**ESCUELA COLOMBIANA DE INGENIERIA**

**PROGRAMACIÓN ORIENTADA A OBJETOS**

**INTERFAZ**

**S12: 2019-02**

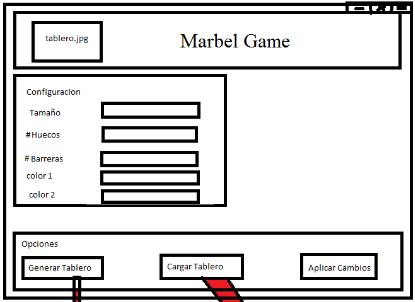
**Taller Marbel Game**

Miguel Angel Fuquene

Iván Camilo Rincón

**Diseño general en archivo .jng adjunto**

**Diseño (MVC)**



Los componentes gráficos son:

COMPONENTES SUPERIORES:

Es la ventana más grande de la aplicación y es de tipo JFrame

COMPONENTES INTERMEDIOS:

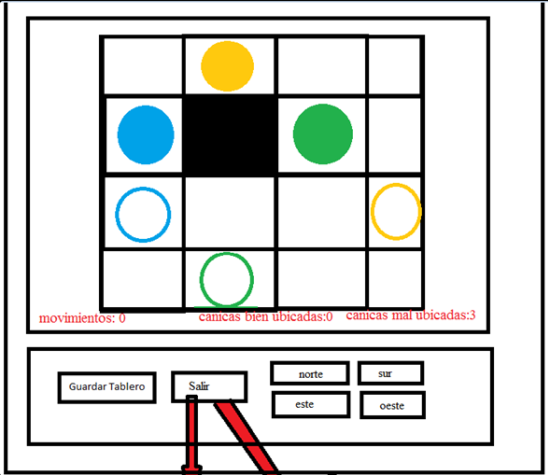
En este caso tenemos 3 contenedores tipo JPanel(El que encierra la imagen inicial, el de configuración y el de opciones).

COMPONENTES BASICOS:

-Tenemos 2 componente tipo JLabel en el Jpanel superior(tablero.jpg y Marbel Game) y su estilo es de un FlowLayout.

-Tenemos 6 componentes JLabel(configuraion,tamaño,#Huecos,#barrera,color1, color2) y 5 JTextField pertenecientes al JPanel del medio y tiene un estilo de GridLayout.

-Tenemos 3 JButton en el JPanel inferior (Generar Tablero, Cargar Tablero, Aplicar Cambios) y tiene un estilo de FlowLayout.



Los componentes gráficos son:

COMPONENTES SUPERIORES:

Un JFrame el cual se ve como contenedor de todos los componentes graficos.

COMPONENTES INTERMEDIOS:

• Para este caso tenemos un JPanel el cual agrupa todos los JButton

COMPONENTES BASICOS:

• Un conjunto de botones de tipo JButton que toman los nombres:

1. Guardar Tablero

2. Salir

3. Norte

4. Sur

5. Este

6. Oeste

• Este JPanel que empaqueta los botones está usando dos estilos, el primero un borderLayout el cual se encuentra usando las zonas Center and East y GridLayout:

1. Center: agrupa dos JButton

2. East:

Dentro de esta estar usando el segundo estilo GridLayout, el cual consta con una matriz de botones de 2X2(Norte,Sur,Este,Oeste).

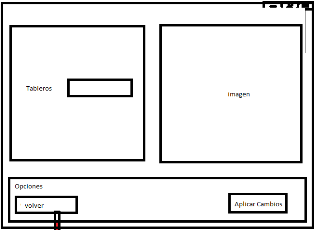
• Existe otro JPanel que empaqueta la imagen (El tablero) y un conjunto de JLabel:

Este JPanel toma dos estilos:

1. Un BorderLayout el cual toma las áreas Center y South:

 Center: Se encuentra una JLabel el cual representa el tablero

 South:Se encuentra un conjunto de texto de tipo JLabel, los cuales siguen el estilo de GridLayout de una matriz de 1X3



Los componentes graficos son:

COMPONENTES SUPERIORES:

La ventana principal o mas grande que seria un JFrame

COMPONENTES INTERMEDIOS:

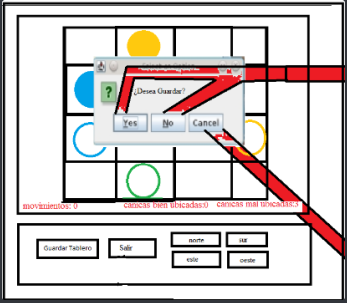
Tendria tres JPanel (izquierda,derecha e inferior) y tendría estilo de BorderLayout

COMPONENTES BASICOS:

Tenemos 1 componente tipo JLabel(Tableros) y 1 componente tipo JTextField , pertenecen al JPanel de la izquierda y tienen un estilo FlowLayout.

Tenemos 1 componente tipo JLaber(imagen) que pertenece al JPanel de la derecha.

Tenemos 2 componentes tipo JTextField(volver, Aplicar Cambios) que pertenece al JPanel inferior y están con un estilo FlowLayout.



Esta Ventana Posee Todos los componentes gráficos antes ya mencionados, pero adicionalmente posee un componente intermedio:

• JOptionPane\*: el cual nos dice si se desea guardar el tablero en el que se encuentra, cabe recalcar que este es un componente que ya implementado.

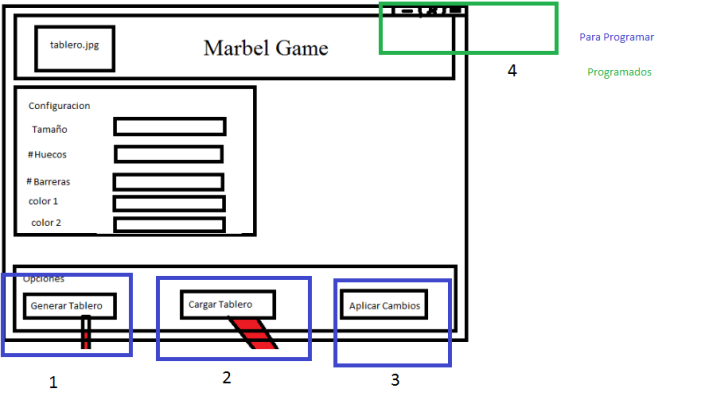
El cual posee su propio JFrame, con un estilo BorderLayoud usando las áreas:

1. North: el cual posee un texto el cual representa un JLabel

2. South: posee un conjunto de botones de tipo JButton que siguen el estilo GridLayout de una matriz de 1X3

3. West: un JLabel que representa una imagen de signo de interrogación.

**Diseño (MvC)**



1

**Evento:** Generar tablero

**Oyente**: ActionListener

**Metodo del Oyente:** actionPerformed(ActionEvent ev)

**Accion:** Se va a abrir otra ventana donde se mostrará el tablero predeterminado por **Marbel Game,** si no se modificó la configuración inicial de lo contrario lo creara con la configuración ingresada.

2

**Evento: -**CargarTablero

**Oyente:** ActionListener

**Metodo del Oyente:** actionPerformed(ActionEvent ev)

**Accion:** Se va a abrir otra ventana el cual mostrará un JTextField el cual se podrá escribir para buscar el nombre del tablero guardado

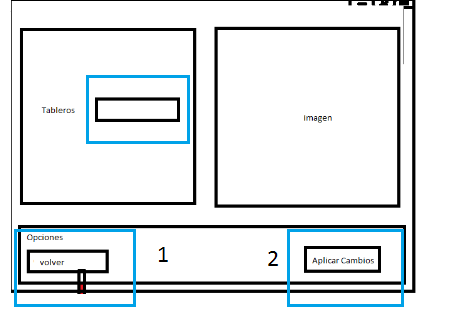
3

**Evento: -** Actualización de cambios

**Oyente:** ActionListener

**Metodo del Oyente:** actionPerformed(ActionEvent ev)

**Accion:** Realiza todos los cambios que se introdujeron en el JTexField y si no se introdujo nada deja los predeterminados

Para

1

**Evento**: Volver a la pestaña anterior

**Oyente:** ActionListener

**Metodo del Oyente:** actionPerformed(ActionEvent ev)

**Acción:** Vuelve a abrir la pestaña en la que se encontraba anteriormente

2

**Evento:** Si encontró el tablero

**Oyente:** ActionLisener

**Metodo del Oyente:** actionPerformed(ActionEvent ev)

**Accion:** Cargay Remplaza el tablero actual, por el que se introdujo en la Box

**Evento:** No encontró el tablero

**Oyente:** ActionLisener

**Metodo del Oyente:** actionPerformed(ActionEvent ev)

**Accion:** Le dice al usuario que no se encontró un tablero para cargar.

3

**Evento:** Encontrar el nombre

**Oyente:** DocumentListener

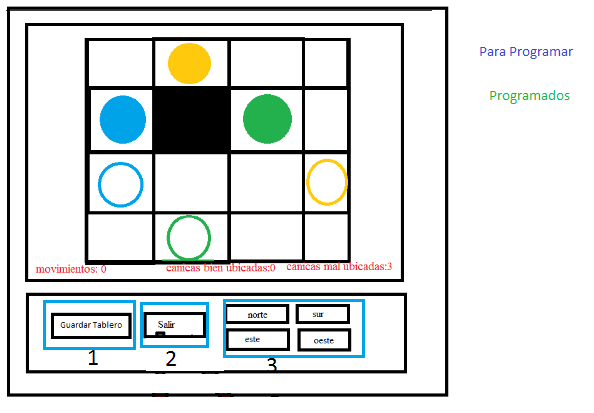
**Metodo del Oyente**:

changedUpdate(DocumentEvent ev),

insertUpdate(DocumentEvent ev)

removeUpdate(DocumentEvent ev)

**Accion:** Buscara dentro de todos los nombres de los archivos guardados, si coincide con el ingresado.



Para

1

**Evento:**Si esta Guardado

**Oyente:** ActionLisener

**Metodo del Oyente:** actionPerformed(ActionEvent ev)

**Accion:** deberá actualizar el archivo ya guardado

**Evento:** No Esta Guardado

**Oyente:** ActionLisener

**Metodo del Oyente:** actionPerformed(ActionEvent ev)

**Accion:** deberápreguntarle al usuario como guardar el tablero

2

**Evento:**Si esta Guardado

**Oyente:** ActionLisener

**Metodo del Oyente:** actionPerformed(ActionEvent ev)

**Accion:** debera volver al menú de inicio.

**Evento:** No Esta Guardado

**Oyente:** ActionLisener

**Metodo del Oyente:** actionPerformed(ActionEvent ev)

**Accion:** deberápreguntarle al usuario si desea guardar el tablero

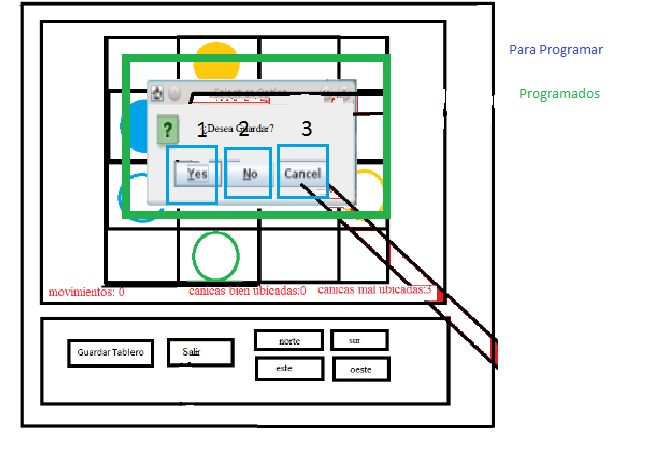
3

**Evento:** mover tablero

**Oyente:** ActionLisener

**Metodo del Oyente:** actionPerformed(ActionEvent ev)

**Accion:** deberá actualizarse el tablero, moviendo todas las canicas en la dirección especificada siempre y cuando al menos una ficha no este en su respectivo hueco

Para

1

**Evento: No esta Guardado el tablero actual**

**Oyente:** ActionLisener

**Metodo del Oyente:** actionPerformed(ActionEvent ev)

**Accion:** Debera guardar el tablero actual en un archivo y volver al menú de inicio

2

**Evento: No esta Guardado el tablero actual**

**Oyente:** ActionLisener

**Metodo del Oyente:** actionPerformed(ActionEvent ev)

**Accion:** No guarda el tablero y vuelve al menú de inicio.

3

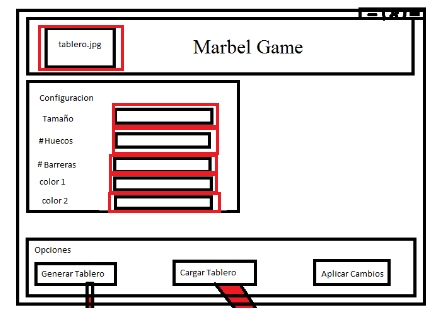
**Evento: No esta Guardado el tablero actual**

**Oyente:** ActionLisener

**Metodo del Oyente:** actionPerformed(ActionEvent ev)

**Accion:** No guarda el tablero y vuelve a la ventana en la que se encontraba

**Diseño (MVC)**



Se definirán las siguientes variables:

Variables de la clase tablero:

Private int tamaño

Private int huecos

Private int barreras

Private String color1

Private String color2

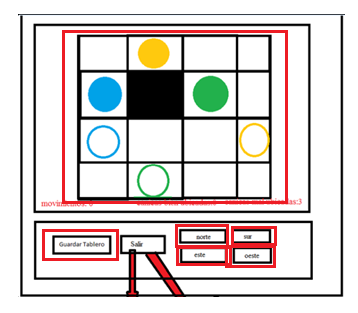
Variable de la clase marbelGame:

Private HashMap <String , String> = new HashMap<String , String>

Metodos de la clase marbelGame:

Public Tablero buscarTablero(String nombreTablero)

Public void actualizarParametros(int tamaño , int huecos, int barreras, String color1 , String color2)



Para esta imagen se definirán los siguientes atributos:

Private String nombreTablero =””

Private int numMovimientos = 0

Private int numCanicasBien = 0

Private int numCanicasMal = 0

Private int [][] tableroCanicos = new int [tamaño][tamaño]

Y se crearan los siguientes métodos y se guardaran en la clase tablero:

deslizarNorte()

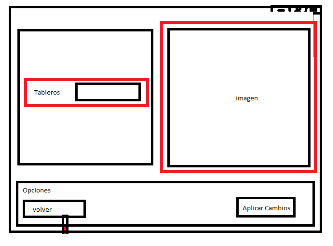
deslizarSur()

deslizarEste()

deslizarOeste()

Adicionalmente un metodo en la clase marbelGames

guardarTablero(String nombre)



Se crearan los siguientes métodos de la clase marbelGame:

Public String toString()