**PROGRAMACIÓN ORIENTADA A OBJETOS**

**INTERFAZ**

**2025-1**

**Laboratorio 5/6**

# **OBJETIVOS**

1. Desarrollar una mini aplicación gráfica considerando el patrón MVC.
2. Implementar el esquema de manejo de eventos con clases anónimas
3. Experimentar el comportamiento de las ventanas JFrame, JDialog y JOptionPane
4. Seleccionar los lienzos más apropiados para un diseño: JPanel y JTabbedPane
5. Revisar las posibilidades de los estilos:FlowLayout, BorderLayout y GridLayout
6. Apropiar algunos componentes básicos: JLabel, JTextField, JButton, JMenuBar,
7. Apropiar algunos componentes especiales: JFileChooser y JColorChooser
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. Deben publicar el avance al final de la sesión y la versión definitiva en la fecha indicada en los espacios correspondientes.

**CONTEXTO**

El objetivo es implementar el juego  **DMaxwell**

# El trabajo se debe hacer desde **CONSOLA**

|  |  |
| --- | --- |
| El propuesto por ustedes  DMaxwellGUI | El acordado en laboratorio  DMaxwell |
| **Vista - Controlador** | **Modelo** |

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

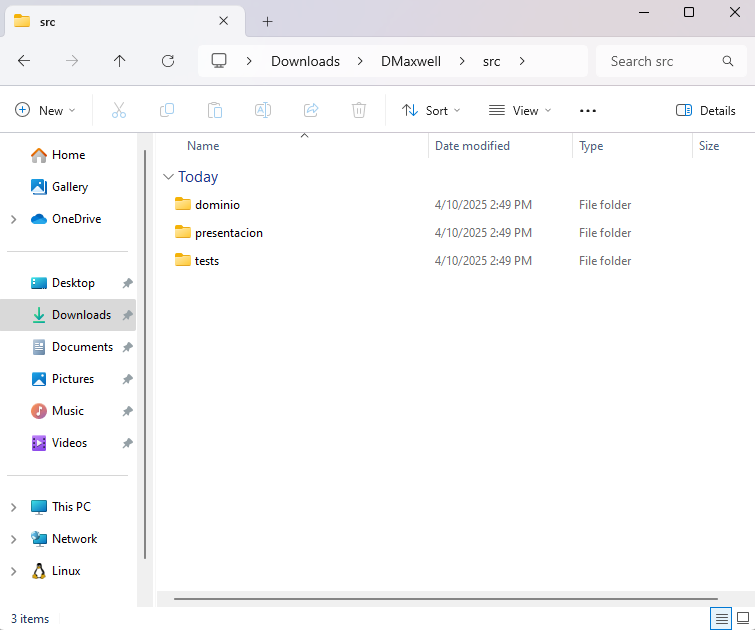
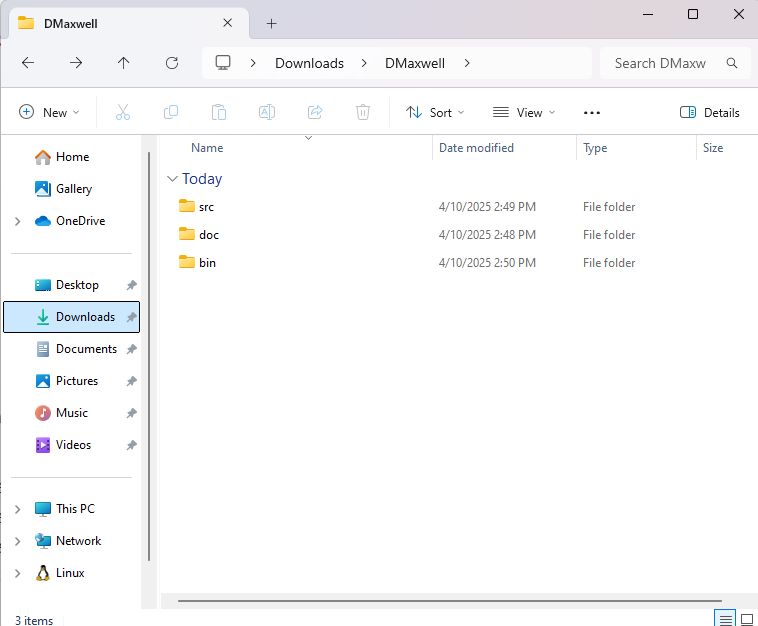
## DESARROLLO

### Directorios

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

3. 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.

* Se usa mkdir nombre\_al\_crear para esta sección.



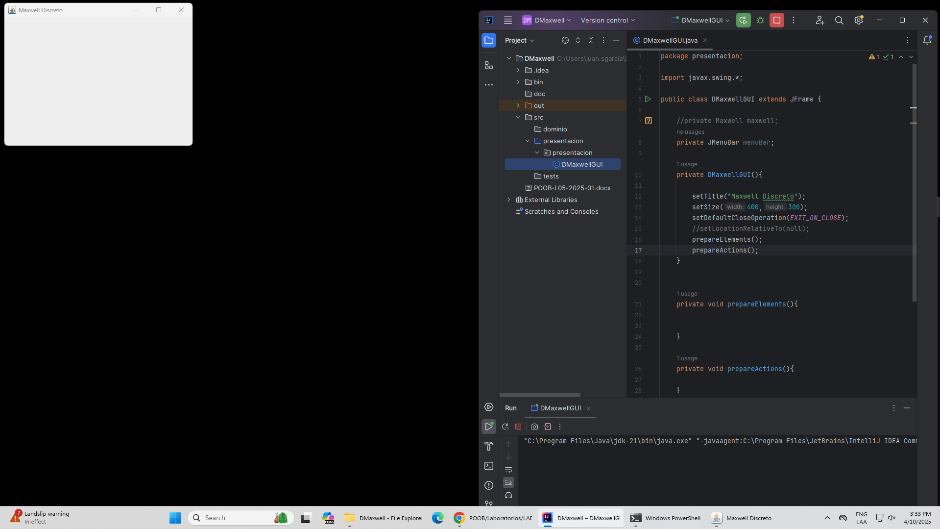
**Ciclo 0: Ventana vacía – Salir**

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

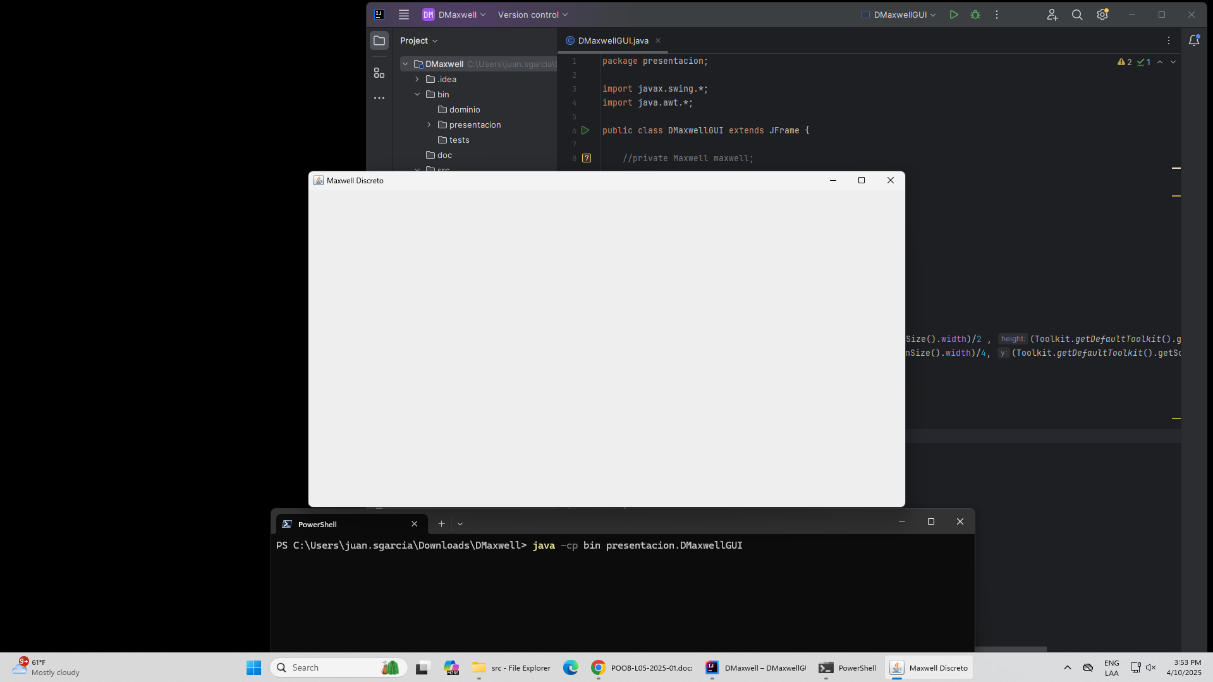
El objetivo es implementar la ventana principal de **DMaxwell** con un final adecuado desde el icono de cerrar. Utilizar el esquema de prepareElements-prepareActions.

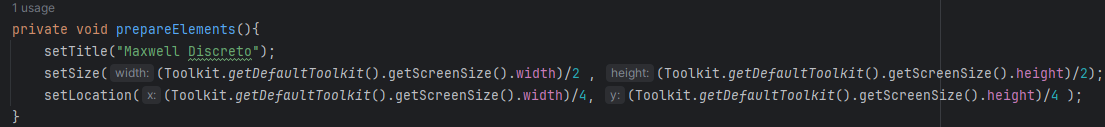
1. Construyan el primer esquema de la ventana de **DMaxwell** únicamente con el título “**Maxwell Discreto**”. Para esto cree la clase  **DMaxwellGUI** como un JFrame con su creador (que sólo coloca el título) y el método main que crea un objeto **DMaxwellGUI** y lo hace visible. Ejecútenlo. Capturen la pantalla.

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



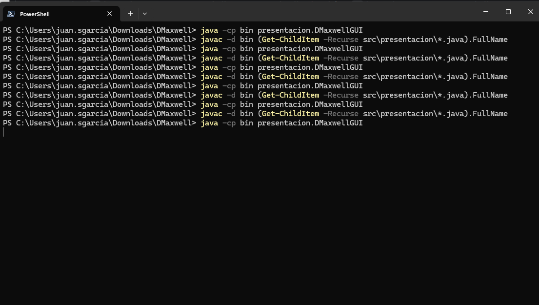
1. Modifiquen el tamaño de la ventana para que ocupe un cuarto de la pantalla y ubíquenla en el centro. Para eso inicien la codificación del método prepareElements. Capturen esa pantalla.





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

* Se cierra la ventana sin embargo, la ejecución sigue, y esto se ve por la consola:



* La forma de que se termine la ejecución por consola es usando el comando de ctrl + c en Windows.

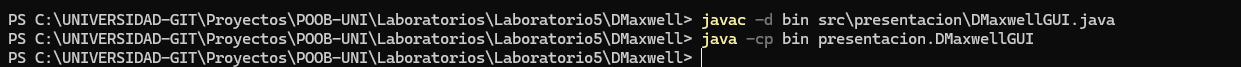
1. Estudien en JFrame el método setDefaultCloseOperation. ¿Para qué sirve? ¿Cómo lo usarían si queremos confirmar el cierre de la aplicación? ¿Cómo lo usarían si queremos simplemente cerrar la aplicación?

* Establece la operación que se realizará de forma predeterminada cuando el usuario inicie un "cierre" en este marco. Debe especificar una de las siguientes opciones:
  + DO\_NOTHING\_ON\_CLOSE (definido en WindowConstants): No hagas nada; requiere que el programa maneje la operación en el windowClosing método de un WindowListener objeto registrado .
  + HIDE\_ON\_CLOSE (definido en WindowConstants): Oculta automáticamente el marco después de invocar cualquier WindowListener objeto registrado.
  + DISPOSE\_ON\_CLOSE (definido en WindowConstants): Ocultar y eliminar automáticamente el marco después de invocar cualquier WindowListener objeto registrado.
  + EXIT\_ON\_CLOSE (definido en JFrame): Salga de la aplicación usando el System exit método. Use esto solo en aplicaciones.

1. **Interfaz de usuario gráfica, Texto

   El contenido generado por IA puede ser incorrecto.**Texto

   El contenido generado por IA puede ser incorrecto.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.



**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 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.

* Para implementar el menú de la aplicación se requieren varios componentes de Swing. El principal es JMenuBar, que actúa como contenedor del menú completo. Dentro de él, se utiliza un JMenu llamado "Archivo" que agrupa las opciones del menú. Las opciones individuales como "Nuevo", "Abrir", "Salvar" y "Salir" se implementan con JMenuItem. También se utiliza JSeparator para dividir visualmente las opciones, lo cual mejora la legibilidad. JMenuBar y JMenu deben declararse como atributos de la clase, ya que forman parte estable de la interfaz gráfica y podrían necesitar ser manipulados desde otros métodos. Los JMenuItem pueden ser atributos si se van a usar en métodos como prepareActionsMenu, pero podrían ser variables locales si solo se usan en la construcción inicial.
* Además de esto añadimos la opción de configuración para más adelante.

1. Interfaz de usuario gráfica, Aplicación, Word

   El contenido generado por IA puede ser incorrecto.Texto

   El contenido generado por IA puede ser incorrecto.Construya la forma del menú propuesto (prepareElements - prepareElementsMenu) . Ejecuten. Capturen la pantalla.

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

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

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Aplicación, Word

El contenido generado por IA puede ser incorrecto.

**Ciclo 2: Salvar y abrir**

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

El objetivo es preparar la interfaz para las funciones de persistencia

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

* El componente JFileChooser es una herramienta proporcionada por Swing que permite a los usuarios seleccionar archivos o directorios a través de un cuadro de diálogo. Es muy útil para implementar funcionalidades de "Abrir" y "Guardar" en aplicaciones Java.
* **JFileChooser()**: Este es el constructor básico del JFileChooser, que crea un cuadro de diálogo vacío de selección de archivo. Cuando se llama a este constructor, el cuadro de diálogo se configura para trabajar con archivos de manera predeterminada.
* **showOpenDialog(Component parent)**: Este método muestra el cuadro de diálogo para abrir un archivo. Recibe un componente padre (parent), que usualmente es la ventana principal. El cuadro de diálogo permite al usuario seleccionar un archivo para abrir. Este método devuelve un valor entero que indica la acción realizada:
  + JFileChooser.APPROVE\_OPTION si el usuario selecciona un archivo y hace clic en "Abrir".
  + JFileChooser.CANCEL\_OPTION si el usuario cancela la operación.

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.

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

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

Interfaz de usuario gráfica, Texto, Aplicación, Word

El contenido generado por IA puede ser incorrecto.**Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.**

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

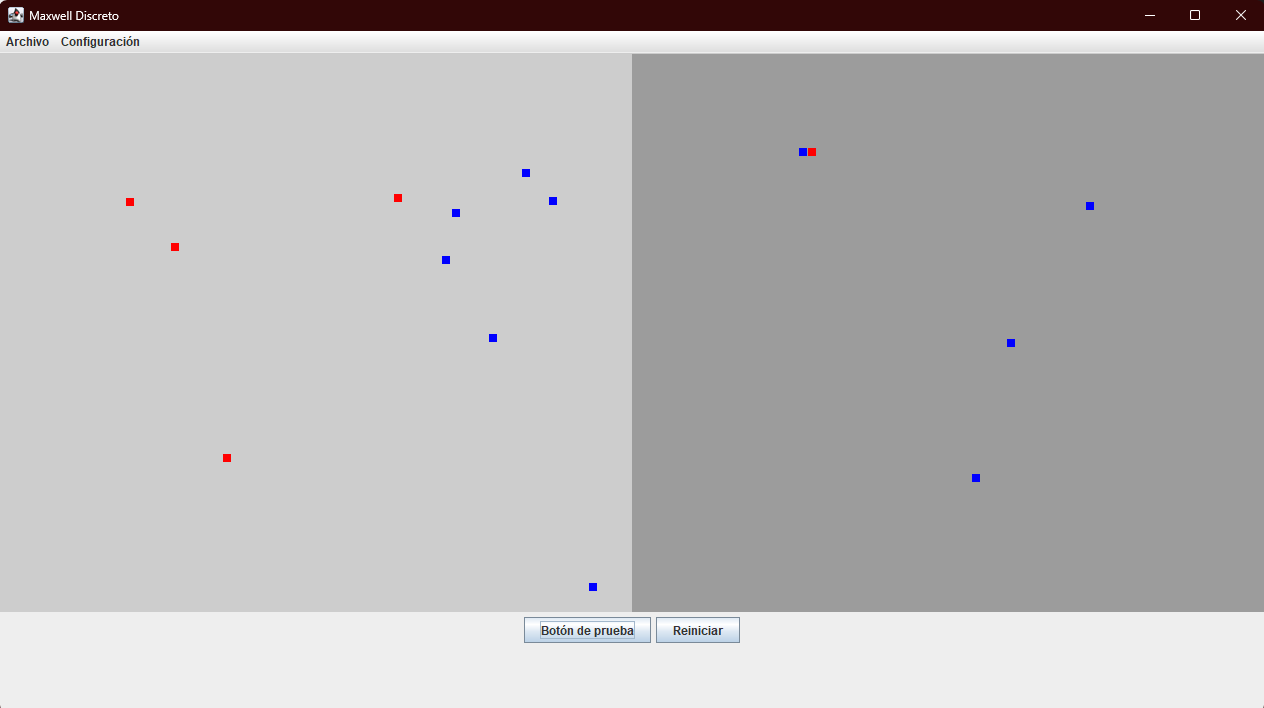
Gráfico, Gráfico de dispersión

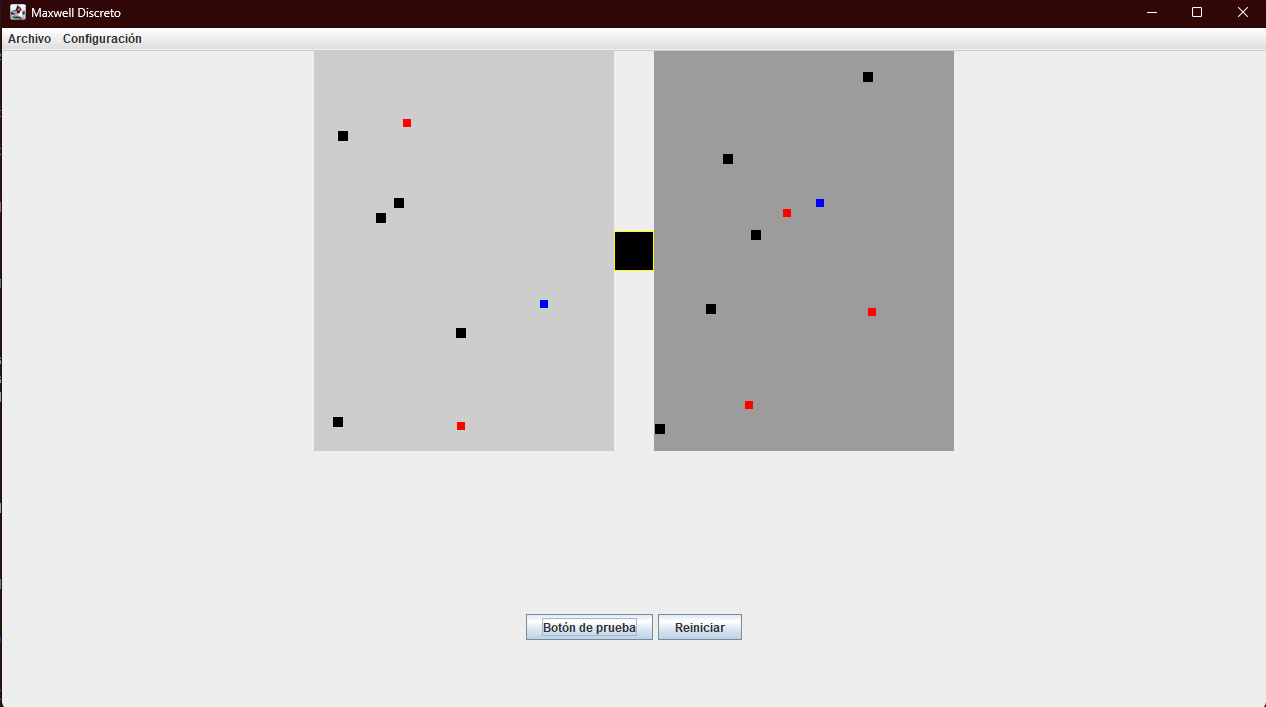
El contenido generado por IA puede ser incorrecto.

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.

1. Ejecuten y capturen la pantalla.





**Ciclo 4: Cambiar colores**

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

El objetivo es implementar este caso de uso.

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

En vista necesitamos los componentes JMENUITEM, JDIALOG y JCOLORCHOOSER

En controlador necesitamos los métodos mostrarDialogoColores(), actualizarColorParticulas(colorViejo, colorNuevo), cambiarColorFondoPaneles(colorNuevo)

También se usaron eventos de ActionListener Y JColorChooser.showDialog()

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

El método tiene la siguientes estructura

public static Color showDialog(Component parent, String title, Color initialColor)

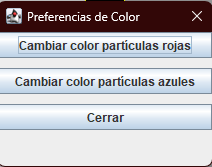
como parametros tiene parent que es la componente padre para centrar el diálogo (puede ser null)

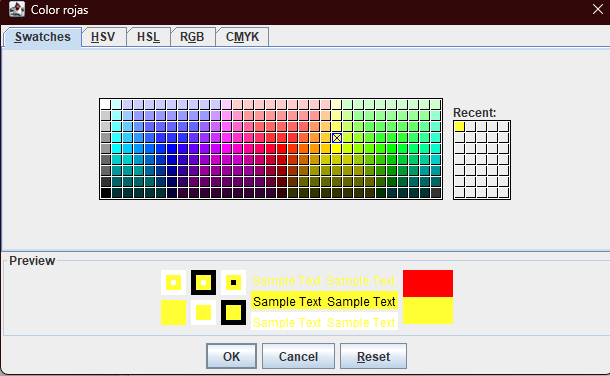
también tiene el parámetro title que es el que uno desee para la ventana y initialColor que es de tipo Color en el que se establece el color de entrada o inicial. Finalmente retorna un color que es el seleccionado o null si la operación se cancela.

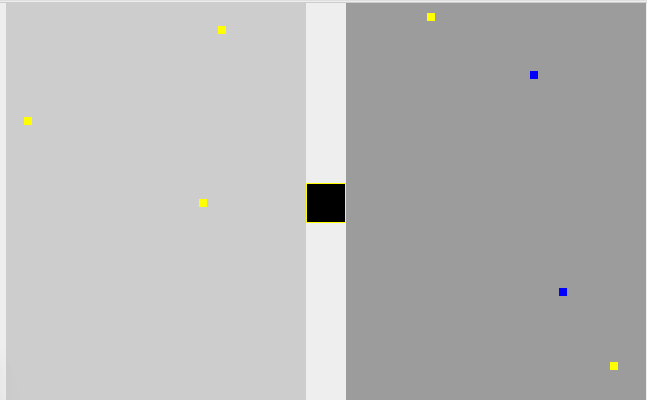
1. Implementen los componentes necesarios para cambiar el color de las fichas.

ok implementado

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







**Ciclo 5: Modelo DMaxwell**

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

El objetivo es implementar la capa de dominio para **DMaxwell**

1. Construya los métodos básicos del juego (**No olvide MDD y BDD)**
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.
4. Implementen los componentes necesarios para jugar .¿Cuántos oyentes necesitan?

¿Por qué?

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

**[BONO] 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.

* Para implementar la funcionalidad de reinicio en la aplicación Maxwell Discreto, se utilizaron los siguientes elementos:
  + 

1. Implementen los elementos necesarios para reiniciar

Texto

El contenido generado por IA puede ser incorrecto.

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

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.Gráfico, Gráfico de dispersión

El contenido generado por IA puede ser incorrecto.

**[BONO] Ciclo 8: Cambiar el tamaño**

### [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

* Cambiamos lo siguiente en si:



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

* 

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

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

RETROSPECTIVA

1. ¿Cuál fue el tiempo total invertido en el laboratorio por cada uno de ustedes? (Horas/Hombre)  
   • 18 horas – Sánchez  
   • 18 horas – Ruiz
2. ¿Cuál es el estado actual del laboratorio? ¿Por qué?  
   El laboratorio está completamente terminado. Consideramos que hemos hecho un buen trabajo, cumpliendo con todos los objetivos establecidos.
3. Seleccionen una práctica XP del laboratorio. ¿Por qué consideran que es importante?  
   Elegimos la práctica en la que, al detectar un bug, se crean pruebas. Esta práctica fue crucial porque nos permitió aprender más sobre pruebas unitarias y, al mismo tiempo, mejoró la calidad de nuestro código.
4. ¿Cuál consideran fue el mayor logro? ¿Por qué? ¿Cuál consideran que fue el mayor problema? ¿Qué hicieron para resolverlo?  
   Nuestro mayor logro fue completar el laboratorio con éxito y comprender todos los procesos involucrados. El desafío más grande fue desarrollar la interfaz gráfica de usuario (GUI) desde cero, ya que es un tema complejo y requiere un dominio preciso de los comandos. Para resolverlo, buscamos ejemplos y documentación que nos ayudaron a comprender mejor el proceso y las herramientas necesarias.
5. ¿Qué hicieron bien como equipo? ¿Qué se comprometen a hacer para mejorar los resultados?  
   Consideramos que trabajamos de manera eficiente en muchas áreas, pero sentimos que la comunicación podría haberse mejorado. Nos comprometemos a mantener una comunicación más constante y clara en futuras actividades para optimizar nuestros resultados.
6. ¿Qué referencias usaron? ¿Cuál fue la más útil? Incluyan citas con estándares adecuados.  
   Utilizamos varias fuentes de referencia para guiarnos en el desarrollo del laboratorio. La más útil fue la documentación oficial de la API de Java, que nos proporcionó información precisa y clara sobre las herramientas que utilizamos.

Cita en formato APA:  
Oracle. (n.d.). *Java Platform, Standard Edition 21 API Specification*. Oracle. Recuperado de <https://docs.oracle.com/en/java/javase/21/>