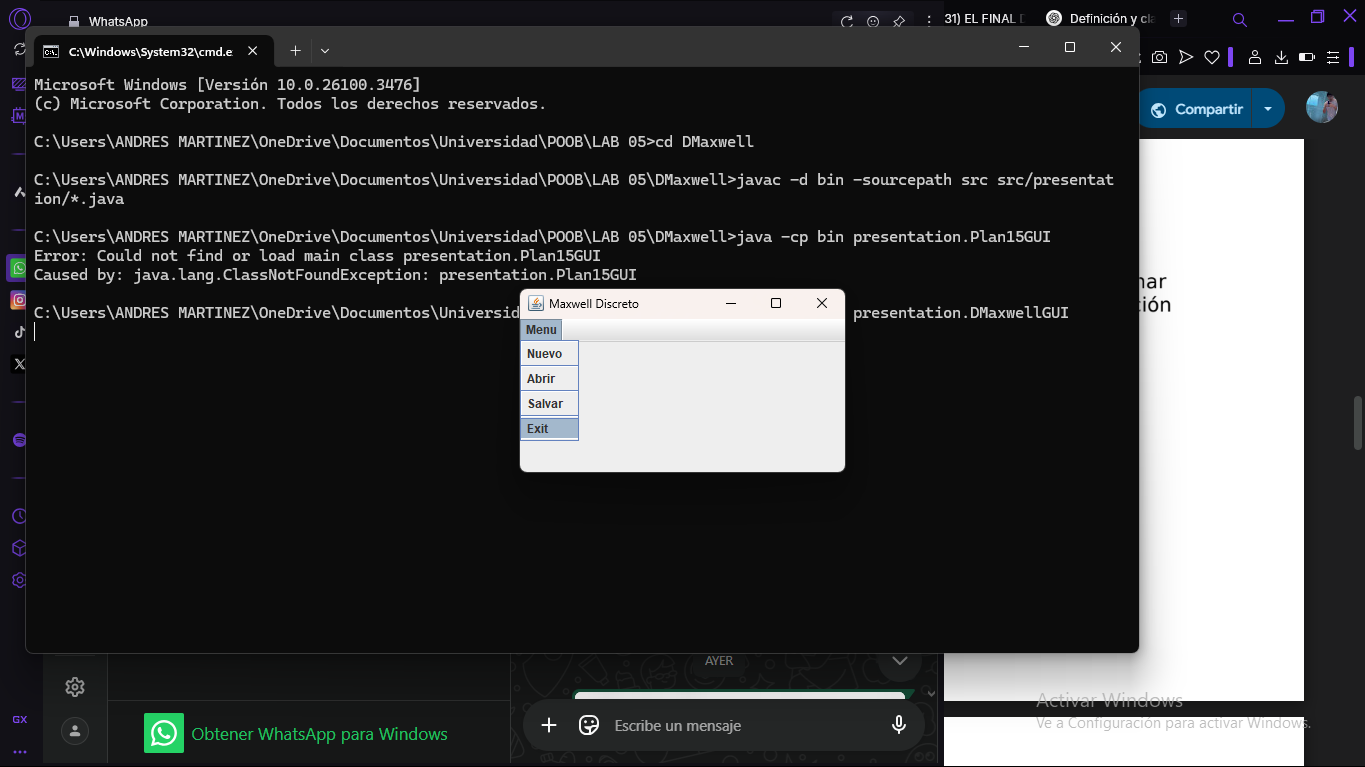
Los componenetes visuales necesarios son: JMenuBar JMenu JMenuItem JMenuSeparator.  
Definimos como atributo el menuBar, y si queremos ponerle la logica de acciones a los items, lo ponemos de atributo. Y las variables serias JMenu y sus respectivos JMenuItems.

1. Construya la forma del menú propuesto (prepareElements - prepareElementsMenu) . Ejecuten. Capturen la pantalla.



1. Preparen el “oyente” correspondiente al icono cerrar con confirmación (prepareActions - prepareActionsMenu). Ejecuten el programa y salgan del programa. Capturen las pantallas.





# **Ciclo 2: Salvar y abrir**

El objetivo es preparar la interfaz para las funciones de persistencia

1. Detalle el componente JFileChooser especialmente los métodos: JFileChooser, showOpenDialog, showSaveDialog, getSelectedFile.

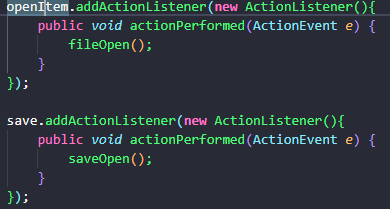
JFileChooser es un componente pre-implementado que permite seleccionar un archivo. Así este devuelve una instancia del selector de archivos.

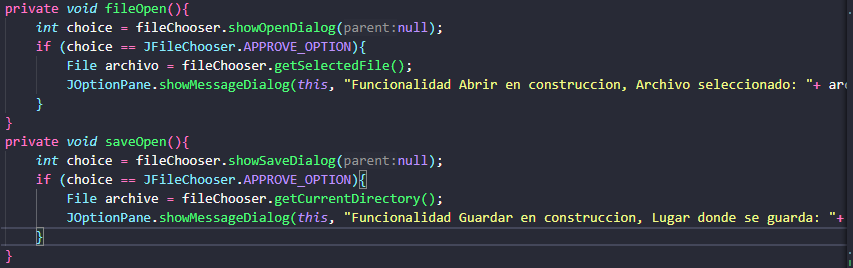
ShowOpenDialog muestra una ventana para abrir archivos

ShowSaveDialog muestra una ventana para guardar archivos

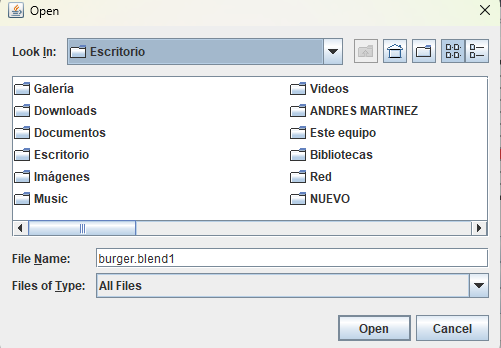
GetSelectedFile devuelve el archivo que el usuario selecciono

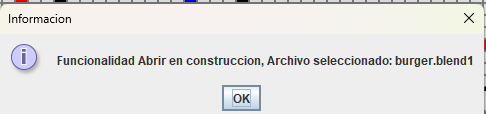
1. Implementen parcialmente los elementos necesarios para salvar y abrir. Al seleccionar los archivos indique que las funcionalidades están en construcción detallando la acción y el nombre del archivo seleccionado.

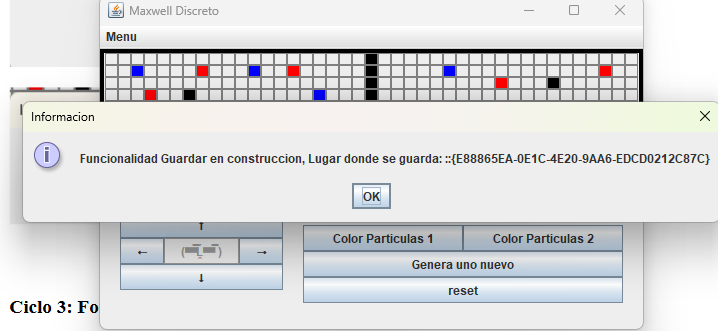




1. Ejecuten las dos opciones y capturen las pantallas más significativas.



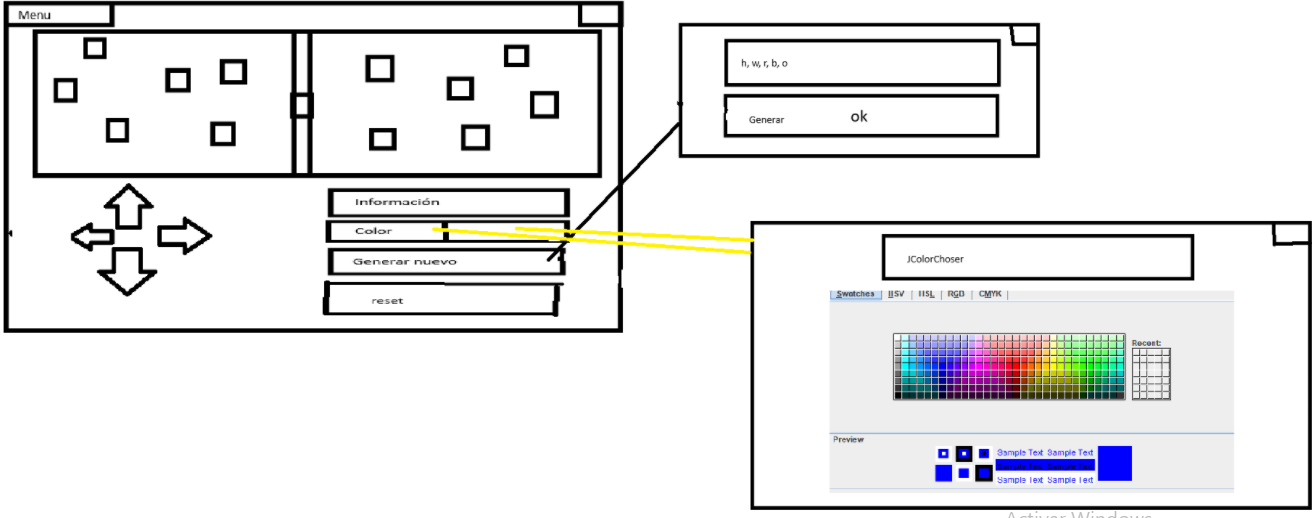




# **Ciclo 3: Forma de la ventana principal**

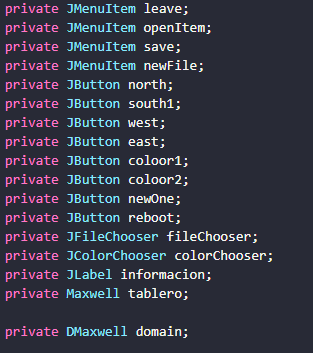
El objetivo es codificar el diseño de la ventana principal (todos los elementos de primer nivel)

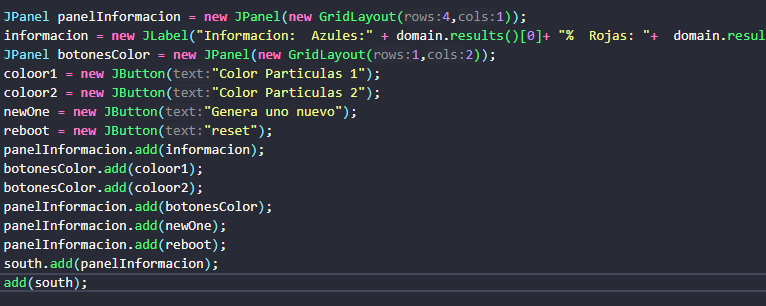
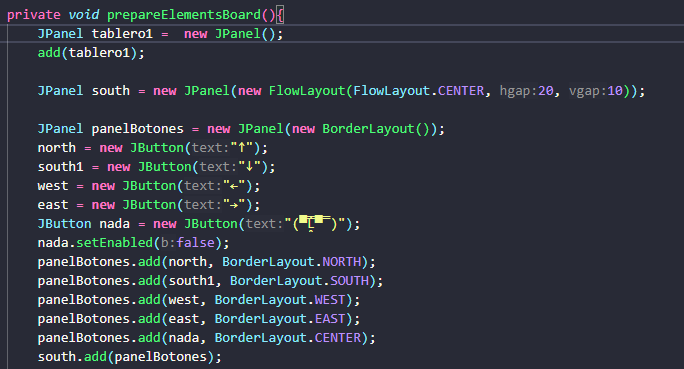
1. Presenten el bosquejo del diseño de interfaz con todos los componentes necesarios. (Incluyan la imagen)

Vamos a necesitar JButton, JLabel, fileChooser, colorChooser, ademas de los elementos para el menu. Para generar el nuevo tablero, usamos JTextField para la toma de datos.

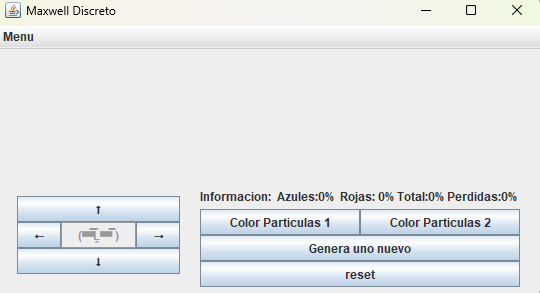
1. Continúen con la implementación definiendo los atributos necesarios y extendiendo el método prepareElements(). Para la zona del tablero definan un método prepareElementsBoard() y un método refresh() que actualiza la vista del tablero considerando, por ahora, el tablero inicial por omisión. Este método lo vamos a implementar realmente en otros ciclos.

Definimos como atributos los componentes que necesitaremos mas adelante como los que tienen un listener





1. Ejecuten y capturen la pantalla.



# **Ciclo 4: Cambiar colores**

El objetivo es implementar este caso de uso.

1. Expliquen los elementos (vista – controlador) necesarios para implementar este caso de uso.

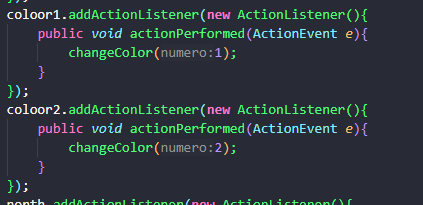
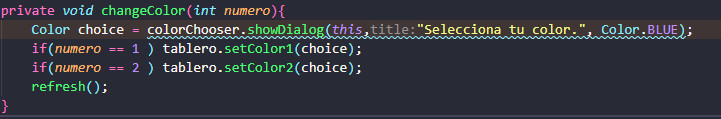
Necesitamos un componente colorChooser en el cual se va a mostrar la ventana en donde el usuario puede escoger los colores. Asi mismo crear listeners asociados a los botones de cambio de color.

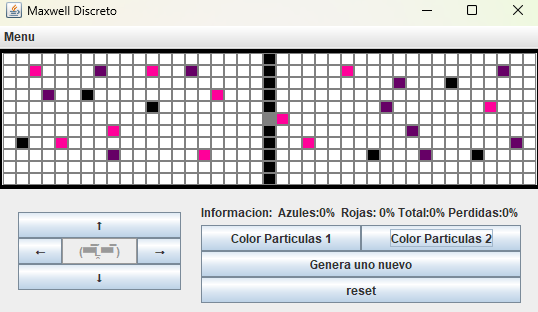
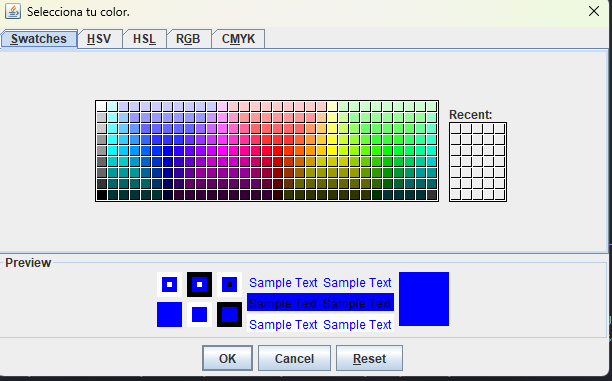
1. Detalle el comportamiento de JColorChooser especialmente el método estático showDialog

JColorChooser es una herramienta gráfica que ofrece Java para permitir que el usuario seleccione un color visualmente. showDialog es un método estático que muestra una ventana emergente (modal) con el selector de colores. Este método espera a que el usuario elija un color o cierre la ventana, y devuelve el color seleccionado, si hubo alguno. Por lo que debemos manejar esta confirmación.

1. Implementen los componentes necesarios para cambiar el color de las fichas.

Asi, dependiendo el boton que representa a que particulas cambiarles el color, va a tener uno predeterminado:



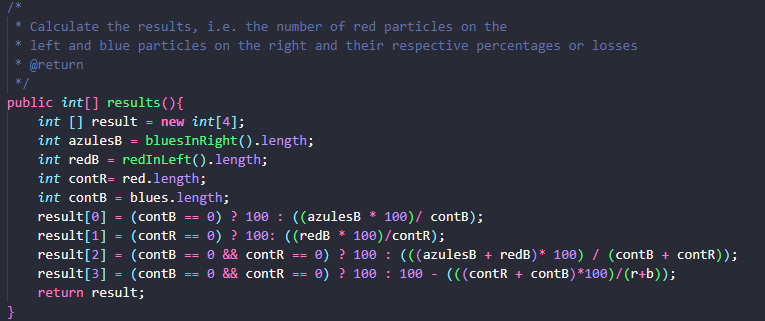
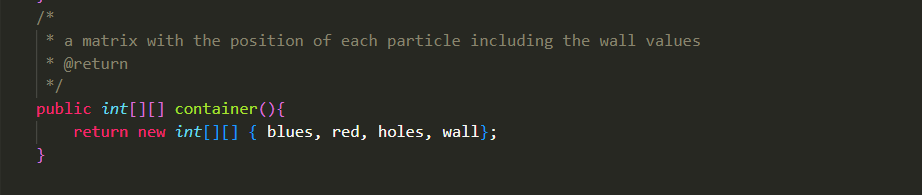
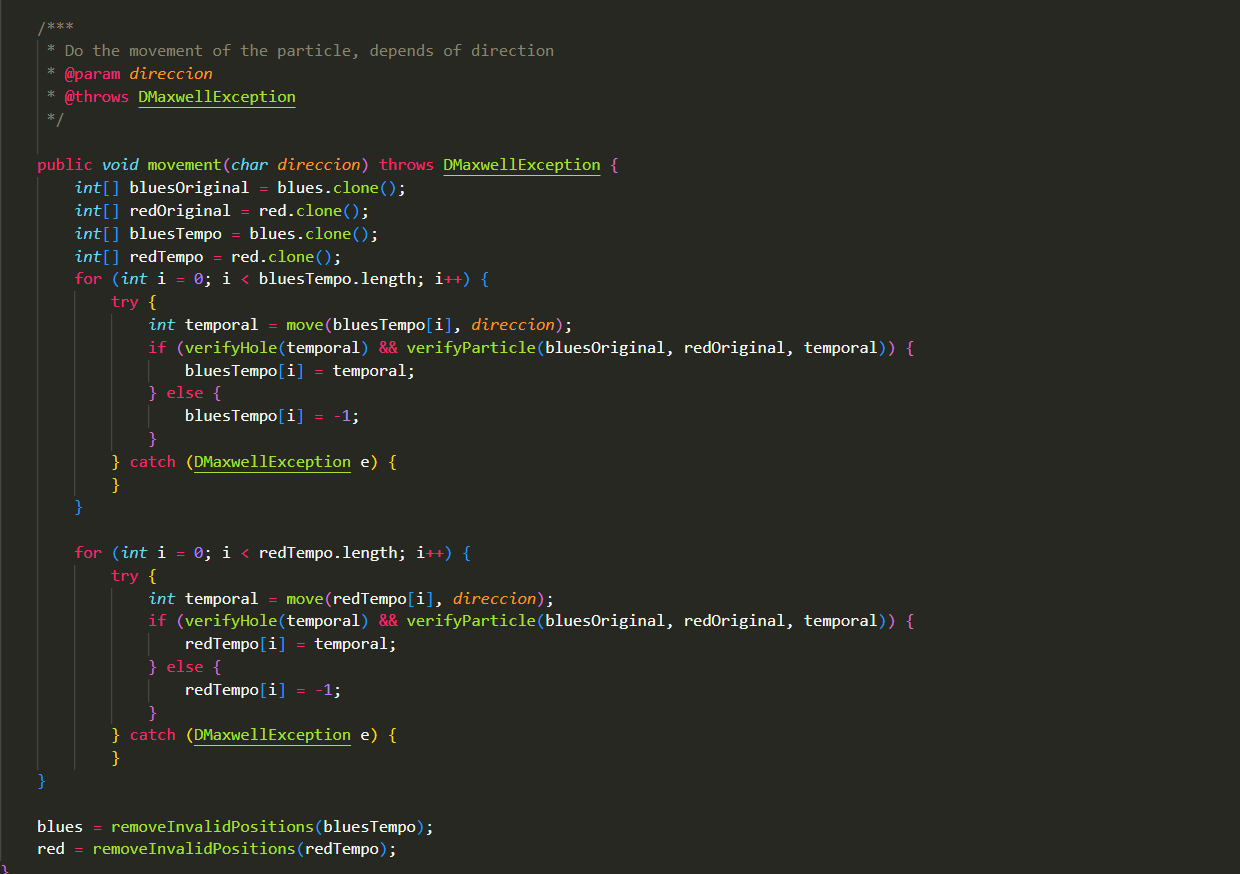
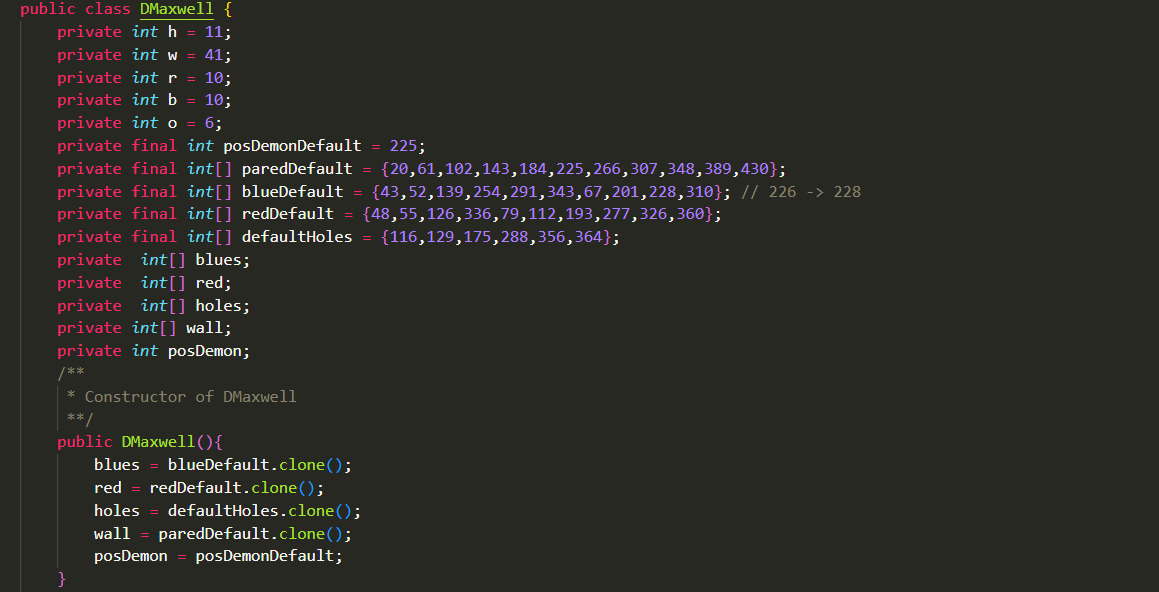
1. Ejecuten el caso de uso y capture las pantallas más significativas.

# **Ciclo 5: Modelo DMaxwell**

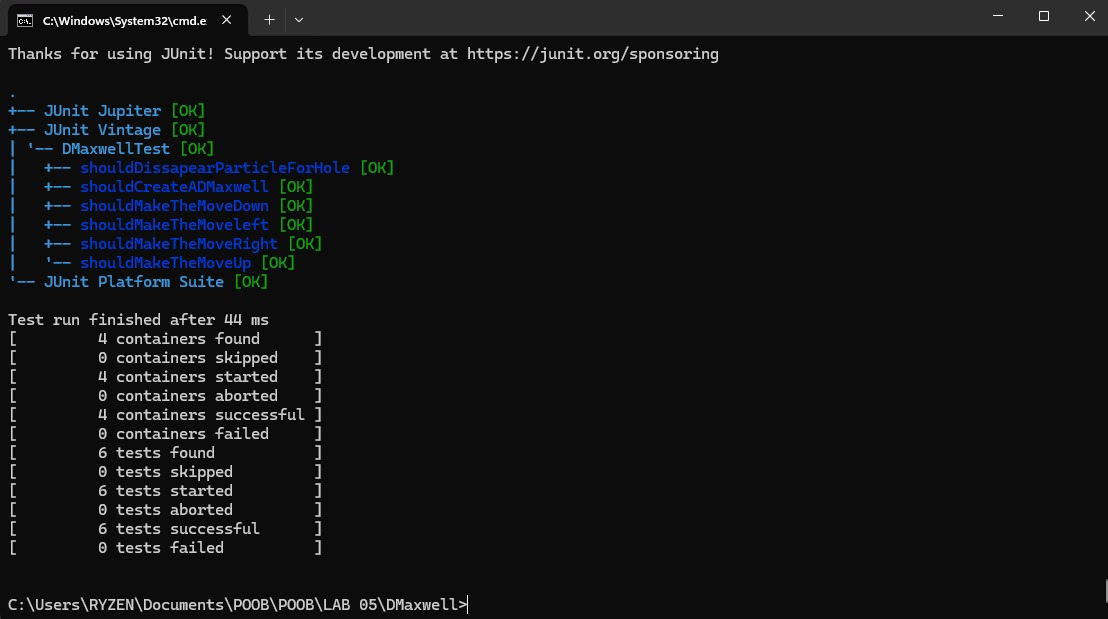
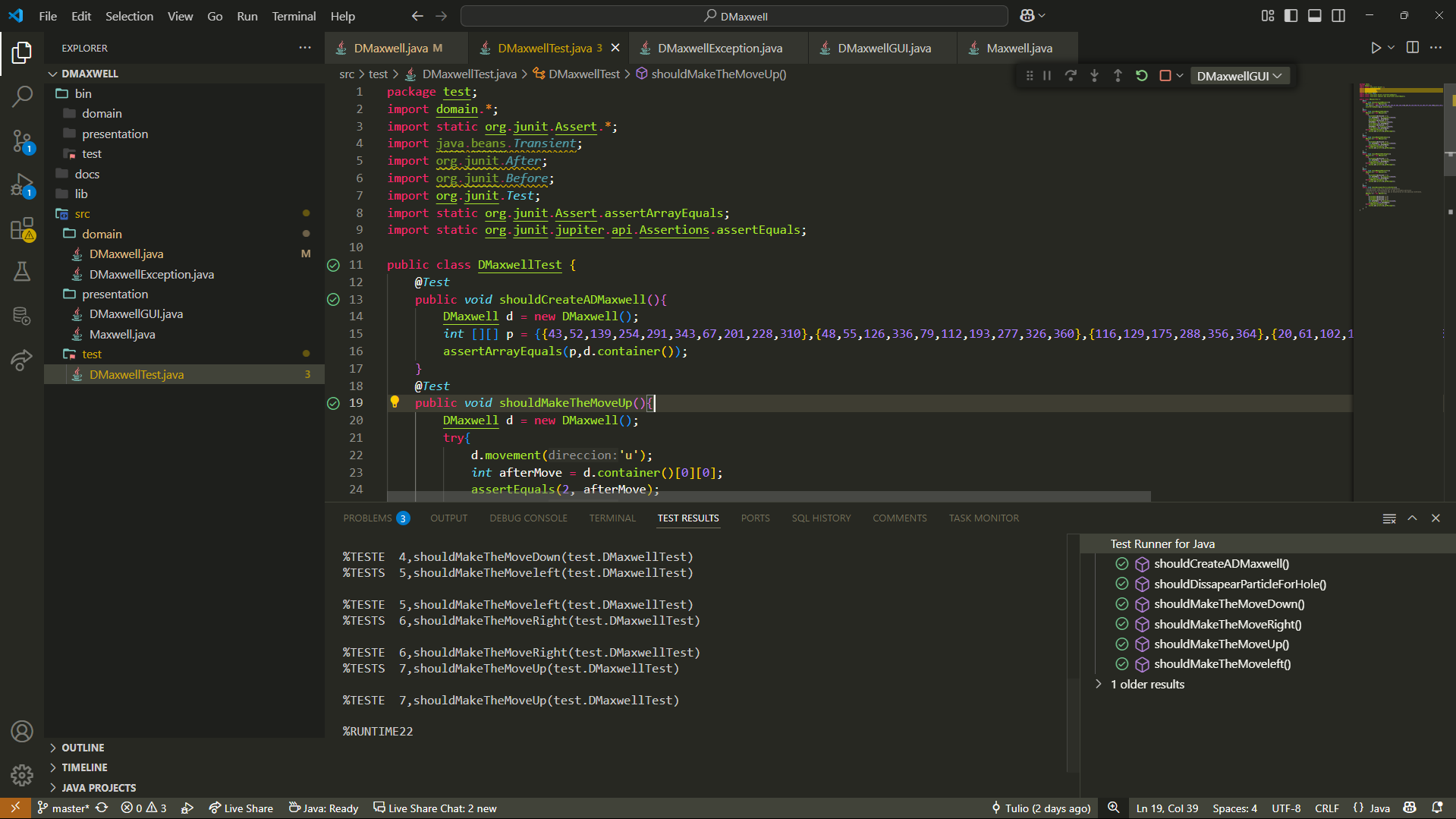
El objetivo es implementar la capa de dominio para DMaxwell

1. Construya los métodos básicos del juego (No olvide MDD y BDD)

Los metodos basicos que definimos fueron los constructores, move, container y results:

Teniamos unas dudas ya que no segun nosotros si una particula no se movia las demas tampoco, asi que lo hicimos asi. Pero despues de preguntar si una no se puede mover se queda ahi y las demas si se mueven. En el primer caso le mostrabamos un mensaje al usuario, ahora como está definido si deja moverlas.

1. Ejecuten las pruebas y capturen el resultado.



# **Ciclo 6: Jugar**

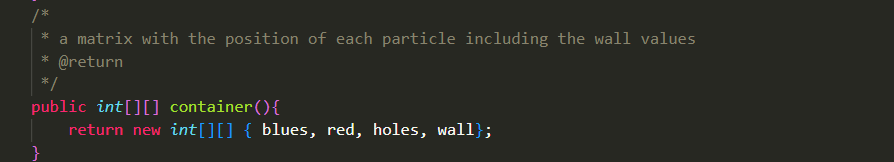
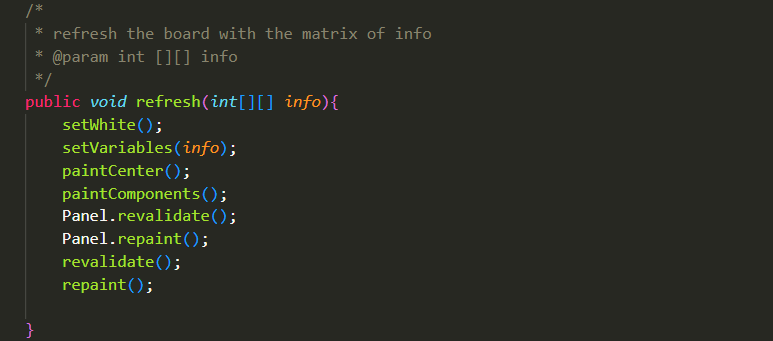
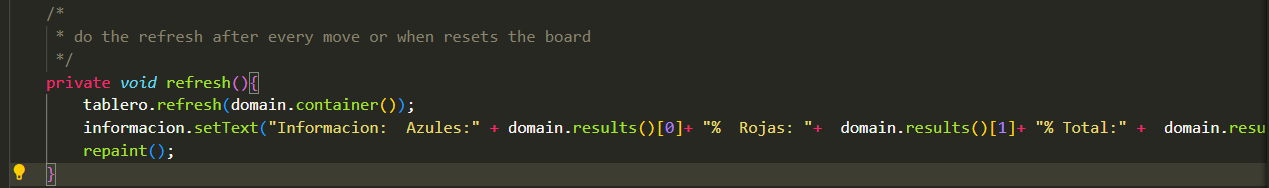
El objetivo es implementar el caso de uso jugar. (En Astah Se encuentran los casos de Uso Definidos).

1. Adicione a la capa de presentación el atributo correspondiente al modelo.



1. Perfeccionen el método refresh() considerando la información del modelo de dominio.

Después de cada movimiento se va a llamar el método refresh, con la información acorde que se encuentra en dominio, en tablero se va a volver a repintar todo el tablero tras el movimiento seleccionado por el usuario.



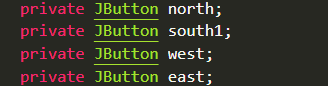
1. Expliquen los elementos necesarios para implementar este caso de uso.

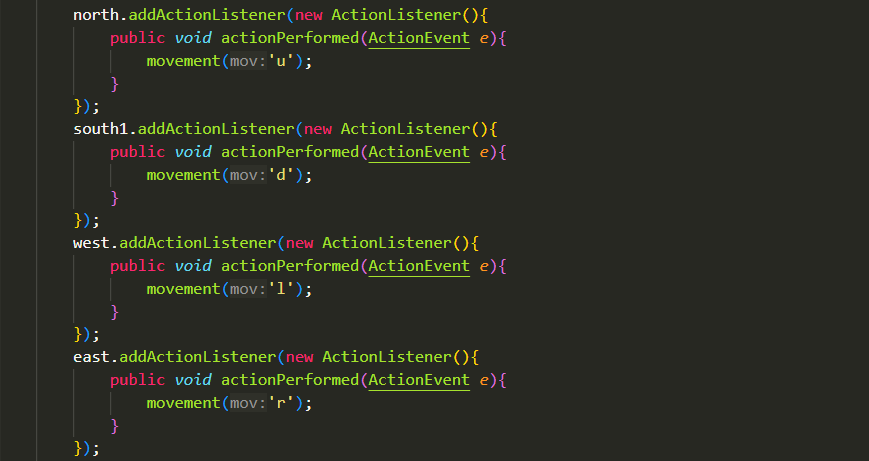
Los elementos necesarios para implementar este caso de uso fue en parte los ciclos que hicimos antes: Principalemente fue el dominio, ya que lo perfeccionamos para que trabajara correctamente y añadimos. Asi mismo los botones correspondientes para que interactuen con el usuario y su respectiva logica. Añadiendo nuevas funcionalidades y listeners para lo nuevo.

1. Implementen los componentes necesarios para jugar.¿Cuántos oyentes necesitan? ¿Por qué?

La cantidad de oyentes que definimos fueron cuatro, para hacer el movimiento correspondiente,

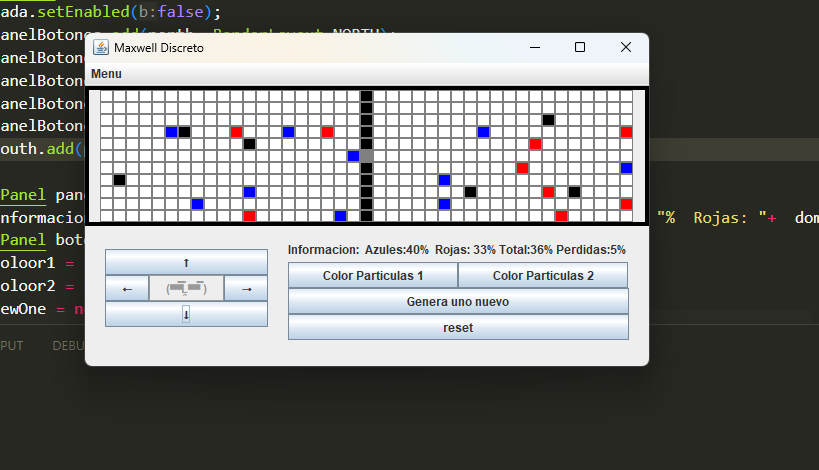
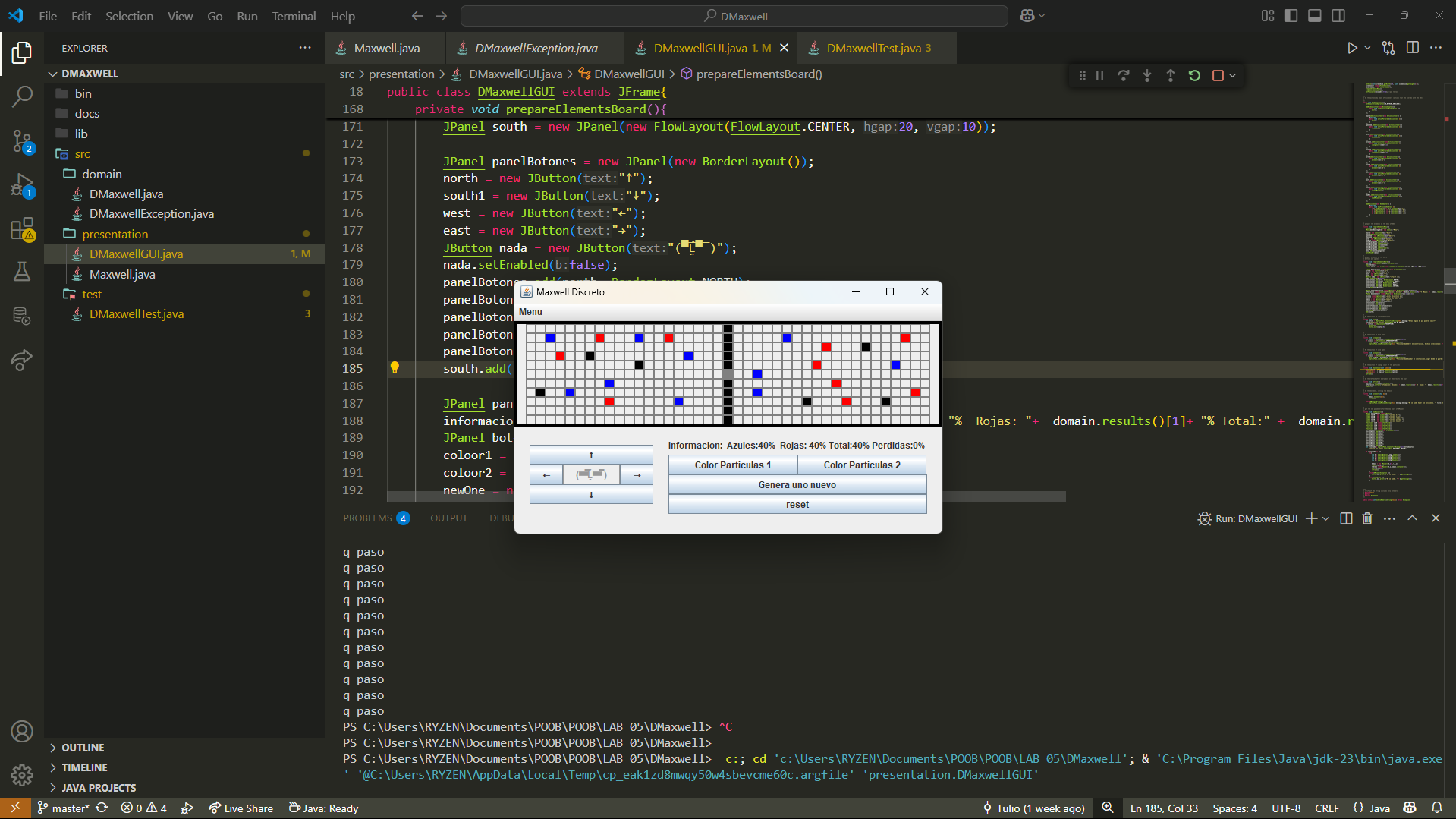
Abajo, arriba, derecha e izquierda:





1. Ejecuten el caso de uso y capture las pantallas más significativas.

Inicialmente tenemos el tablero:

Al hacer diferentes movimientos podemos observar que se han desplazado las particulas de manera correcta.

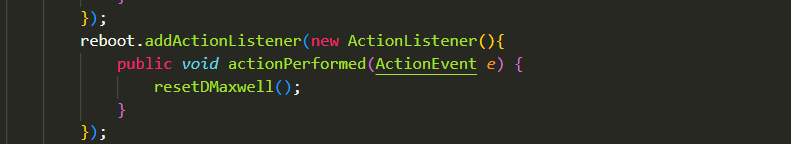
# **Ciclo 7: Reiniciar**

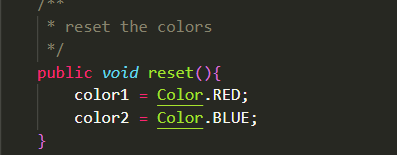
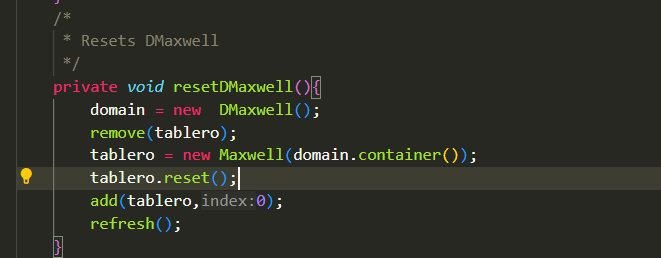
El objetivo es implementar este caso de uso. (En Astah Se encuentran los casos de Uso Definidos).

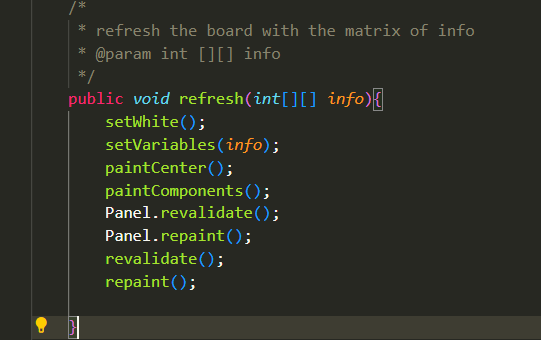
1. Expliquen los elementos a usar para implementar este caso de uso.

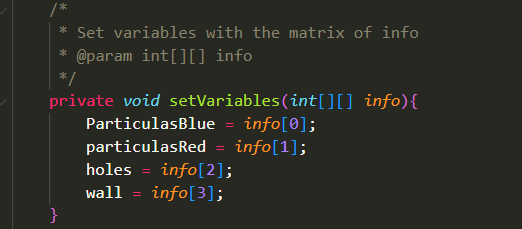
Los elementos que implementamos para desarrollar este caso de uso fueron un botón que, cada vez que se hiciera clic, reiniciaría el estado del tablero. En la capa de presentación tenemos un método llamado resetDMaxwell, en el cual se crea una nueva instancia de DMaxwell. Luego, removemos el tablero anterior y agregamos el nuevo al JPanel. Después, llamamos al método refresh, el cual limpia todo el panel (dejándolo en blanco), restablece los valores por defecto, pinta el centro y los componentes. Utilizamos revalidate(), que permite actualizar automáticamente el diseño del layout, y finalmente llamamos al método repaint().

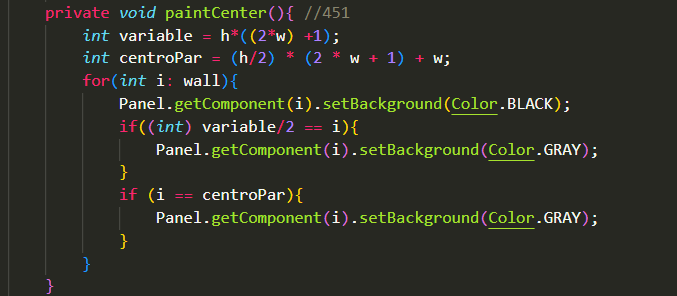
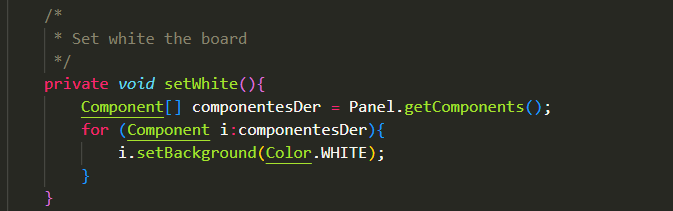
1. Implementen los elementos necesarios para reiniciar





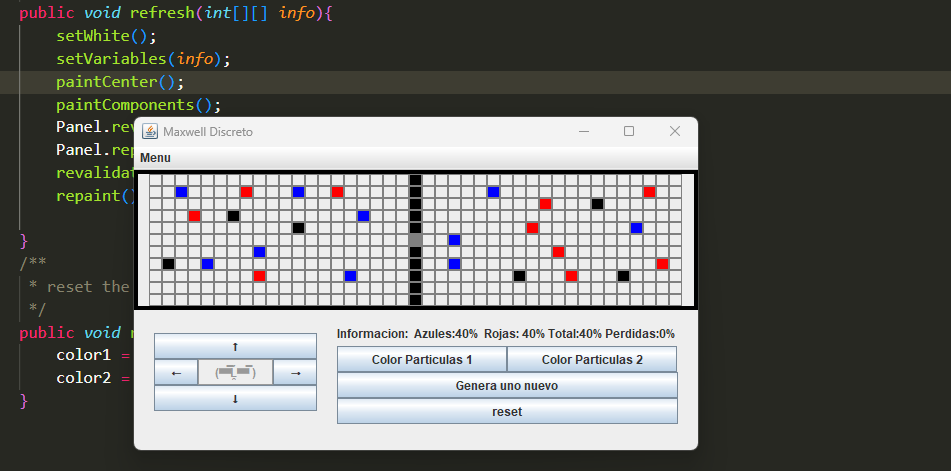


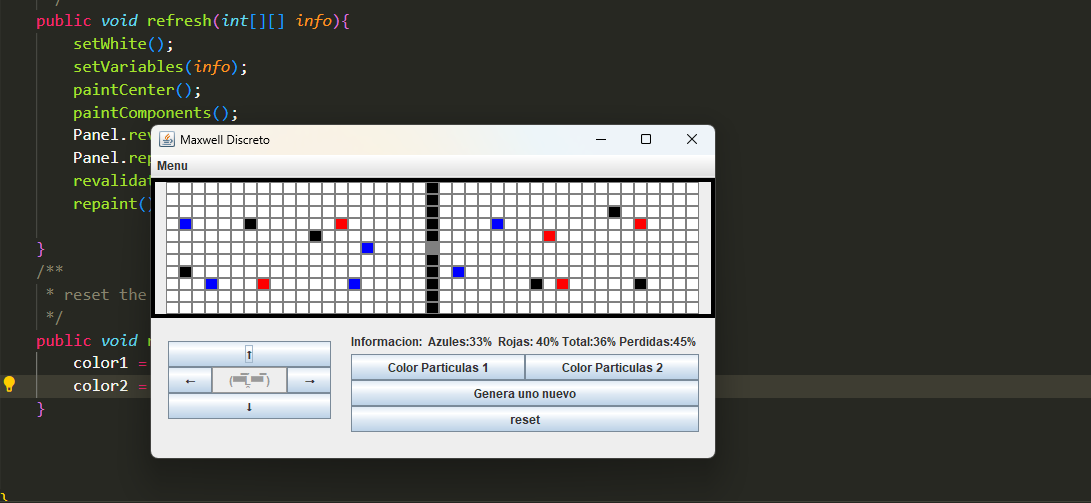


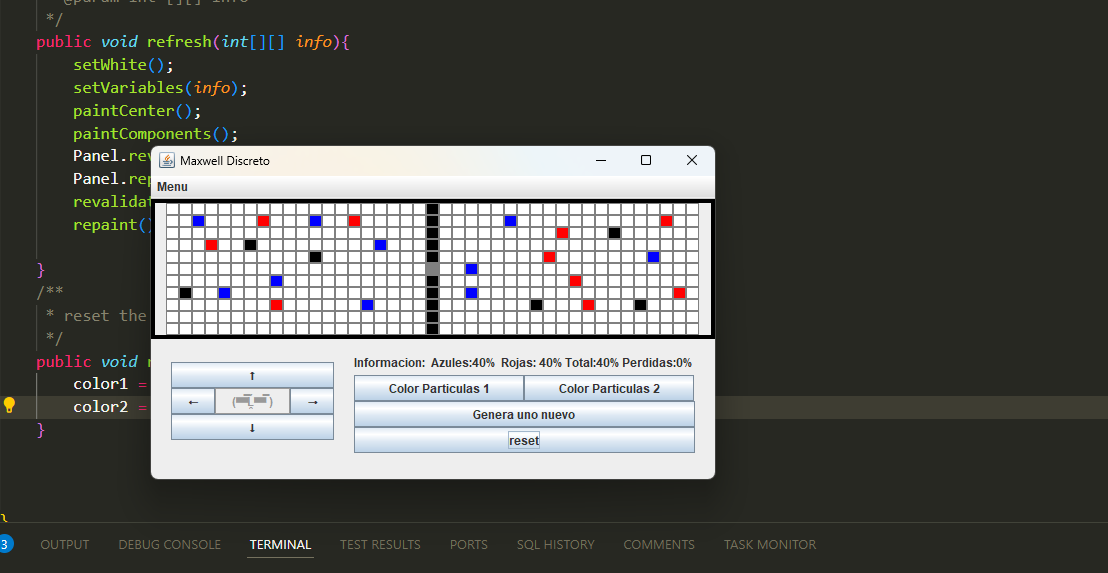


1. Ejecuten el caso de uso y capture las pantallas más significativas.

Inicialmente tenemos el tablero

Después de realizar ciertos movimientos:

Y al utilizar el botón reset, volvemos al estado inicial



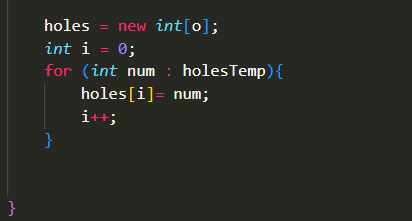
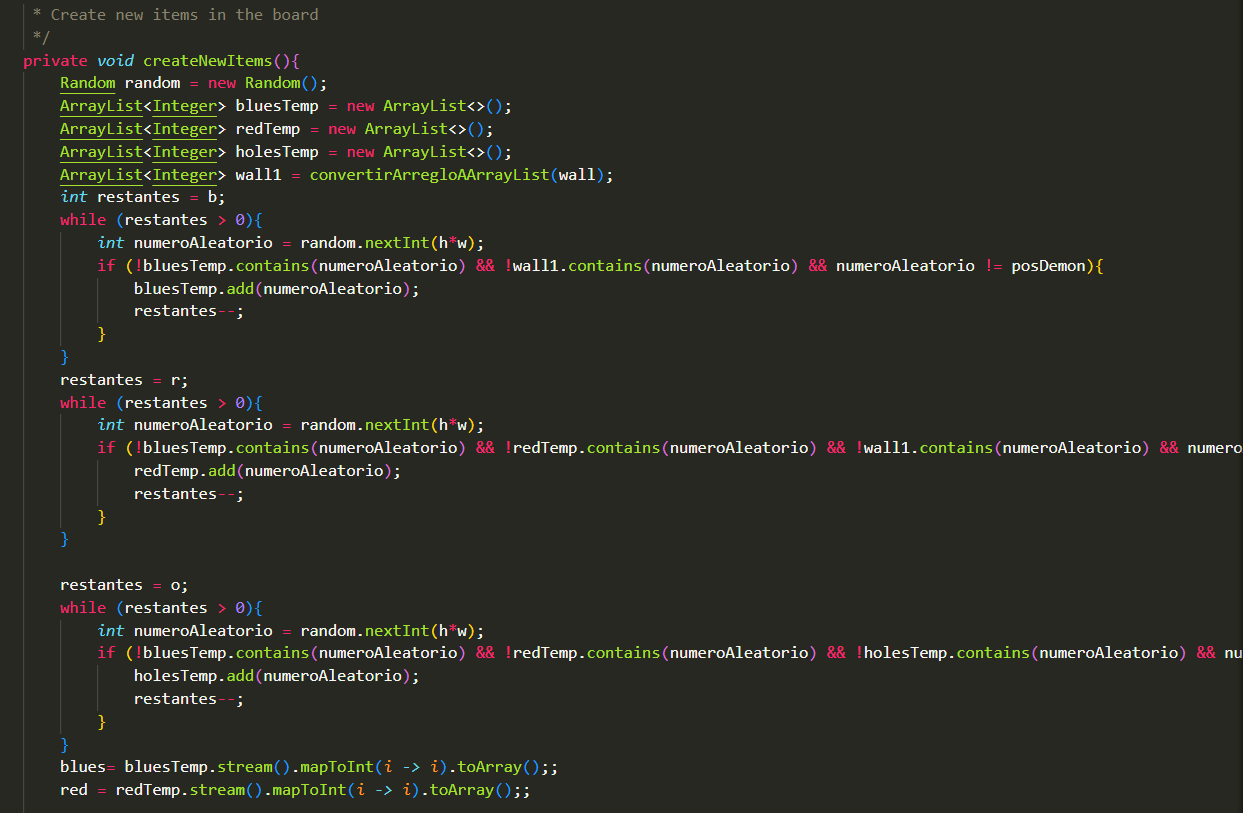
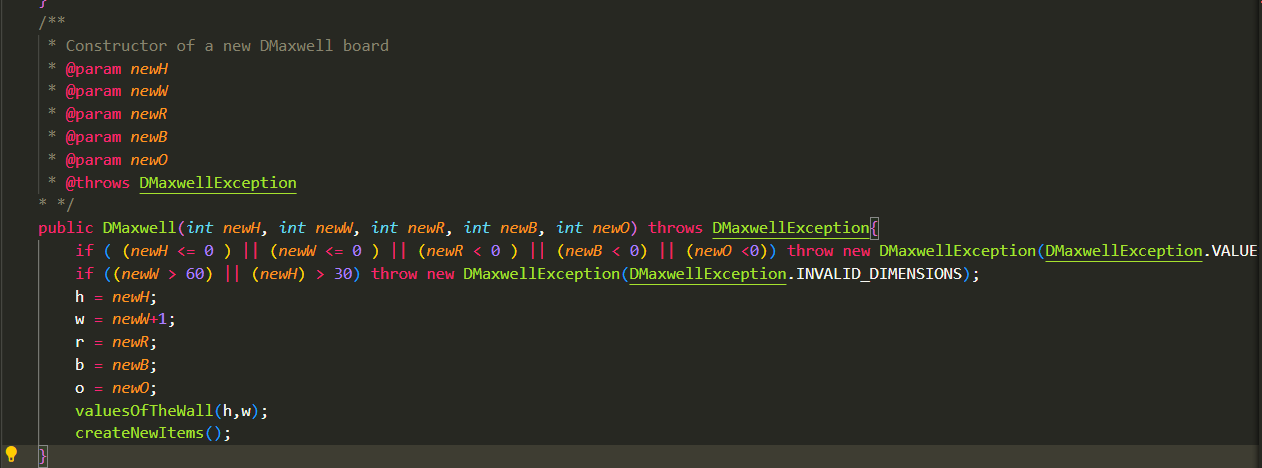
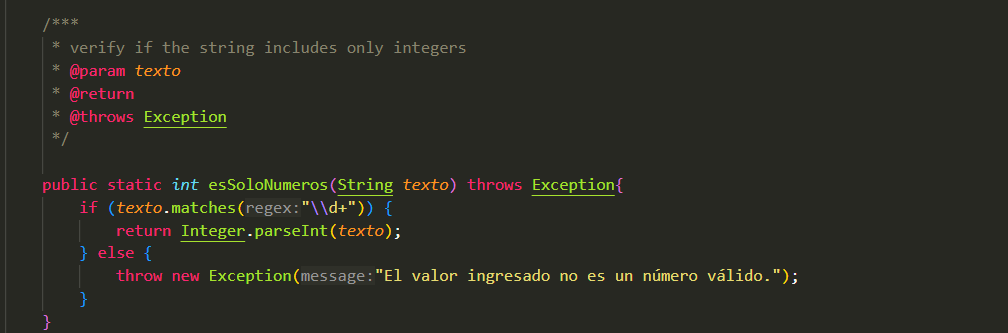
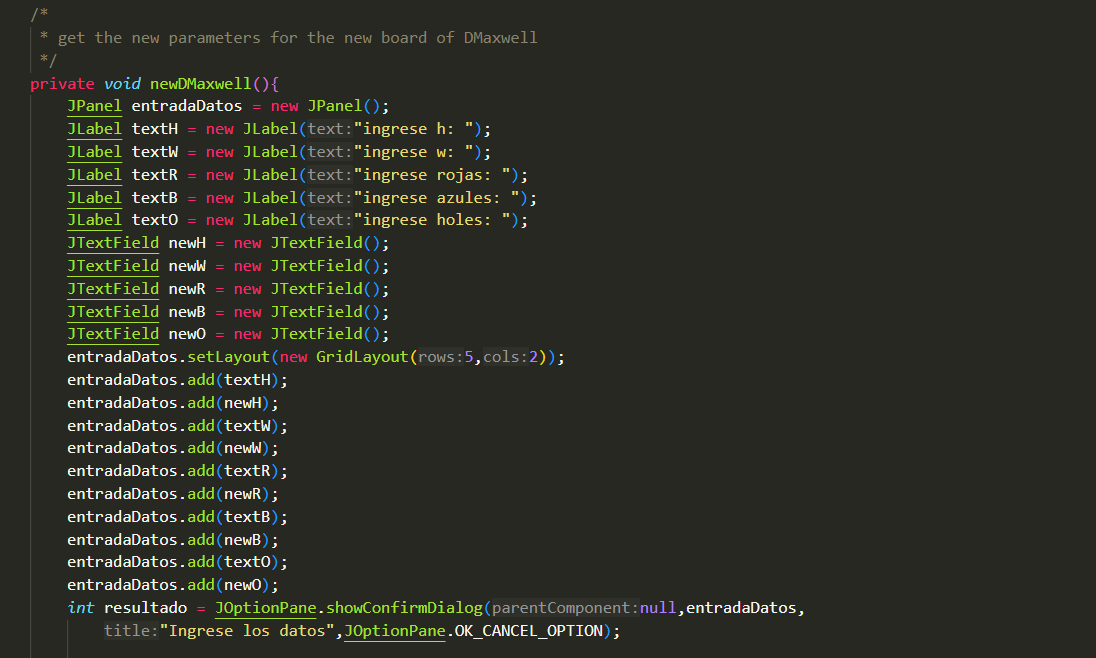
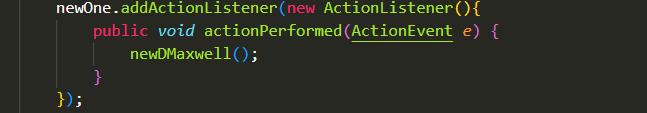
# **Ciclo 8: Cambiar el tamaño**

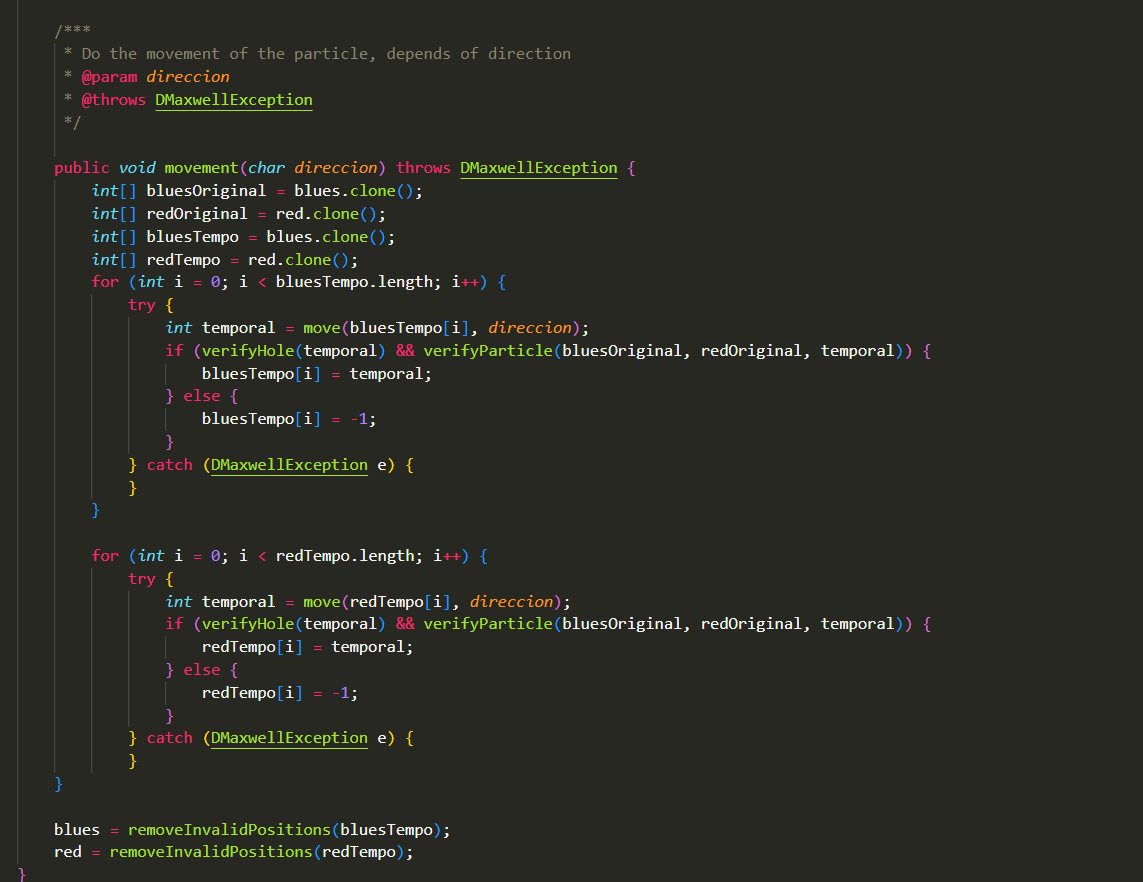
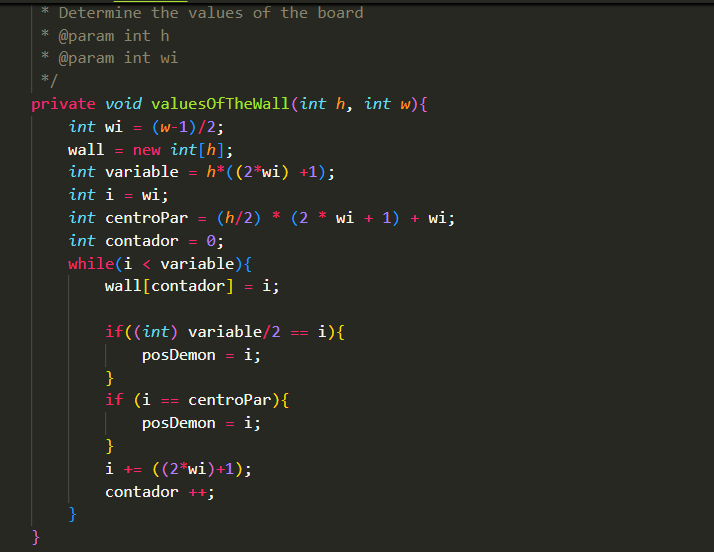
El objetivo es implementar este caso de uso. (En Astah Se encuentran los casos de Uso Definidos).

1. Expliquen los elementos a usar para implementar este caso de uso

Para desarrollar este caso de uso implementamos un botón “Genera uno nuevo”, al hacer uso de esto desarrollamos un método en el que por medio de JLabel y JTextField vamos a recibir los datos de la nueva configuración que el usuario quiere generar, solamente se pueden ingresar números, para esto especificamente tenemos una función que revisa esto, si es válida la información que el usuario ingresa se va a instancia un nuevo DMaxwell (h, 2\*w,r,b,o) se va a remover el tablero que esta por defecto y se va agregar este nuevo, utilizamos nuevamente el método refresh para que sea visible en la ventana.

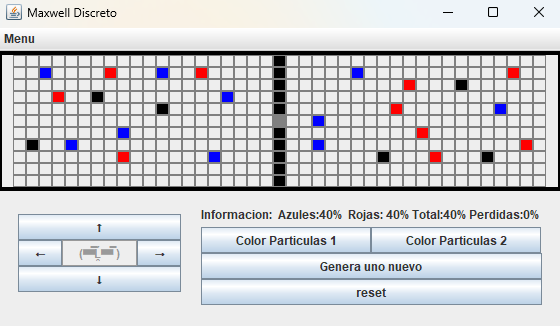
1. Implementen los elementos necesarios para cambiar el tamaño del juego



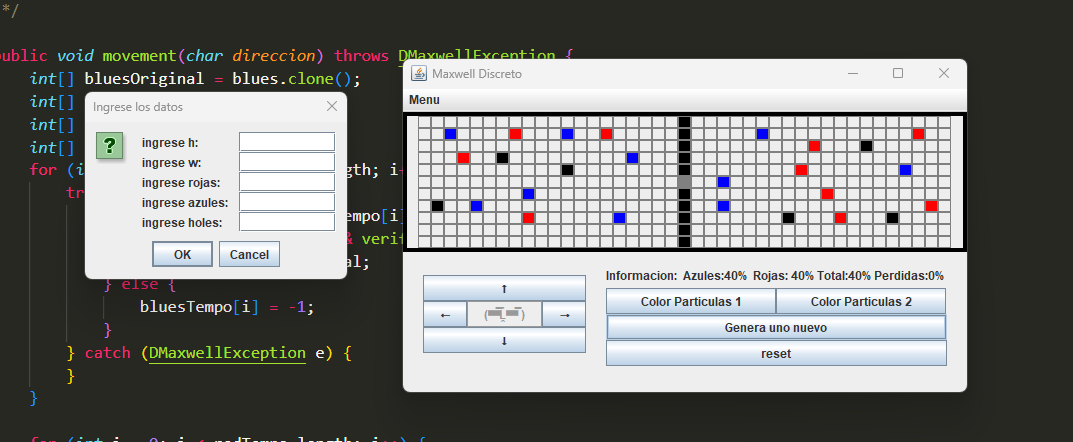


1. Ejecuten el caso de uso y capture las pantallas más significativas.

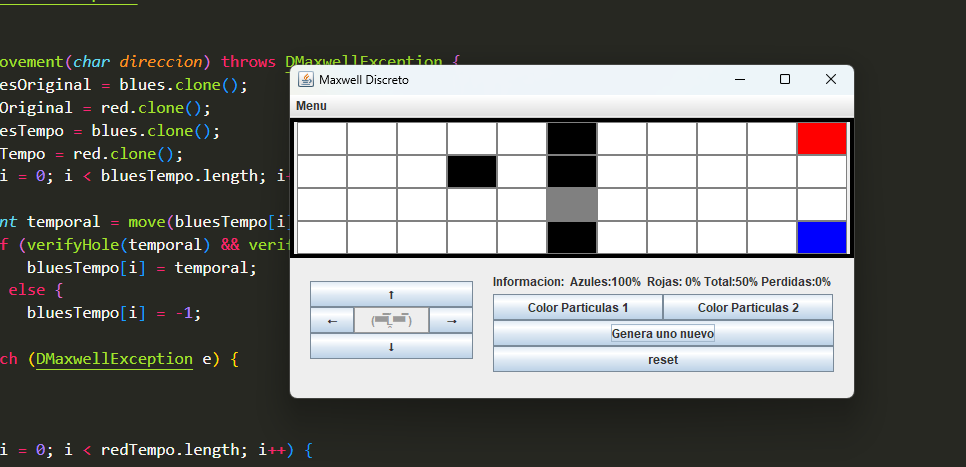
Inicialmente al ejecutar DMaxwellGUI, tenemos:



Al utilizar el botón de generar uno nuevo, se nos abre una ventana emergente:



Al ingresar valores como [4,5,1,1,1], obtenemos:



Lo que es acorde, con respecto a la configuración que se menciono anteriormente.

# Retrospectiva

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

Hombre)

Andrés Cardozo: 24 Horas

Tulio Riaño: 24 Horas

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

El estado actual del laboratorio es completo, se desarrollaron todos los ciclos incluyendo los bonos de manera acorde y correcta.

3. Considerando las prácticas XP del laboratorio de hoy ¿por qué consideran que son

Importante?

Consideramos que la práctica XP asociada a los test es la más importante en caso de que se presente un bug, lo podemos anticipar mediante casos de prueba para identificar su causa.

4. ¿Cuál consideran fue su mayor logro? ¿Por qué? ¿Cuál consideran que fue su mayor

problema? ¿Qué hicieron para resolverlo?

Nuestro mayor logro fue saber como implementar todo lo que esta relacionado con la capa de presentación (Interfaz Gráfica), ya qué puede llegar a ser confuso como manejar todo tipo de capas.

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

Resultados?

De nuevo resaltamos la comunicación y disposición para desarrollar el laboratorio, así mismo mantenemos nuestro compromiso por la ayuda mutua que se evidencia en el desempeño.

6. ¿Qué referencias usaron? ¿Cuál fue la más útil?

La mayoría de las referencias que se adjuntaron fueron demasiado útil a la hora de implementar el desarrollo correspondiente a la interfaz.

<https://docs.oracle.com/javase/7/docs/api/javax/swing/JFrame.html#setDefaultCloseOperation(int>)

<https://es.stackoverflow.com/questions/35656/cómo-puedo-concatenar-dos-arreglos>

<https://oregoom.com/java/jcolorchooser/>

[ArrayList to Array Conversion in Java : toArray() Methods | GeeksforGeeks](https://www.geeksforgeeks.org/arraylist-array-conversion-java-toarray-methods/)