**LABORATORIO NRO. 5**

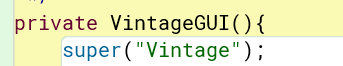
**JEISSON CASALLAS Y CAMILO MURCIA**

**Vintage**

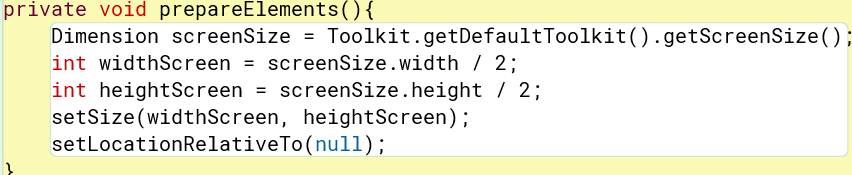
**Ciclo 0: Ventana vacía - Salir**

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

**1. Construyan el primer esquema de la ventana de Vintage únicamente con el título “Vintage”. Para esto cree la clase VintageGUI como un JFrame con su creador (que sólo coloca el título) y el método main que crea un objeto VintageGUI 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)**



**2. 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.**



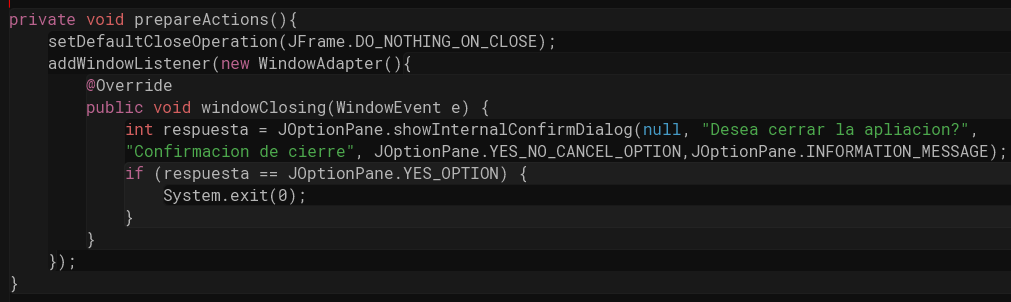
**3. Traten de cerrar la ventana. ¿Termina la ejecución? ¿Qué deben hacer para terminar la ejecución?**

No no termina ejecucion, se debe enviar un mensaje antes para confirmar, y seguido a esto, terminar la aplicación con System.exit(0)

**4. 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?**

La función setDefaultCloseOperation en Java se utiliza para establecer la operación por defecto cuando el usuario intenta cerrar la ventana (JFrame) de la aplicación. Esta función toma como argumento una constante que indica la acción que se debe realizar.

**5. 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.**



* Inicialmente plantemos este método, pero usando el confirmDialog es más adecuado para este requisito.

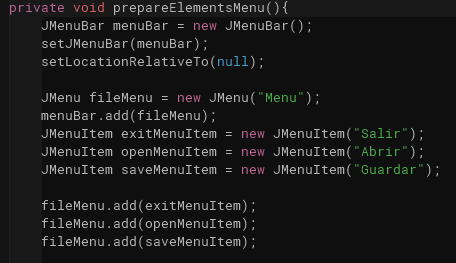
**Ciclo 1: Ventana con menú – Salir**

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.

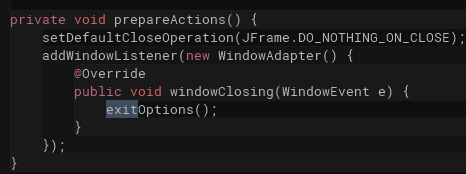
**1. Expliquen los componentes visuales necesarios para este menú. ¿Cuáles serían atributos y cuáles podrían ser variables del método prepareElements? Justifique.**

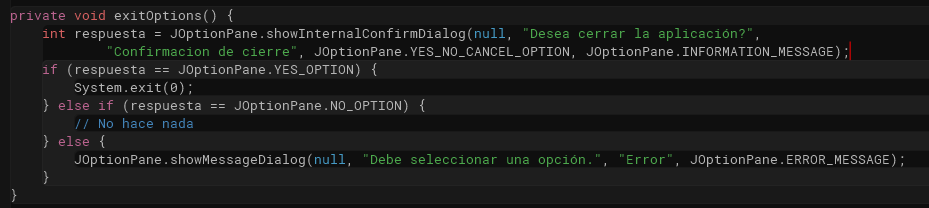
En el caso de los componentes visuales, tenemos atributos de la clase, que vienen siendo el Jframe y el JMenuBar que son la parte de la estructura principal para la implementación del menú. Como variables locales del metodo prepareElements tenemos “Dimension”,”widthScreen”,”heightScreen” que ya vienen siendo locales del metodo utilizadas al momento de definir las dimensiones de la pantalla.

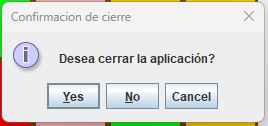
**2. Construya la forma del menú propuesto (prepareElements - prepareElementsMenu). Ejecuten. Capturen la pantalla.**



**3. 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 de la biblioteca Swing de Java que proporciona al usuario la capacidad de seleccionar archivos o directorios. Generalmente se usa para abrir o guardar archivos tal como se requiere en este ciclo.

**ShowOpenDialog** es un metodo que nos genera un cuadro para que el usuario seleccione un archivo que desea abrir, este se maneja por medio de un valor entero que nos indica si el usuario selecciono abrir el archivo o si cancelo la operación.

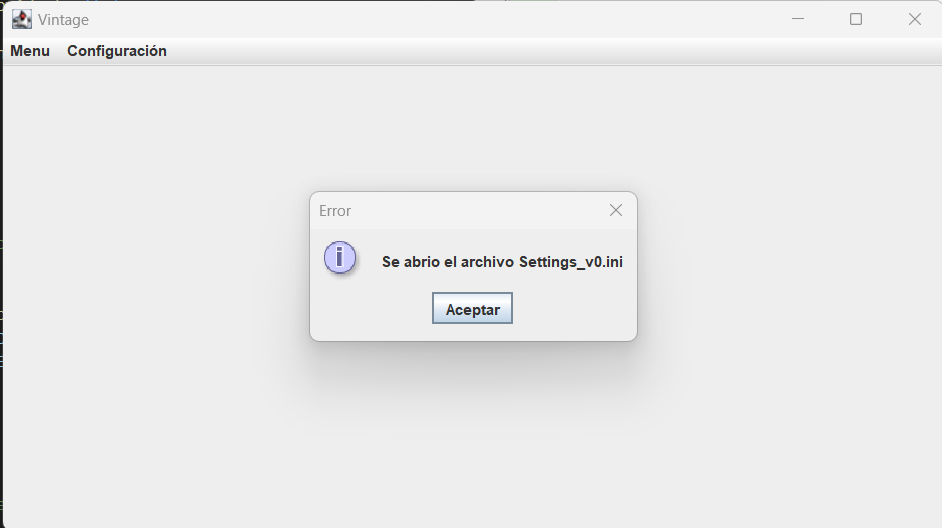
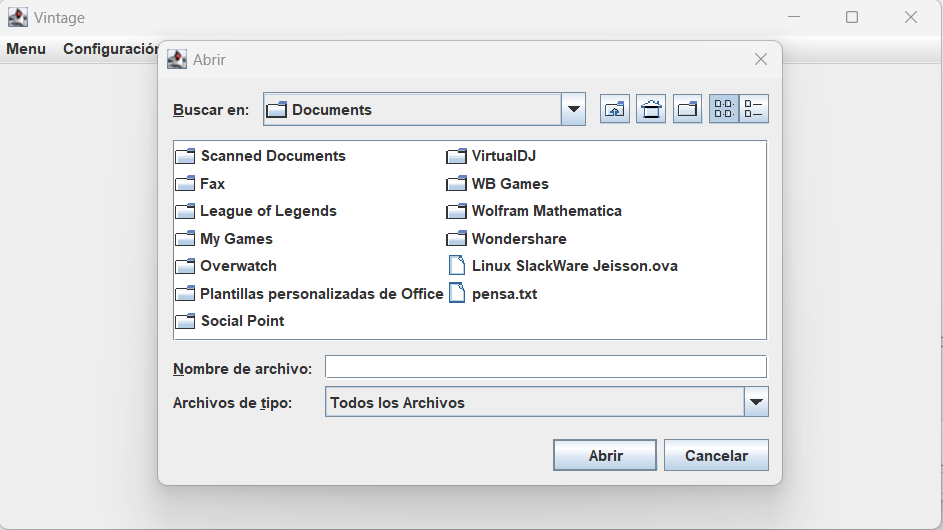
**ShowSaveDialog** con este metodo se genera el cuadro de dialogo para que el usuario seleccione un archivo el cual desea guardar, se maneja también por un entero que nos indica si el usuario guardo el archivo o si cancelo la operación.

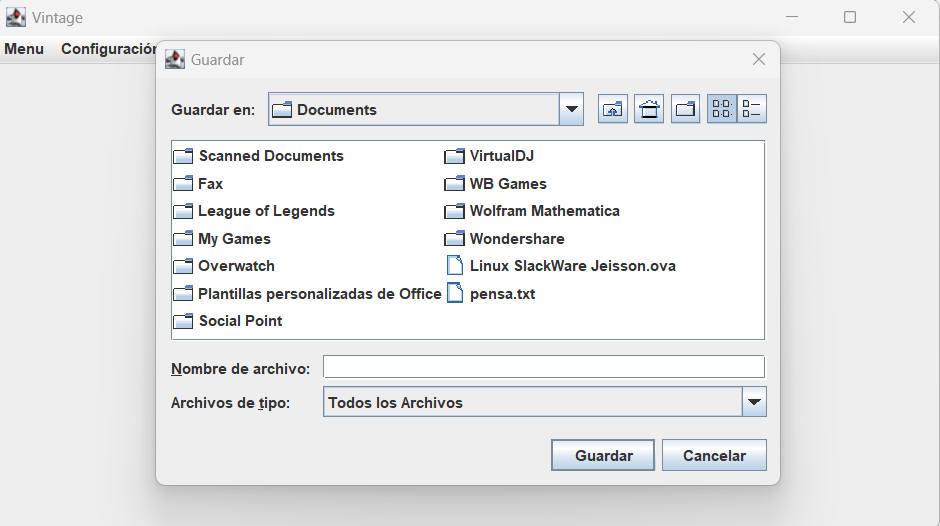
**GetSelectedFile** es una forma de retornar un archivo seleccionado por el usuario.

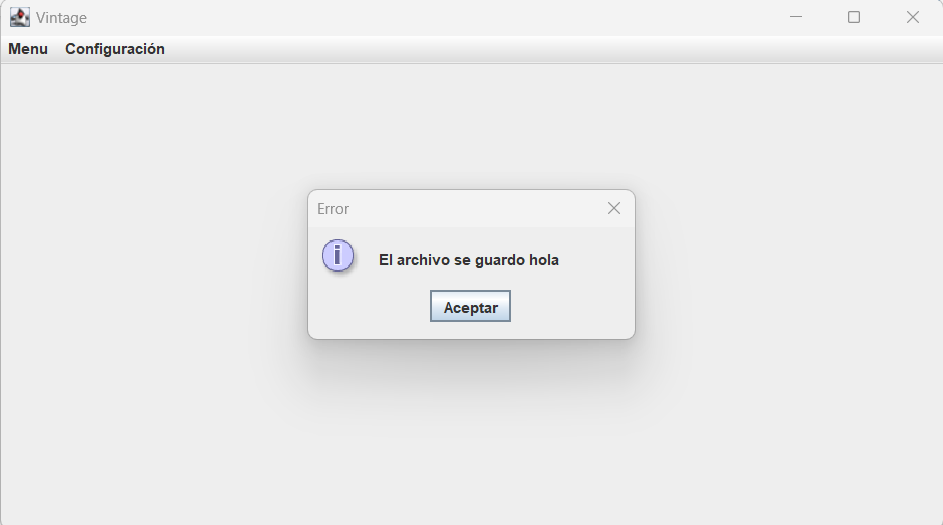
**2. Implementen parcialmente los elementos necesarios para salvar y abrir. Al seleccionar los archivos indique que las funcionalidades están en construcción detallando la acción y el nombre del archivo seleccionado.**

Implementación en la Clase de presentación VintageGUI.

**3.** Ejecuten las dos opciones y capturen las pantallas más significativas.



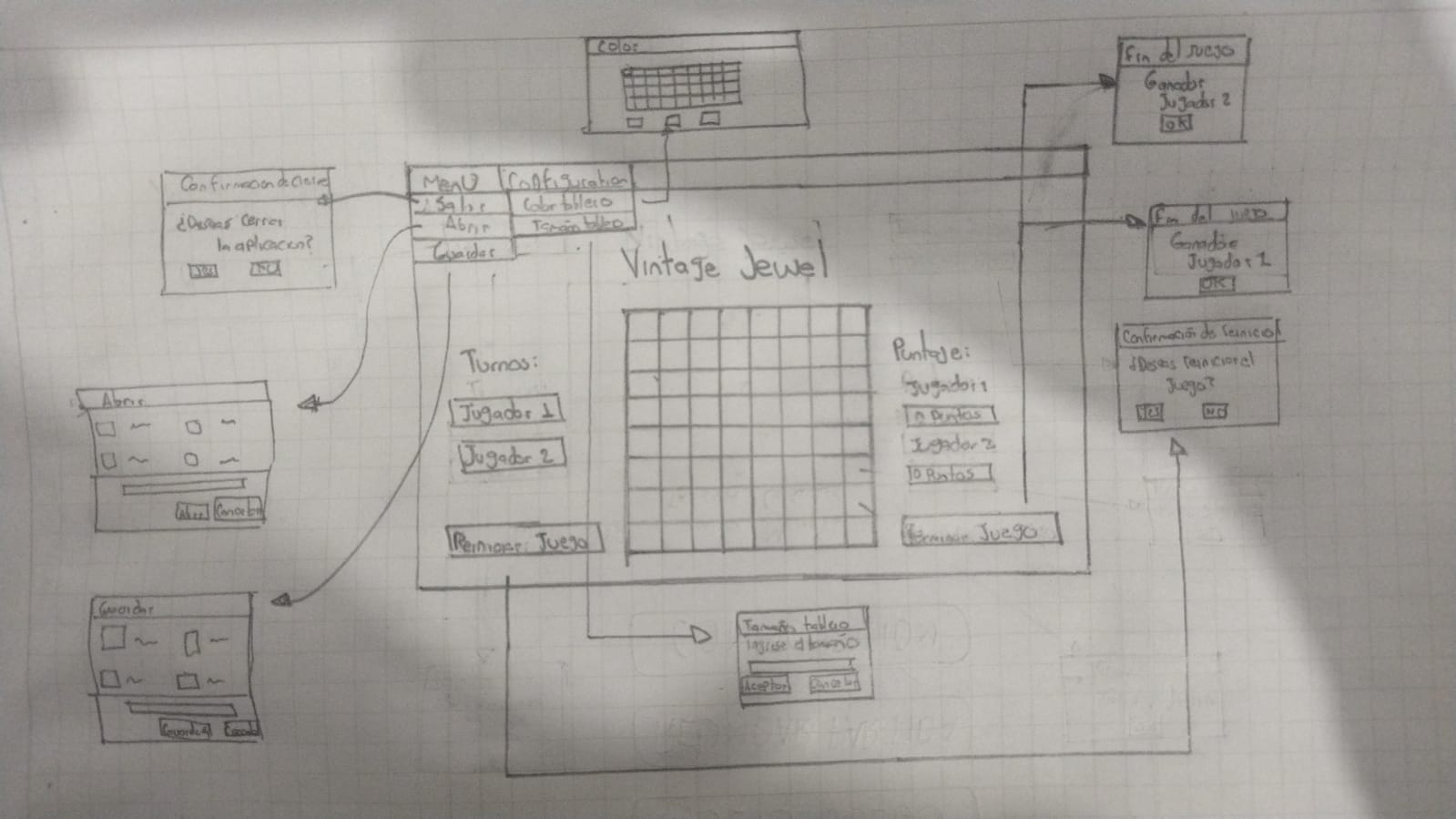




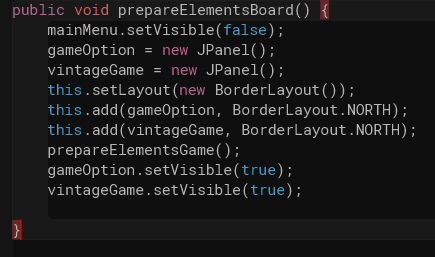
**Ciclo 3: Forma de la ventana principal**

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.**



**2.** Continúe con la implementación definiendo los atributos necesarios y extendiendo el método prepareElements(). Para la zona del tablero defina 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.



**3.** Ejecuten y capturen esta pantalla.

**Ciclo 4: Cambiar color**

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

**Vista (View):**

**Botón de Cambiar Color:**

Un botón en la interfaz gráfica que activará la opción de cambiar el color.

**Lógica para Cambiar el Color:**

Métodos en la vista que permitan cambiar el color del tablero y las fichas.

**JColorChooser:**

Un componente que proporciona una interfaz para que el usuario elija un color.

**Controlador (Controller):**

**Manejador de Eventos para el Botón:**

Un ActionListener que responda cuando se haga clic en el botón de cambiar color.

**Lógica para la Interacción:**

Métodos en el controlador que interactúen con el JColorChooser y actualicen la vista.

**2. Detalle el comportamiento de JColorChooser especialmente el método estático showDialog**

**JColorChooser:**

Es un componente Swing que permite al usuario seleccionar un color.

Proporciona un diálogo modal para elegir un color.

**showDialog:**

Este método estático muestra el diálogo de selección de color.

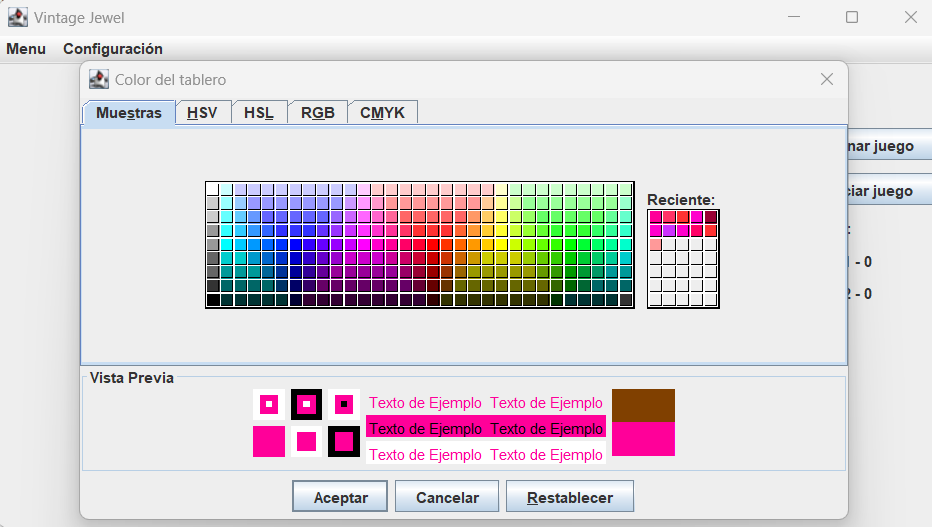
Toma varios argumentos, incluido el componente padre, el título del diálogo y el color inicial.

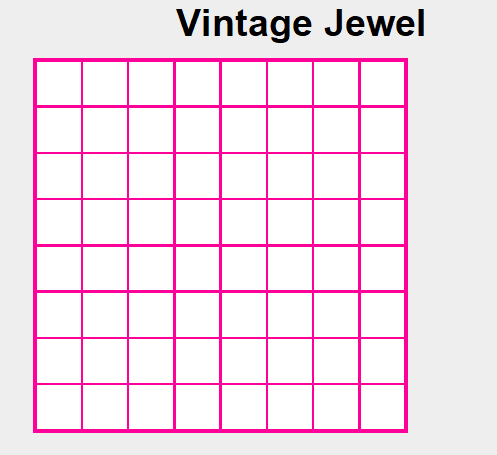
Devuelve el color seleccionado por el usuario o null si el usuario cancela el diálogo.

**3. Implementen los componentes necesarios para cambiar el color del tablero y de las fichas.**

Implementación en la clase de presentación.

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



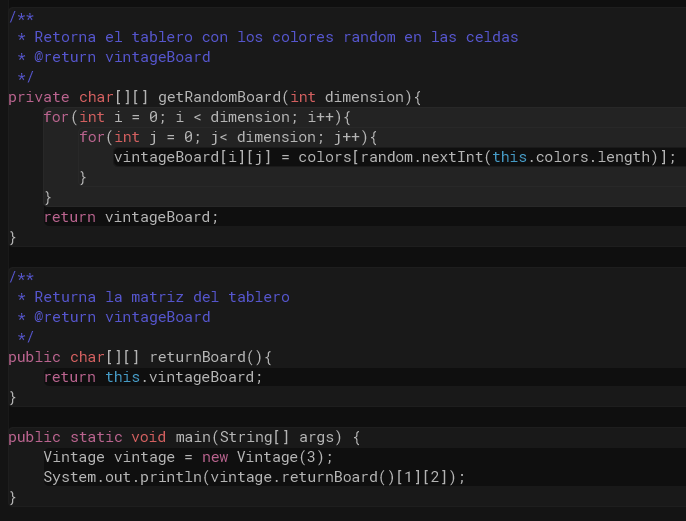


Debido a que no tiene sentido cambiarles el color a las fichas, la implementación esta guiada al color inicial que va a tener el tablero.

**Ciclo 5: Modelo Vintage**

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





**2.** Ejecuten las pruebas y capturen el resultado.

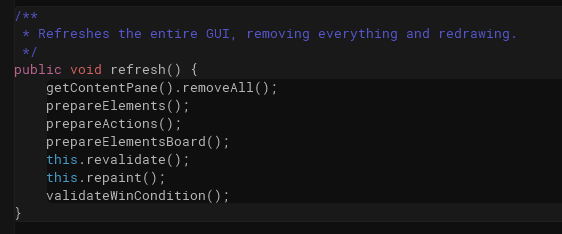
**Ciclo 6: Jugar [En \*.java y lab05.doc]**

El objetivo es implementar el caso de uso jugar.

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



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



**3. Expliquen los elementos necesarios para implementar este caso de uso.**

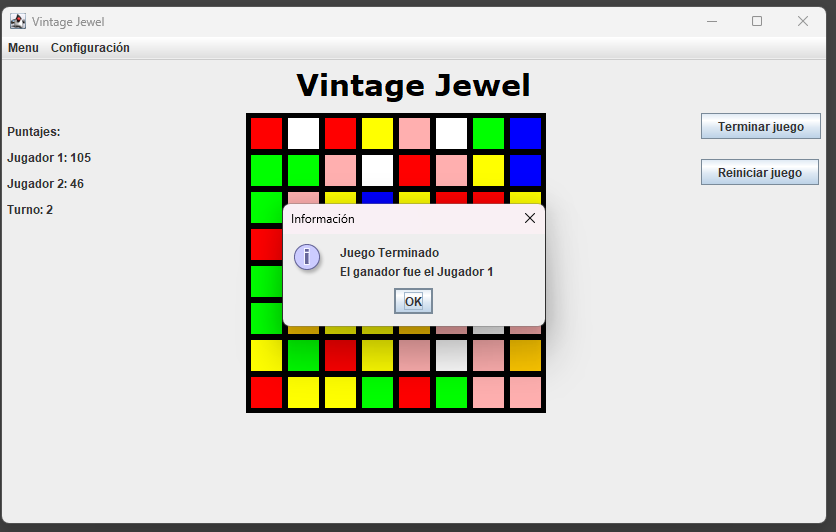
Se necesita la capa de dominio que será la encargada de manejar las acciones que se realicen por medio de los botones del GUI, y donde se validara si hay 3 o más joyas alineadas.

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

La cantidad de oyentes es la misma a la cantidad de botones presentes en el panel llamado game, debido a que estos botones son las joyas, por ende, estos oyentes se crean en cada botón de game.

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



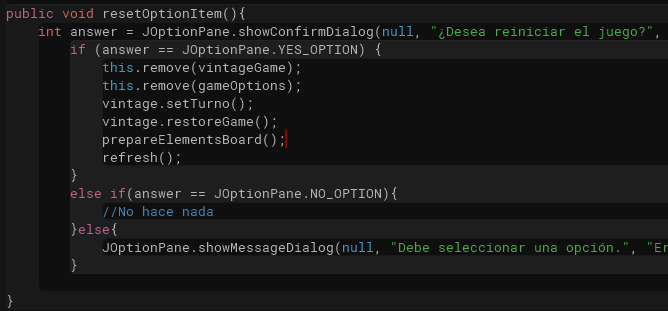


**Ciclo 7: Reiniciar**

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

Para reiniciar el juego, se debe crear un oyente en el botón de reinicio. Este oyente llamará a un método de reinicio que limpiará el tablero, reiniciará el turno y restablecerá la matriz inicial del juego. Al final de este proceso, el tablero se volverá a crear con los elementos iniciales.

**2.** Implementen los elementos necesarios para reiniciar



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





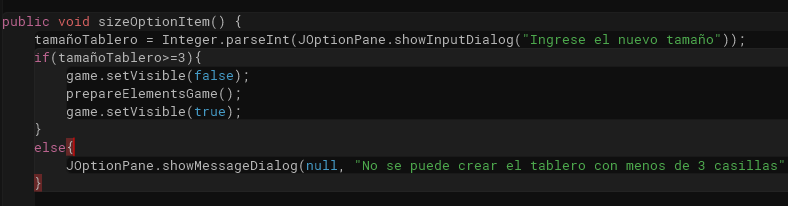


**Ciclo 8: Cambiar el tamaño**

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

Se debe crear un atributo global con el tamaño del tablero, que inicialmente debe ser de 8x8 como indica el PDF, pero que se pueda modificar según el usuario decida, teniendo en cuenta la condición de que no puede ser menor de 3x3, para poder hacer las combinaciones de las joyas.

**2.** Implementen los elementos necesarios para cambiar el tamaño del juego

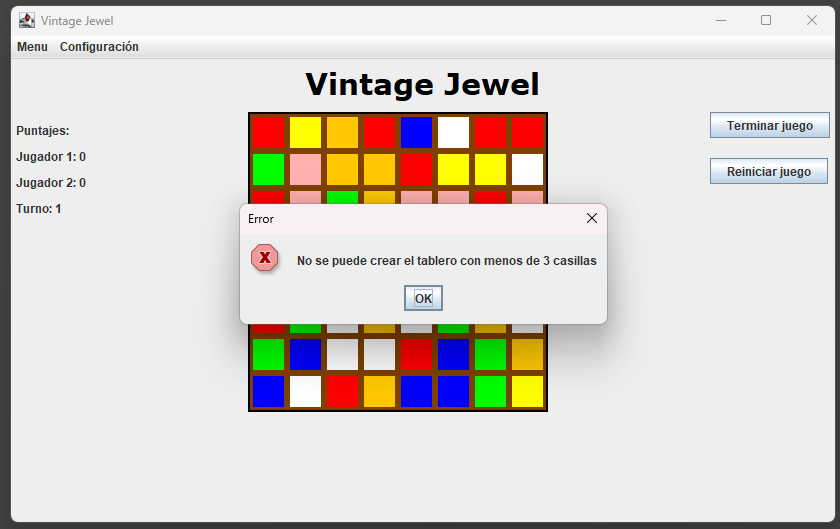


**3.** Ejecuten el caso de uso y capture las pantallas más significativas.

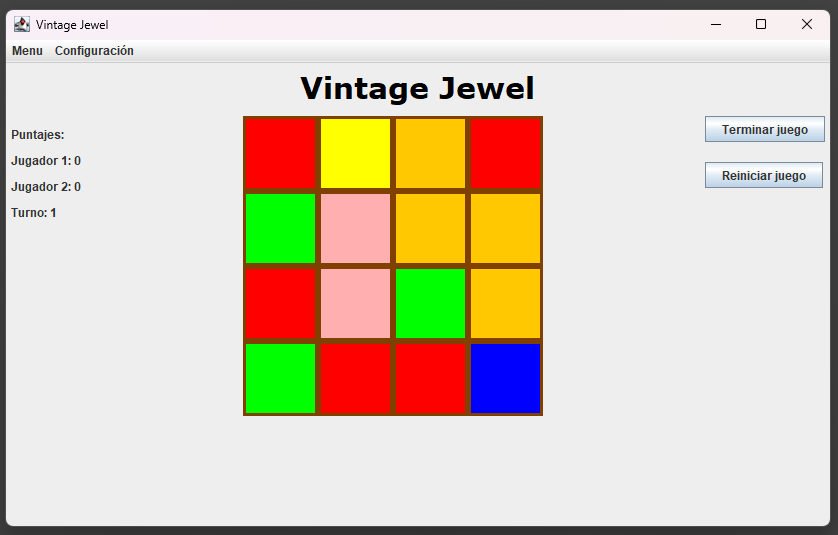
Original:



Menor a 3:



Mayor que 3:



RETROSPECTIVA

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

28 horas por persona

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

Se encuentra realizado en su totalidad, debido a que se le dedico el tiempo necesario

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

Simplicidad en el código, pues dejamos las cosas lo más funcional posible y con el menos código requerido.

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

El mayor logro fue lograr que todo el simulador funcionara, pues es bastante satisfactorio el ver como se logra darle funcionalidad.

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

No sé tuvo mayor problema tecnico en este laboratorio.

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

Cada uno compartió ideas y aporto lo necesario para el laboratorio. Nos comprometemos a seguir así.