# PROGRAMACIÓN ORIENTADA A OBJETOS INTERFAZ

**2020-1**

**Laboratorio 5/6**

**Miguel Ángel Fúquene**

**Iván Camilo Rincon Saavedra**

## OBJETIVOS

1. Desarrollar una mini aplicación gráfica
2. Experimentar el comportamiento de las ventanas JFrame, JDialog y JOptionPane
3. Seleccionar los lienzos más apropiados para un diseño: JPanel, JScrollPane, JTabbedPane
4. Revisar las posibilidades de los estilos:FlowLayout, BorderLayout y GridLayout
5. Ensayar algunos componentes básicos: JLabel, JTextField, JButton, JMenuBar,
6. Ensayar algunos componentes especiales: JFileChooser y JColorChooser
7. Implementar el esquema de manejo de eventos con clases anónimas
8. Vivenciar las prácticas XP : [Acceptance tests](http://www.extremeprogramming.org/rules/functionaltests.html) *are run often and the score is published*

*When* [a bug is found](http://www.extremeprogramming.org/rules/bugs.html) *tests are create*

# ENTREGA

1. Incluyan en un archivo .zip los archivos correspondientes al laboratorio. El nombre debe ser los dos apellidos de los miembros del equipo ordenados alfabéticamente.
2. En el foro de entrega deben indicar el estado de avance de su laboratorio y los problemas pendientes por resolver.
3. Deben publicar el avance al final de la sesión y la versión definitiva en la fecha indicada, en los espacios preparados para tal fin.

# CONTEXTO

El objetivo es implementar el juego **Marbel Game** en versión usuario-computador.

El trabajo se debe hacer desde **CONSOLA**

|  |  |
| --- | --- |
| El propuesto por ustedes MarbelGameGUI | El acordado en laboratorio MarbelGame |
| **Vista - Controlador** | **Modelo** |

**Para la capa de presentación NO deben hacer pruebas de unidad ni diagramas de secuencia**

**DESARROLLO**

## Directorios

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

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

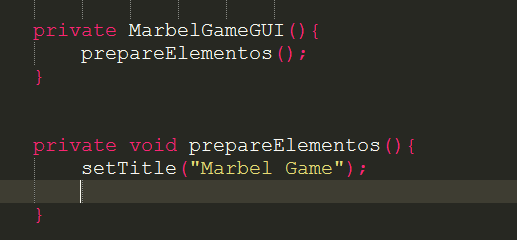
## Ciclo 0: Ventana vacía – Salir [En \*.java y lab05.doc]

**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-**

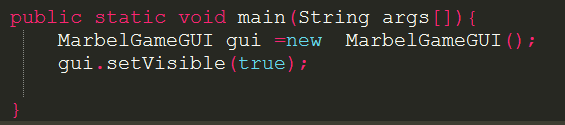
**prepareAcciones.**

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. Capturen la pantalla.**

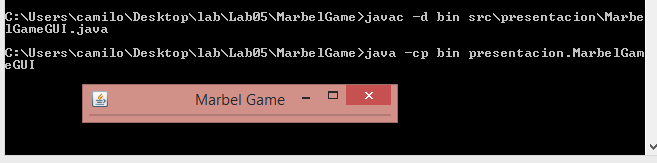
En el constructor



**En el main**

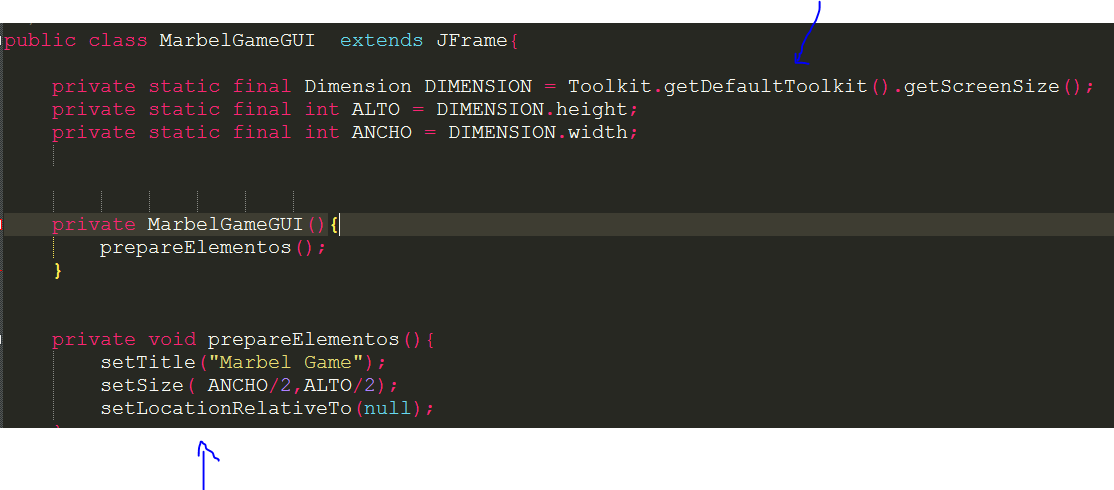


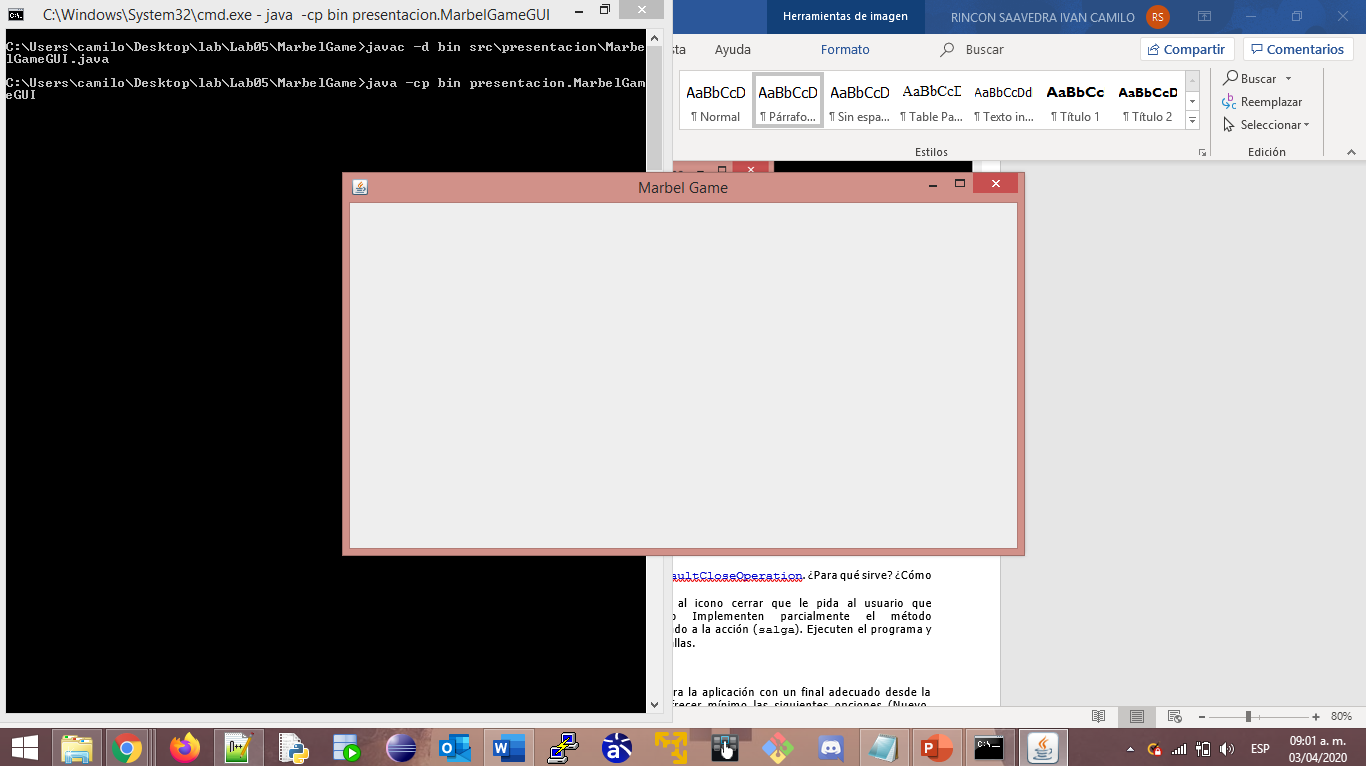
**Ejecutando**

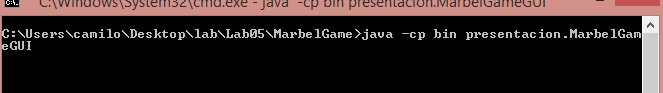


**(Si la ventana principal no es la inicial en su diseño, después moverá el main al componente visual correspondiente)**

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



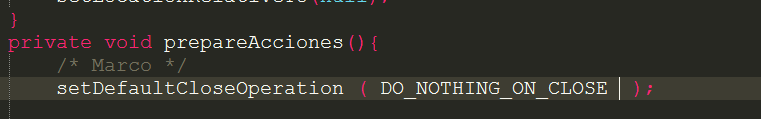


1. **Traten de cerrar la ventana. ¿Termina la ejecución? ¿Qué deben hacer para terminar la ejecución?** ¿Por qué? 

No termina la ejecucion como se logra ver en la imagen anterior, la unica forma es con la combinacion Control + c, porque los mas seguro es que al cerrar la ventana el JFrame queda con un metodo predeterminado de JFame **“JFrame.HIDE\_ON\_CLOSE: oculta el marco, pero mantiene la aplicación en ejecución”.**

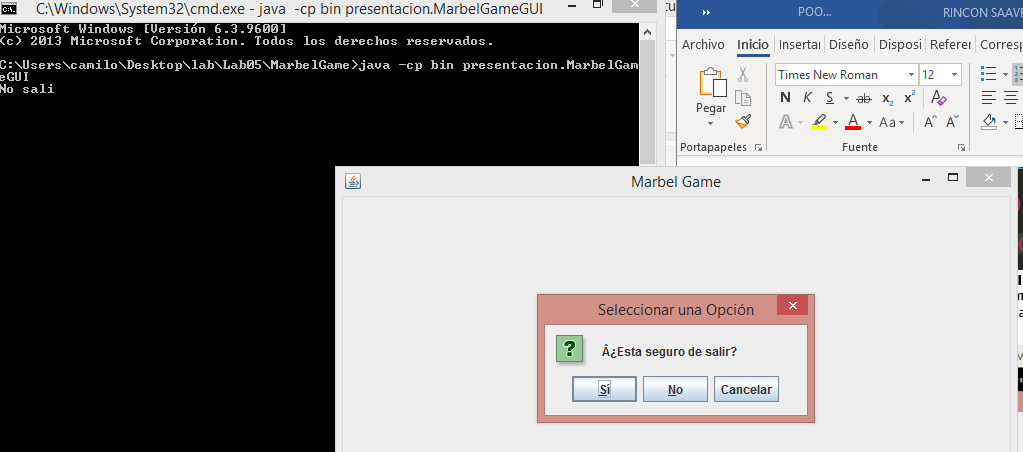
1. **Estudien en JFrame el método setDefaultCloseOperation. ¿Para qué sirve? ¿Cómo lo usarían en este caso?**

Este método sirve para especificar una de varias opciones para el botón de cierre, deberíamos modificar el método usando la constante: **“DO\_NOTHING\_ON\_CLOSE** “**Ignora el clic.”** y este debe ir en prepare acciones**.** Se elige este para que después podamos controlar si cerrar o no cuando se pregunte al usuario si desea salir.

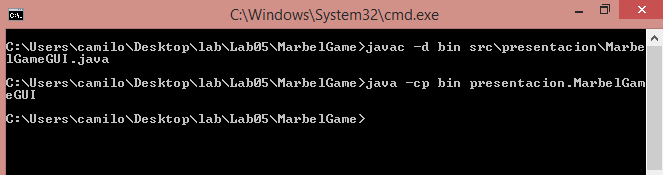


1. **Preparen el “oyente” correspondiente al icono cerrar que le pida al usuario que confirme su selección. Para esto Implementen parcialmente el método** prepareAcciones y el método asociado a la acción (salga). Ejecuten el programa y salgan del programa. Capturen las pantallas.





**Cuando le damos a si**



## Ciclo 1: Ventana con menú – Salir [En \*.java y lab05.doc]

**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**

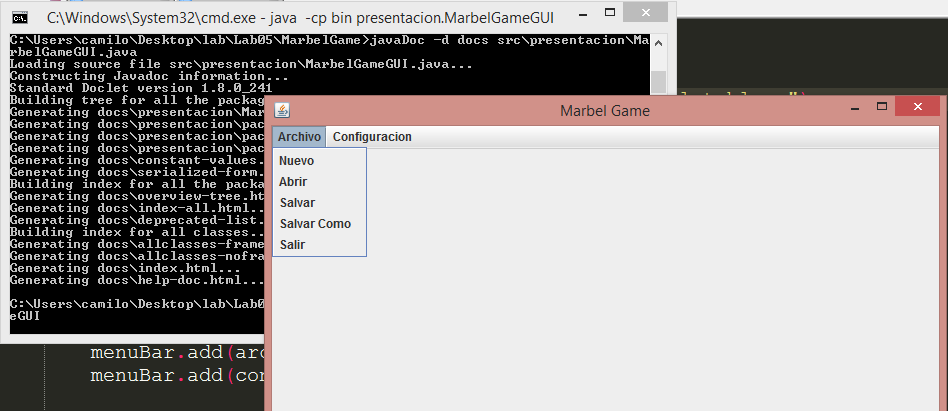
**) . No olviden los criterios de calidad del código.**

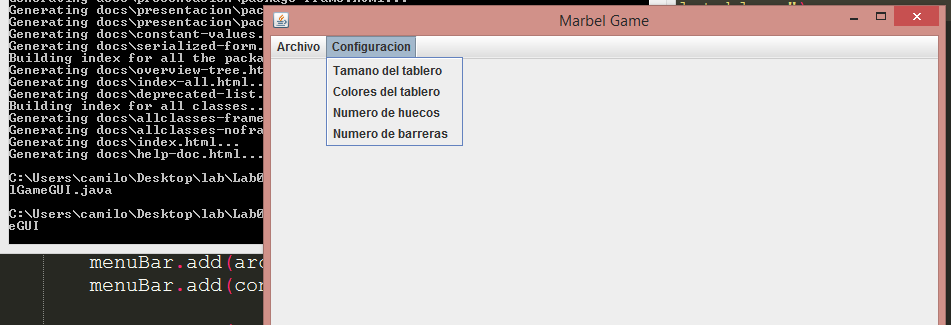
1. **Defina como atributos los componentes visuales necesarios del menú.**

Se deben definir los siguientes atributos con estos tipos:

* JMenuBar menuBar : variable que representa la barra de menú que vemos
* JMenu Archivo,Configuracion : variables que se añadirán a al menuBar ya que encerraran cada uno diferentes JMenuItem
* JMenuItem nuevo, abrir, salvar, salvarComo, salir, tamanoTablero, ColoresTablero, numeroHuecos, numeroBarreras: elementos del menú(JMenu) que al ser pulsado genera un evento, o sea, abre una ventana, pide datos o cualquier evento.

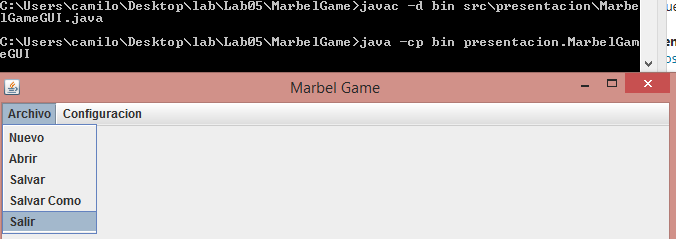
1. **Construya la forma del menú propuesto en su diseño de interfaz ( prepareElementos - prepareElementosMenu) . Ejecuten. Capturen la pantalla.**

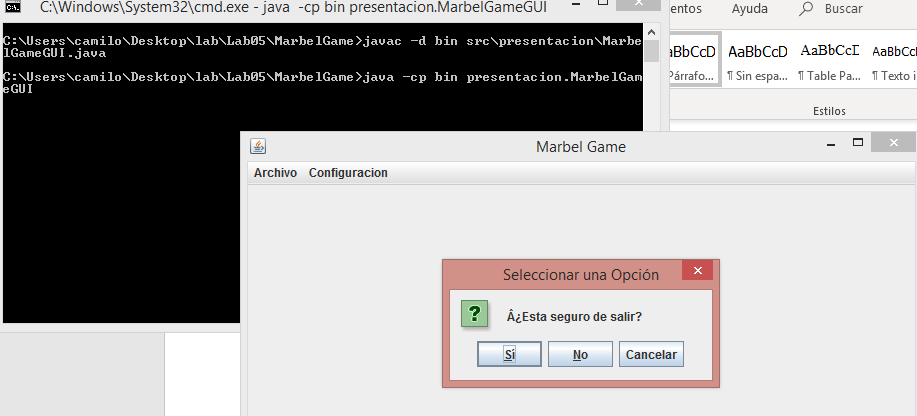


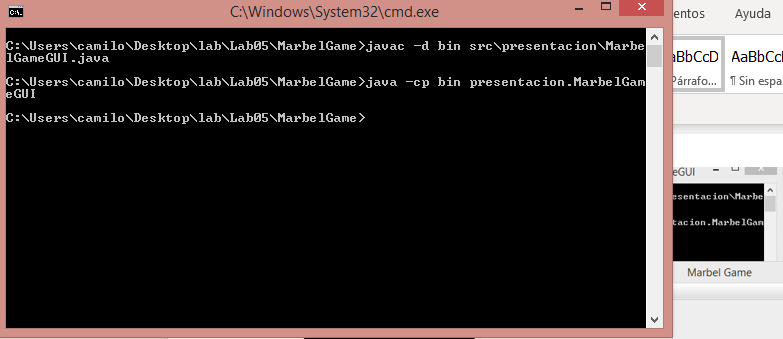


1. **Preparen el “oyente” correspondiente al icono cerrar con confirmación (prepareAcciones). Ejecuten el programa y salgan del programa. Capturen las pantallas.**

Para asignarles un listener a los menuItem se debe colocar nombreMenuItem.nombreListener y después seguir la sintaxis







## Ciclo 2: Salvar y abrir [En \*.java y lab05.doc]

**El objetivo es preparar la interfaz para las funciones de persistencia**

**Un objeto se dice persistente cuando es almacenado en un archivo u otro medio permanente. Un programa puede grabar objetos persistentes y luego recuperarlos en un tiempo posterior**

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

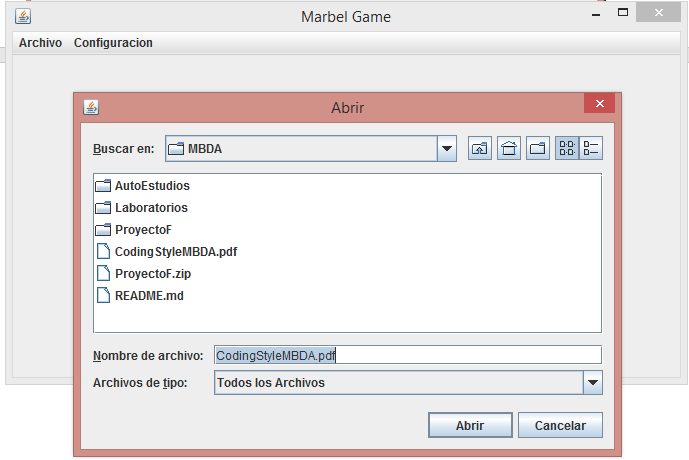
* **JFileChooser**:La Clase **JFileChooser**, nos presenta una ventana de herramientas, que, dependiendo de nuestra lógica, nos permitirá Abrir, Guardar archivos o cualquier otra acción, la ventana es una ventana de Dialogo para navegar por nuestros directorios.
* **getSelectedFile:** Cuando seleccionamos el archivo deseado, lo agregamos a un objeto de tipo **File**por medio del método **getSelectedFile**(), posteriormente recorremos el archivo y lo almacenamos en un String.es decir guarda la ruta donde se planea guardar el archivo.
* **showOpenDialog** se cargará la ventana con lo necesario para navegar por nuestros directorios y seleccionar el archivo que queremos abrir,
* **showSaveDialog:** se cargará una ventana para navegar por nuestros directorios y definir la ruta donde será almacenado. Esto se logra hacer gracias al método **showSaveDialog(this)** de la clase **JFileChooser**, que nos permite cargar la ventana de dialogo y encontrar la ruta donde almacenaremos el archivo.

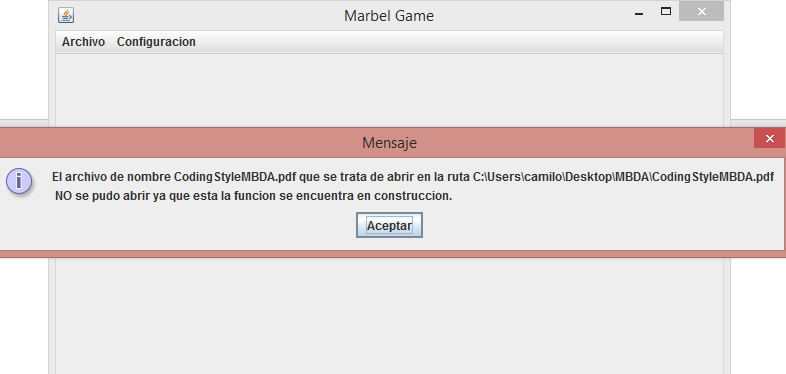
1. **Implementen parcialmente los elementos necesarios para salvar y abrir. Al seleccionar**

**los archivos indiquen que las funcionalidades están en construcción detallando la acción y el nombre del archivo seleccionado.**

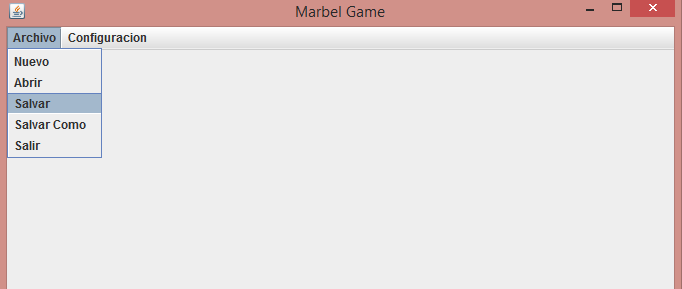
1. **Ejecuten las dos alternativas y capturen las pantallas más significativas.**

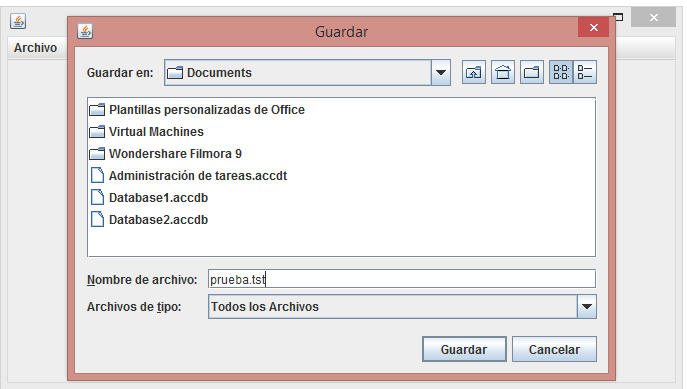
**Para Abrir**

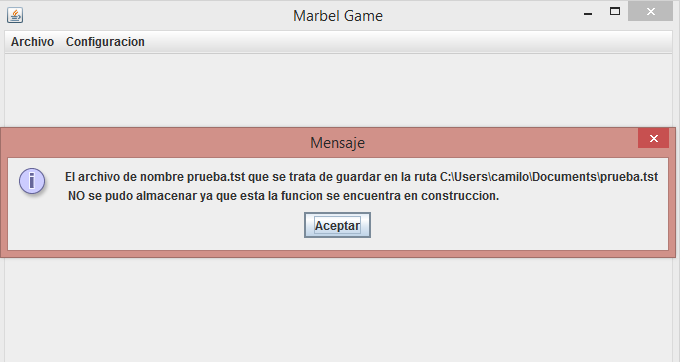




**Para Guardar**

****





## Ciclo 3: Forma de la ventana principal [En \*.java y lab05.doc]

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.**

Se usaron los siguientes componentes Graficos:

* JButton
* JPanel
* JMenuBar
* JMenu
* JMenuItem
* JTextField

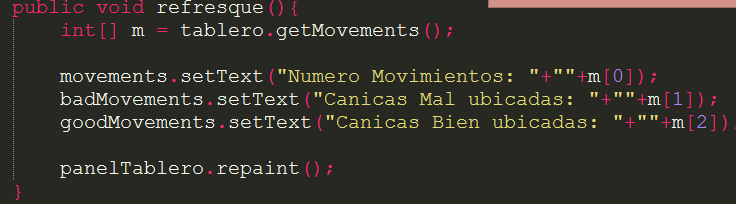
Variables implementadas por nosotros:

* **VentanaConfig:**Panel que contiene todo lo relacionado con la configuracion
* **MarbelGameIcono:** Clase que se encarga de poner una imagen sobre un componente **(Boton)**
* **PantallaJuego:** Frame el cual tiene todos los componentes necesarios el juego
* **PanelFondo:** Panel el cual pone imágenes de fondo
* **PanelTablero:** panel encargado de dibujar el tablero

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**



Dado que decidimos crear otro frame para la zona de juego este **prepareElementosTablero()**

equivale al **prepareElementos** de la clase **PantallaJuego**, el cual para pintar el tablero usa el Panel **PanelTablero, el refresque() se encuentra en la clase Pantalla Juego**

**por omisión (el ejemplo del trabajo en clase) Este método lo vamos a implementar realmente en otros ciclos. Ejecuten y capturen esta pantalla.**



## Ciclo 4: Cambiar color [En \*.java y lab05.doc]

**El objetivo es implementar este caso de uso.**

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

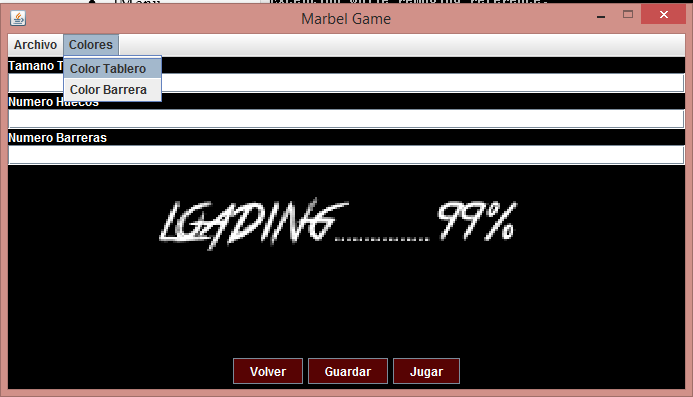
Es necesario tener dos variables tipo **color** las cuales representaran los colores que tomara el tablero y estas enviadas al constructor de una variable de tipo **Marbelgame( lógica del programa )**

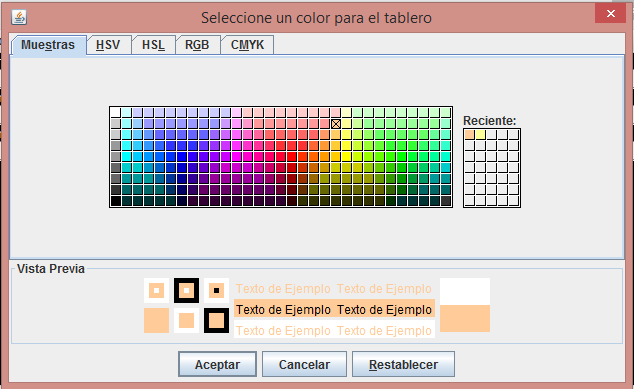
* 1. **Detalle el comportamiento de JColorChooser especialmente el método estático**

**showDialog**

* **JColorChooser:** Este componente nos ayuda a seleccionar un color de una forma más atractiva para el usuario, mostrando una nueva ventana que una gran variedad de colores, dicha ventana ya se encuentra implementada.
* **showDialog:** Muestra un cuadro de diálogo de selección de color modal y bloquea hasta que el cuadro de diálogo esté oculto. Si el usuario presiona el botón "Aceptar", este método oculta / elimina el cuadro de diálogo y devuelve el color seleccionado.
  1. **Implementen los componentes necesarios para cambiar el color del tablero (inicialmente blanco-negro)**

**Se debe crear las variables del siguiente Tipo:**

* JColorChooser
* JMenuBar
* JMenu
* JMenuItem
* Dos variables Color, las cuales representan los colores elegidos por el usuario
  1. **Ejecuten el caso de uso y capture las pantallas más significativas.**



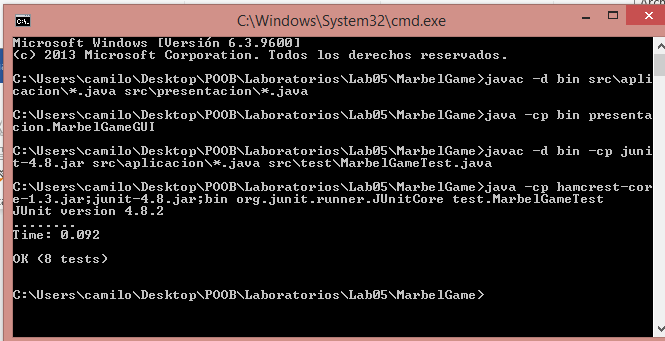


## Ciclo 5: Modelo MarbelGame [En \*.java y lab05.doc]

El objetivo es implementar la capa de aplicación para **MarbelGame** .

1. Construya los métodos básicos del juego (**No olvide MDD y TDD)**
2. Ejecuten las pruebas y capturen el resultado.

**Para compilar ejecutar javac -d bin -cp Junit…. Src\aplicacion\\*.java src\test\\*.java**



## 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 refresque() considerando la información del modelo de dominio.
3. Expliquen los elementos necesarios para implementar este caso de uso.

* **PantallaJuego:** Frame el cual tiene todos los componentes necesarios el juego
* **PanelFondo:** Panel el cual pone imágenes de fondo
* **PanelTablero:** panel encargado de dibujar el tablero
* JButton
* JPanel
* JMenuBar
* JMenu
* JMenuItem
* JTextField
* Usar Funciones de la Case Grafics

1. Implementen los componentes necesarios para jugar .
2. Ejecuten el caso de uso y capture las pantallas más significativas.



**Se da El botón Arriba**



**Se da El botón Derecha**



**Se da El botón Abajo**



**Se da El botón Izquierda**



## Ciclo 7: Reiniciar

**[En \*.java y lab05.doc]**

El objetivo es implementar este caso de uso.

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

* JMenu llamado configuración
* JMenuItem llamdo configuración inicial
* En MarbelGame Guardar una variable de Tipo **Elemento[][]** llamadatablero2

La cual nunca cambie, para que al dar configuracionInicial se ponga como tablero principal

1. Implementen los elementos necesarios para reiniciar
2. Ejecuten el caso de uso y capture las pantallas más significativas.

## 

## 

## 

## 

## 

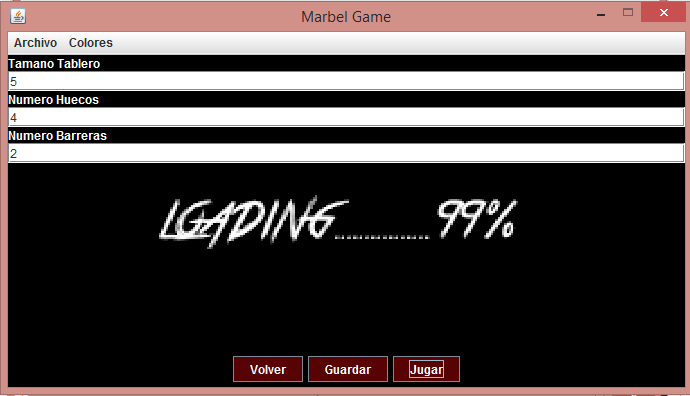
## Ciclo 8: Cambiar la configuración del juego [En \*.java y lab05.doc]

El objetivo es implementar este caso de uso.

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

* JTextFiel para leer las entradas ingresadas por el usuario
* JButton para guardar la configuración
* JPanel el cual encerrara los botones
* Atributos de tipo int que almacenen las entradas según corresponda

1. **Implementen los elementos necesarios para cambiar la configuración del juego: tamaño, número de canicas y número de barreras.**
2. **Ejecuten el caso de uso y capture las pantallas más significativas.**



**Guardar y luego Jugar**



**RETROSPECTIVA**

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

**(Horas/Hombre)**

Iván Camilo Rincón :20 horas.

Miguel Ángel Fúquene Arias: 20 horas.

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

Terminado, puesto que le invertimos el tiempo y la disposición requerida durante toda la semana.

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

La programación a pares, ya que con esto ambos integrantes del grupo aprenden y se practican los temas que se vieron durante la clase, adicionalmente también nos sirvió la práctica “Solo un par integra código a la vez” puesto que podemos hacerle mejoras o trabajarle al laboratorio desde cada uno de nuestros computadores teniendo en cuenta lo que el otro compañero ya había adicionado para no modificar u omitir lo que ya se había hecho.

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

Pudimos aplicar los conceptos vistos en clase satisfactoriamente puesto que pudimos realizar todas las funcionalidades y requerimientos que se pedían.

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

**resolverlo?**

ca

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

**resultados?**

Distribuimos bien el tiempo y utilizamos espacios de estudio fuera de clase para fortalecer los conocimientos y poder realizar todas las funcionalidades que proponía el laboratorio. Nos comprometemos a reforzar los temas vistos.