**PROGRAMACIÓN ORIENTADA A OBJETOS**

**INTERFAZ**

**2020-1**

**Laboratorio 5/6**

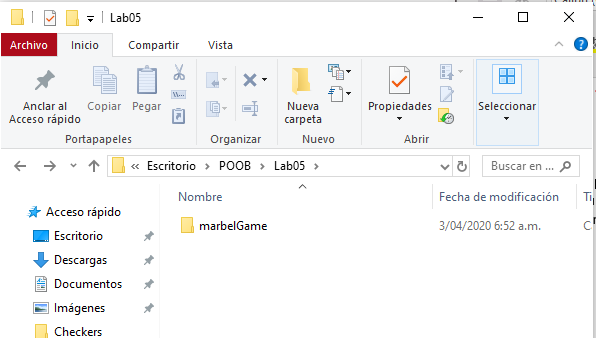
**Santiago Martínez Martínez**

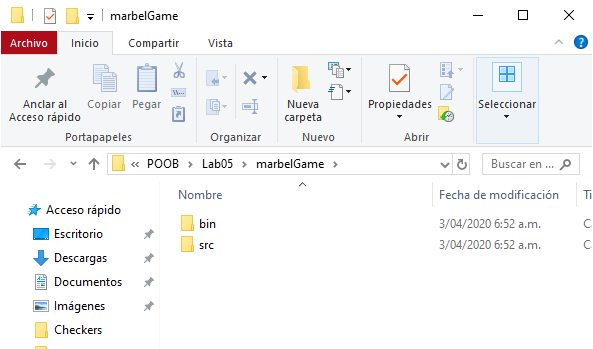
**DESARROLLO**

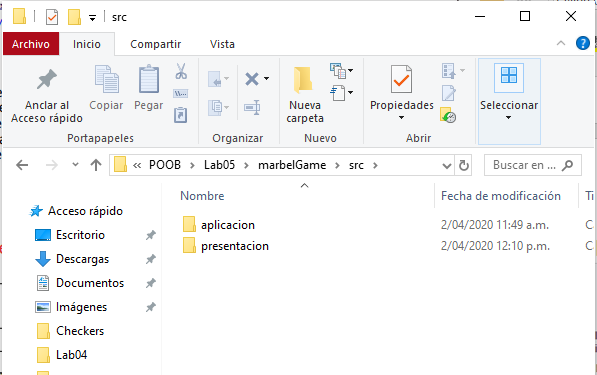
**Directorios**

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

1. Preparen un directorio llamado **marbelGame** con los directorios src y bin y los subdirectorios presentación y aplicación.



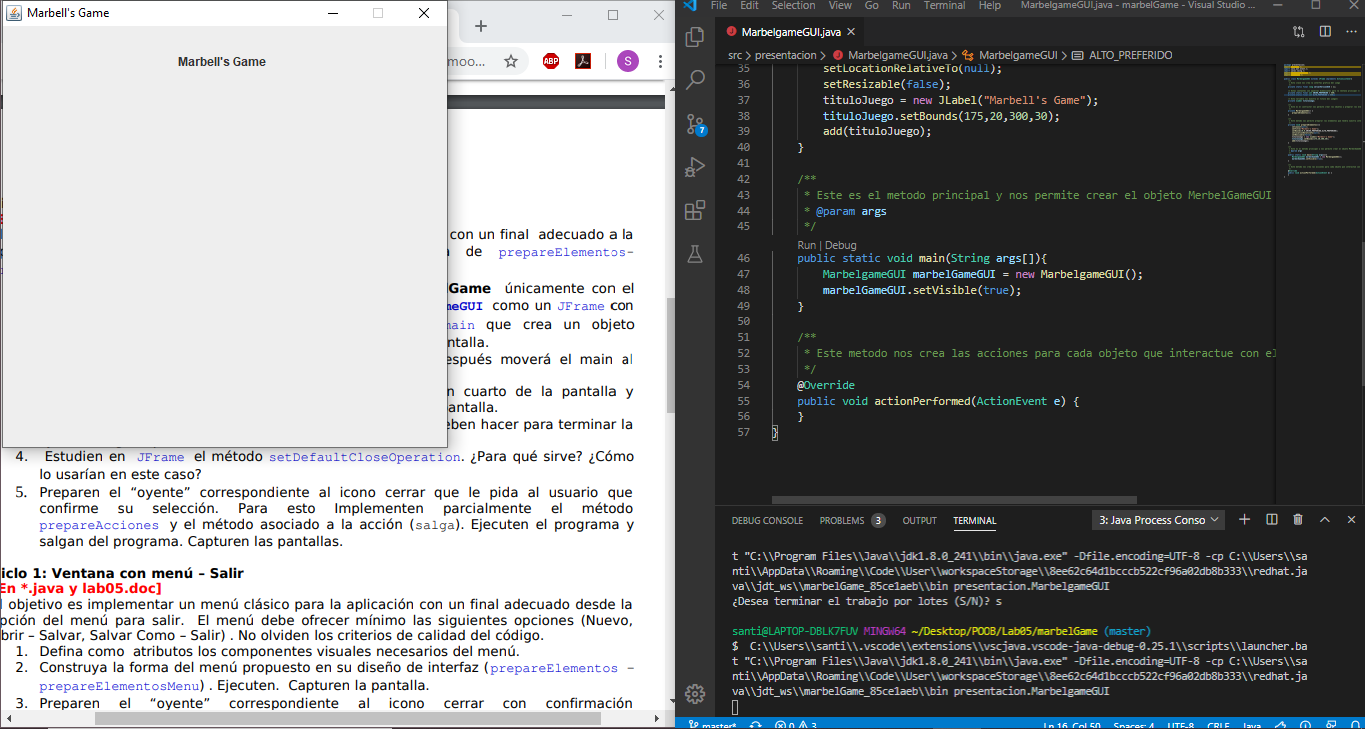




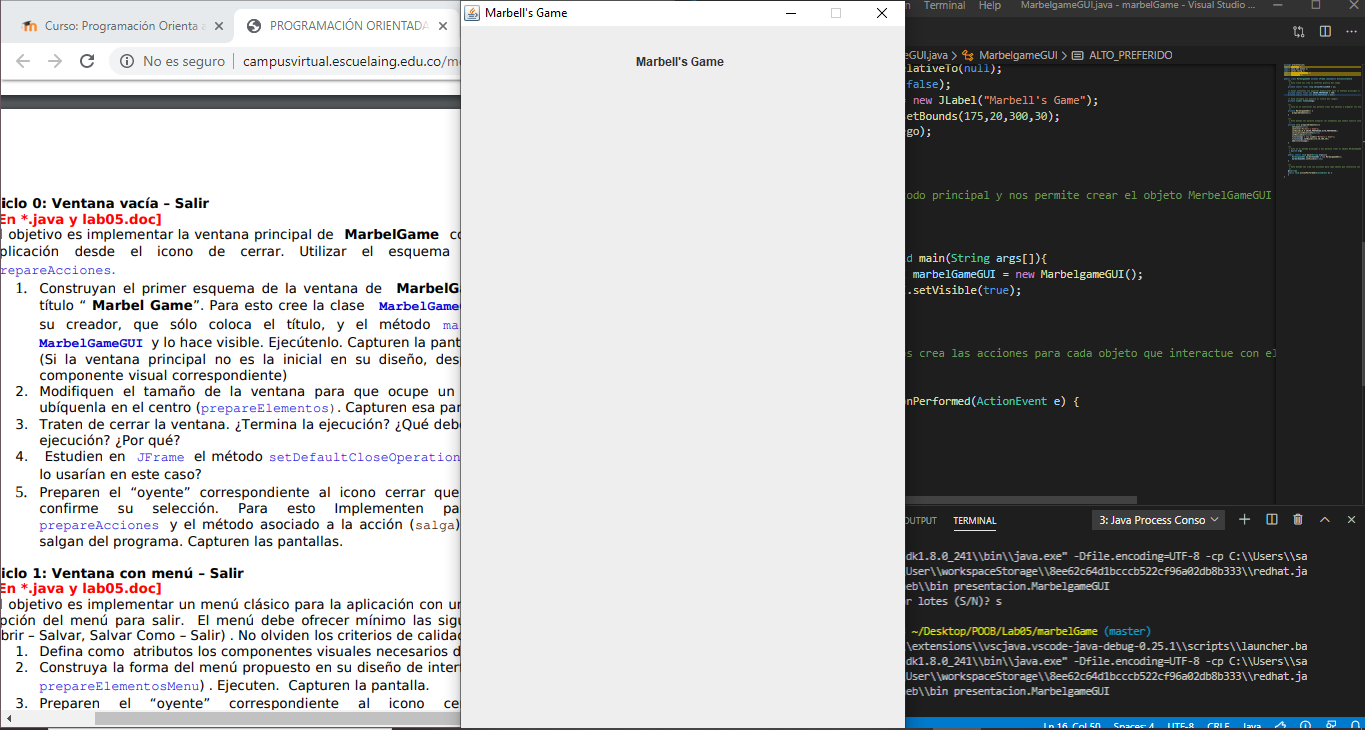
**Ciclo 0: ventana vacía – Salir**

El objetivo es implementar la ventana principal de **MarbelGame** con un final adecuado a la aplicación desde el icono de cerrar. Utilizar el esquema de **prepareElementos-prepareAcciones**.

1. Construyan el primer esquema de la ventana de **MarbelGame** únicamente con el título **“Marbel Game”**. Para esto cree la clase MarbelGameGUI como un JFrame con su creador, que sólo coloca el título, y el método main que crea un objeto **MarbelGameGUI** y lo hace visible. Ejecútenlo. Capture pantalla.



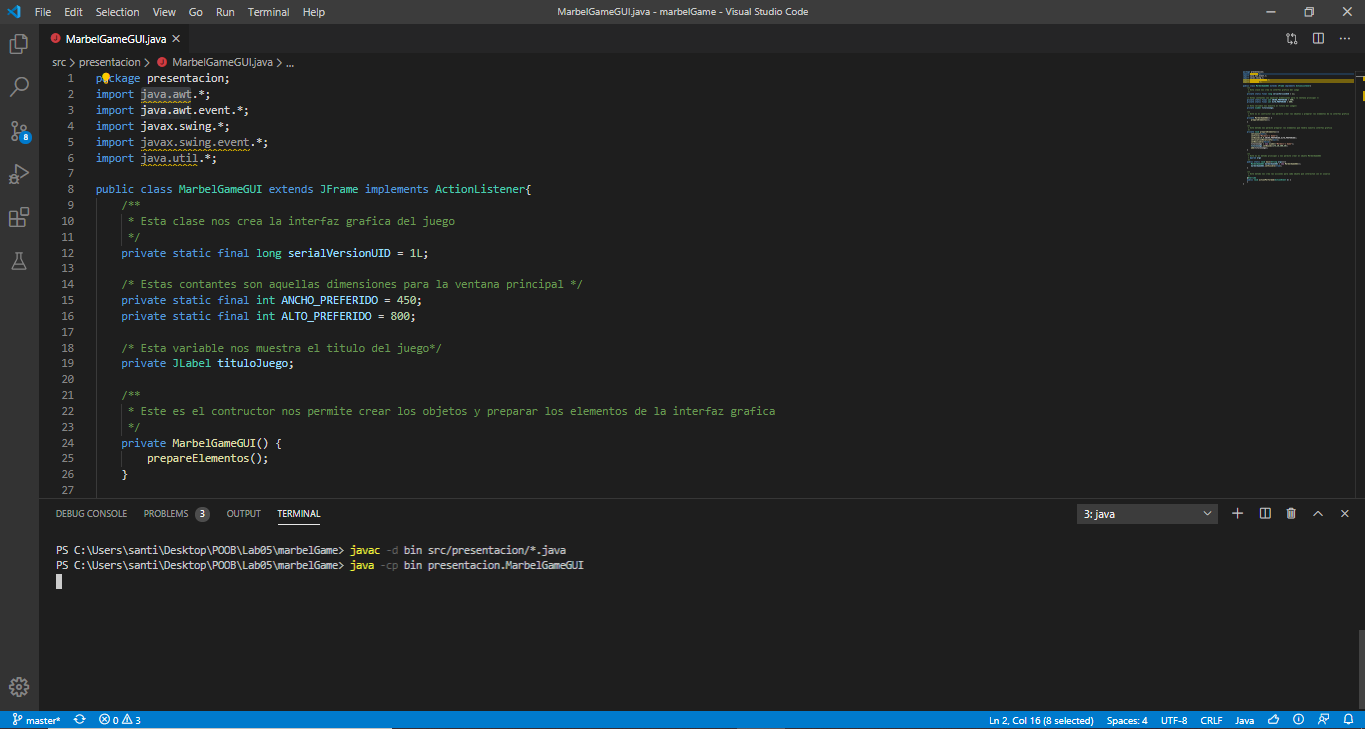
1. Modifiquen el tamaño de la ventana para que ocupe un cuarto de la pantalla y ubíquenla en el centro **(prepareElementos)**. Capturen esa pantalla.



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

Al cerrar la ventana no se termina el programa, solo se cierra la interfaz gráfica, debería de acabar el programa cuando oprimimos ese boto.

Debemos de configurar ese botón y añadirle una acción para que al ser oprimido se acabe el proyecto.

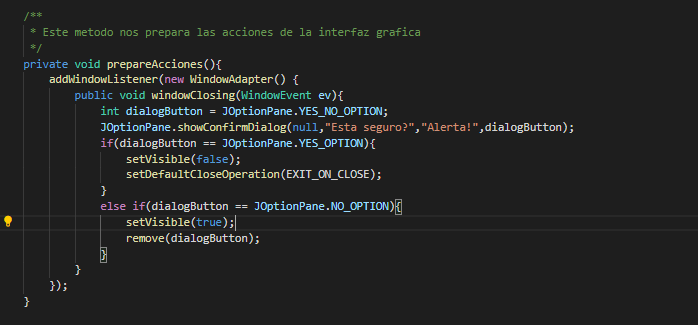


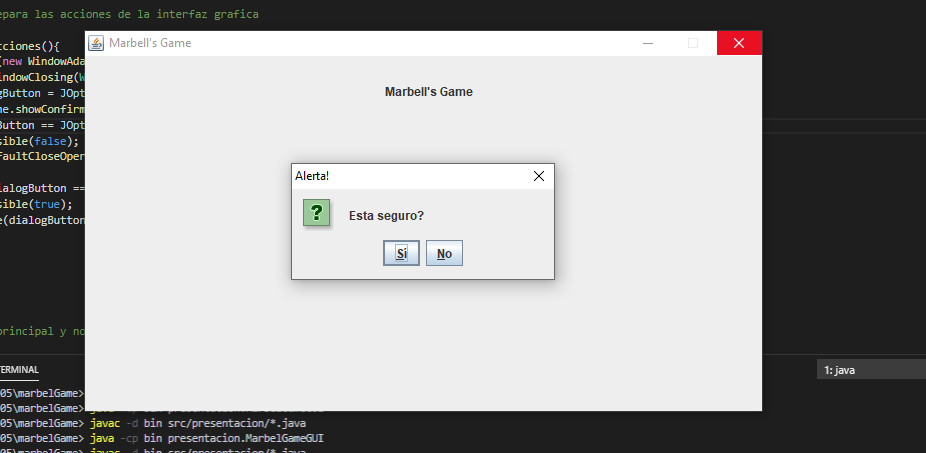
1. Estudien en **JFrame** el método **setDefaultCloseOperation**. ¿Para qué sirve? ¿Cómo lo usarían en este caso?

Establece la operación que sucederá por defecto cuando el usuario cierre la ventana.

En este caso lo podríamos utilizar para poder configurar la acción al cerrar la ventana.

1. Preparen el “oyente” correspondiente al icono cerrar que la pida al usuario que confirmen su selección. Para esto implementen parcialmente el método **prepareAcciones** y el método asociado a la acción (salga). Ejecuten el programa y salgan del programa. Capturen las pantallas.

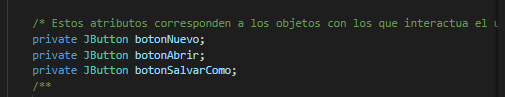




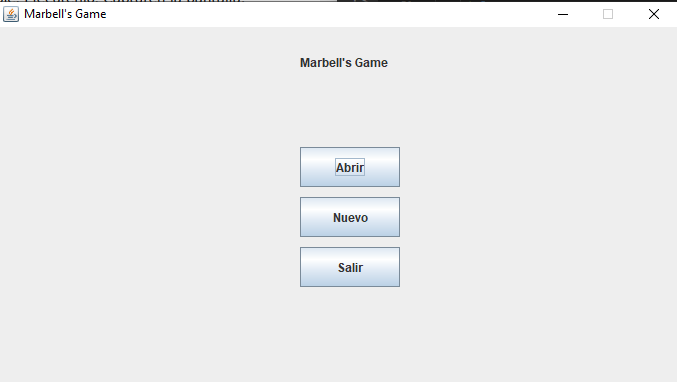
**Ciclo 1: Ventana con menú – Salir**

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, Salvar como - salir). No olviden los criterios de calidad del código.

1. Defina como atributos los componentes visuales necesarios del menú.



1. Construya la forma del menú propuesto en su diseño de interfaz (prepareElementos - prepareElementosMenu). Ejecuten. Capturen la pantalla.



1. Preparen el “oyente” correspondiente al icono cerrar con confirmación (prepareAcciones). 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.**

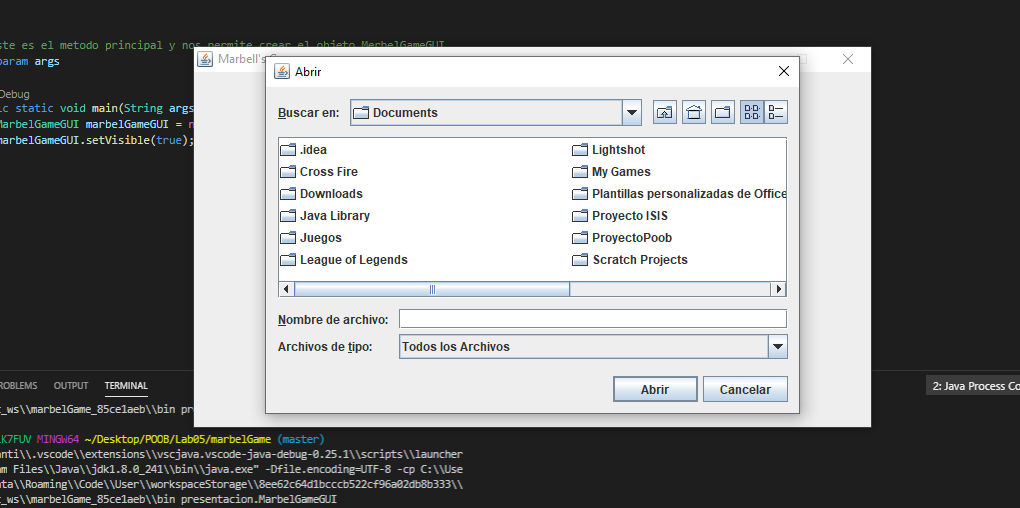
* Componente **JFileChooser**: **JFileChooser**  es una clase java que nos permite mostrar fácilmente una ventana para la selección de un fichero, es decir una ventana para seleccionar los archivos de nuestro computador.
* **JFileChooser:** Es el constructor de la clase JFileChooser, con este método podemos instancias el objeto.
* **ShowOpenDialog:** Este método nos abre una ventana que nos abre el FileExplorer, para la selección de archivos.
* **ShowSaveDialog:** Este método nos abre la ventana para salvar los archivos que deseamos.
* **GetSelectedFile:** Este método retorna el archivo seleccionado

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

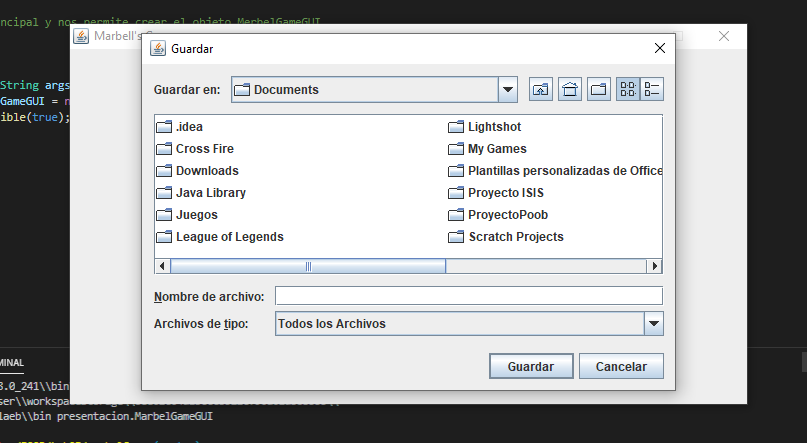
Archivo .java

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

**Botón Abrir:**



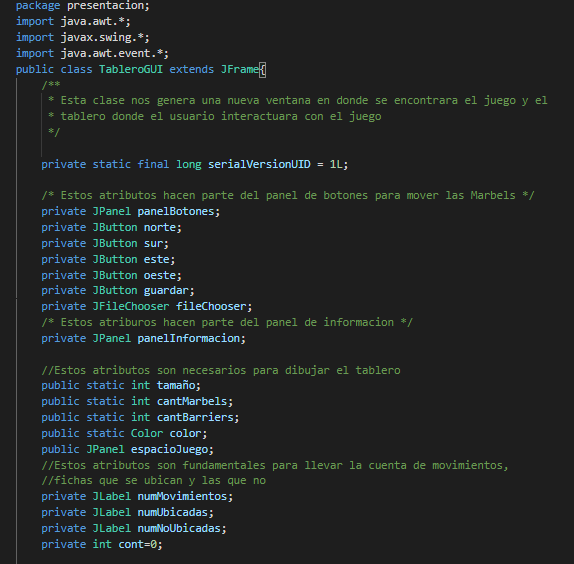
**Botón Guardar:**



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

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

1. Definan como atributos privados todos los componentes visuales necesarios.



1. Continúe con la implementación del método prepareElementos().

Para la zona del tablero defina un método prepareElementosTablero() y un método refresque() que actualiza la vista del tablero considerando, por ahora, un tablero inicial por omisión (el ejemplo del trabajo en clase) Este método lo vamos a implementar reamente en otros ciclos. Ejecuten y capturen esta pantalla.

Archivo .java.

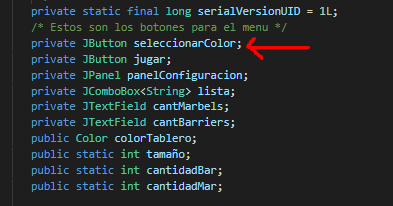


**Ciclo 4: Cambiar color**

El objetivo es implementar este caso de uso.

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

Necesitamos un atributo de tipo JButton al cual le daremos la acción de que al interactuar con el usuario nos muestre el panel para escoger el color.

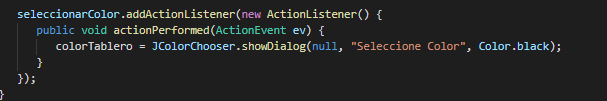


1. Detalle el comportamiento de **JColorChooser** especialmente el método estático **showDialog**

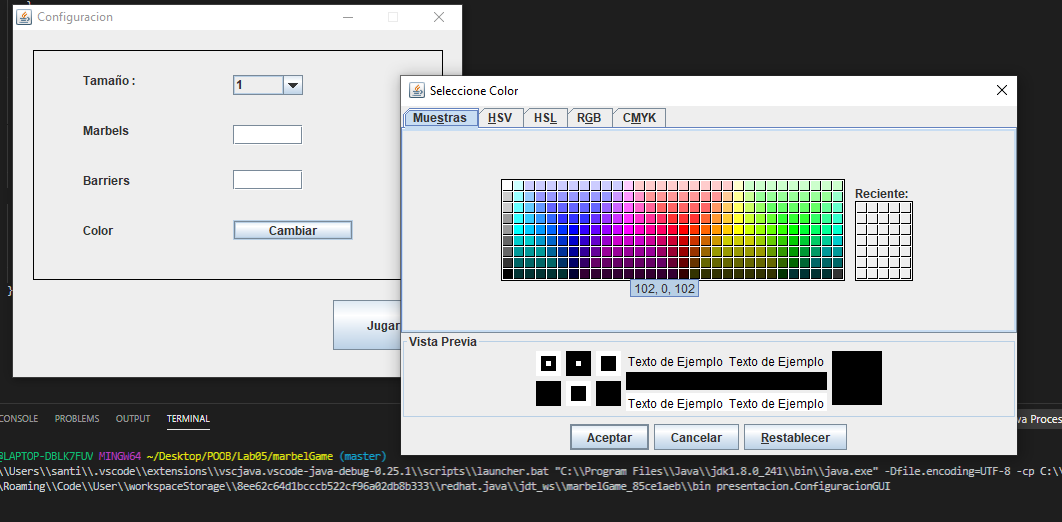
El componente **JColorChooser** nos ayuda a seleccionar un color de una forma mas atractiva para el usuario

El método **showDialog** el método estatico showDialog mustra un cuadro de diálogo de selección de color modal.

1. Implementen los componentes necesarios para cambiar el color del tablero (inicialmente blanco-negro)



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



**Ciclo 5: Modelo MarbelGame**

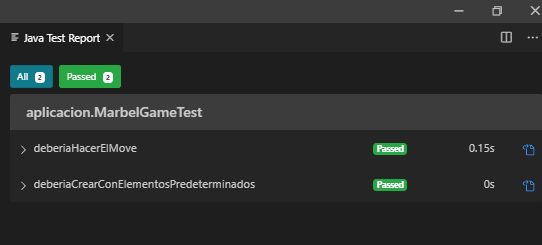
EL objetivo es implementar la capa de aplicación para MarbelGame.

1. Construyan los métodos básicos del juego **(No olvide MDD Y TDD).**

Archivo \*.java y Archivo \*.Astah

1. Ejecuten las pruebas y capturen el resultado.

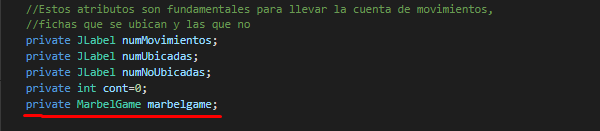


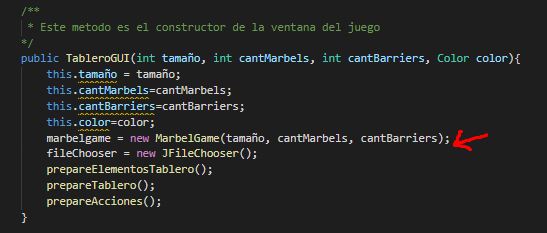


**Ciclo 6: Jugar**

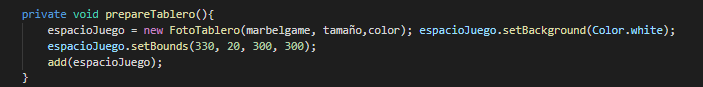
El objetivo es implementar el caso de uso jugar.

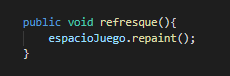
1. Adicione a la capa de presentación el atributo correspondiente al modelo.





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





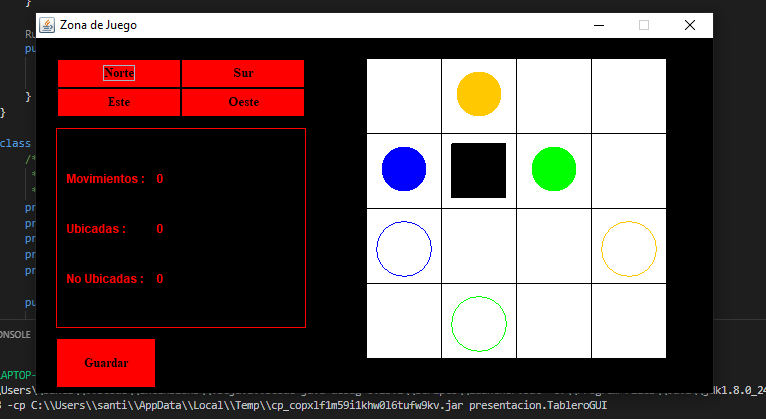
1. Expliquen los elementos necesarios para implementar este caso de uso.

Archivo \*.java

1. Implementen los componentes necesarios para jugar.

Archivo \*.java

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



**Ciclo 7: Reiniciar**

El objetivo es implementar este caso de uso.

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

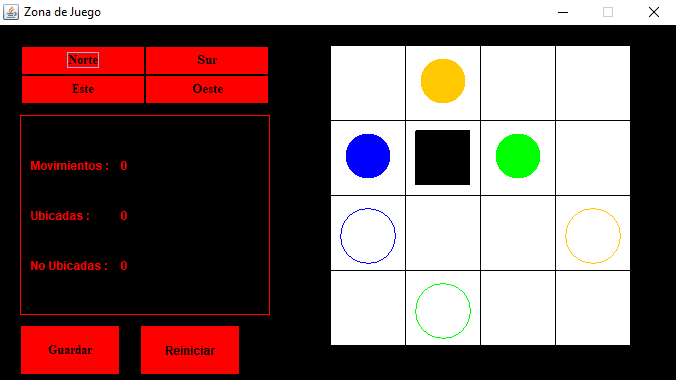
Es necesario implementar un nuevo botón, el cual al interactuar con el usuario reinicie la configuración del tablero a la inicial.

1. Implementen los elementos necesarios para reiniciar

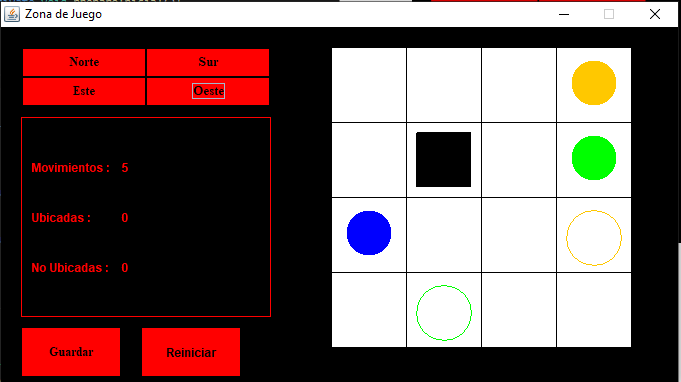
Archivo \*.java

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

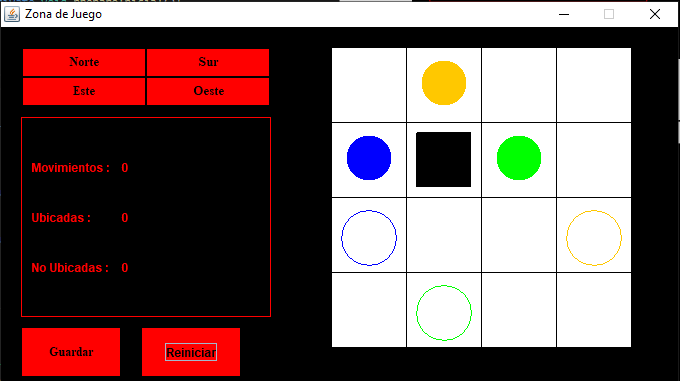
**Sin movimientos:**



**Al realizar movimientos:**



**Al oprimir el botón reiniciar:**



**Ciclo 8: Cambiar la configuración del juego**

El objetivo es implementar este caso de uso.

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

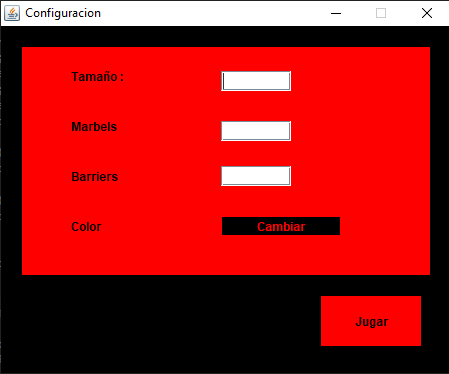
Para poder lograr este caso de uso, debemos de crear un nuevo JFrame el cual sea una ventana específicamente diseñada para poder configurar nuestra zona de juego.

1. Implementen los elementos necesarios para cambiar la configuración del juego.

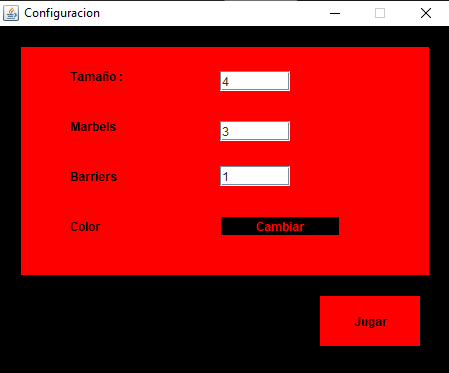
Archivo \*.java

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

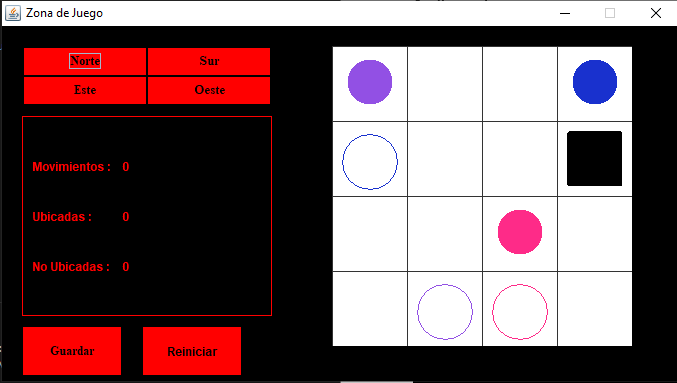
**Ventana de configuración:**



**Configurando:**



**Resultado en ventana de Juego:**



**RETROSPECTIVA**

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

Santiago – 50 horas

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

Creo que está completo el laboratorio.

1. Considerando la práctica XP del laboratorio ¿por qué consideran que es importante?

Repartir bien la carga en el tiempo que se nos da.

1. ¿Cuál consideran fue su mayor logro? ¿Por qué? ¿Cuál consideran que fue si mayor problema? ¿Qué hicieron para resolverlo?

Poder implementar el método **move**, ya que tocaba mover todos los objetos a la vez.

La mayor dificultad fue crear el JPanel para el tablero, se resolvió investigando y leyendo la documentación de Java.

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

Estoy solo en el grupo.