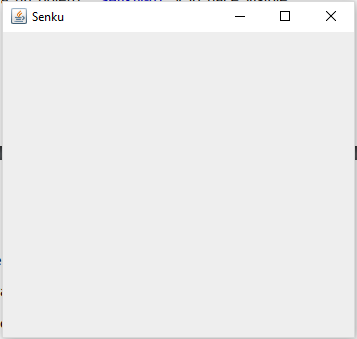
**DESARROLLO**

**Ciclo 0: Ventana vacía – Salir**

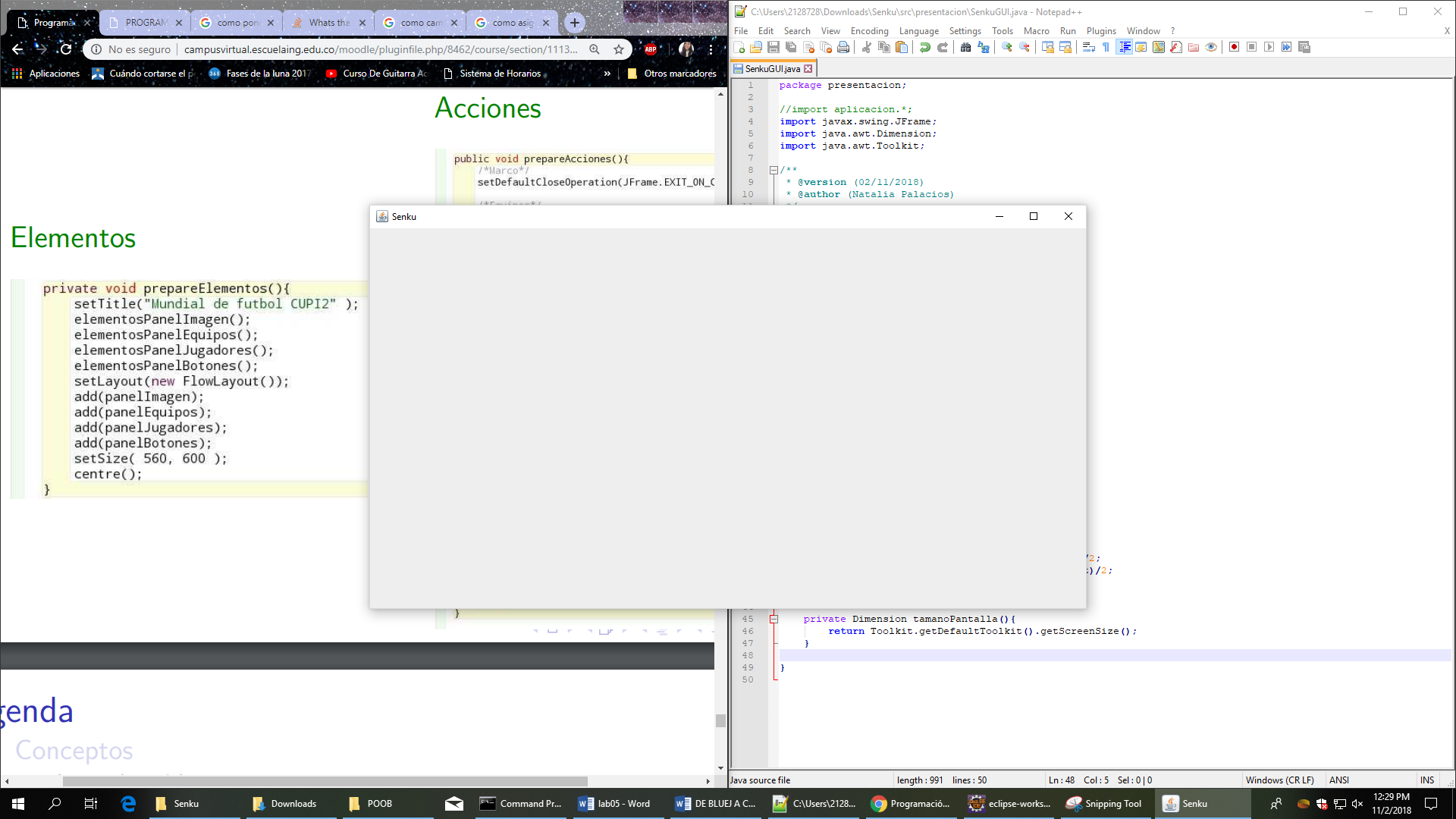
El objetivo es implementar la ventana principal de Senku con un final adecuado a la aplicación desde el icono de cerrar. Utilizar el esquema de prepareElementos - prepareAcciones.

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

(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?

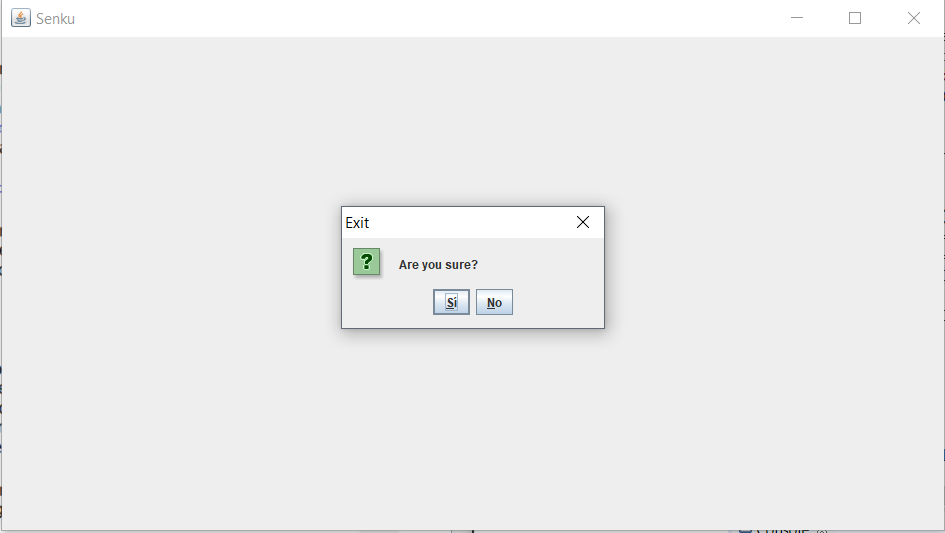
Se cierra la ventana pero el proceso continúa, se debe terminar el proceso desde el método prepareAcciones ya que no ha terminado como tal la ejecución.

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

Se utiliza para terminar todo proceso. En este caso lo utilizaría en el método prepareAcciones para cuando quiera cerrar la ventana.

**this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);**

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.



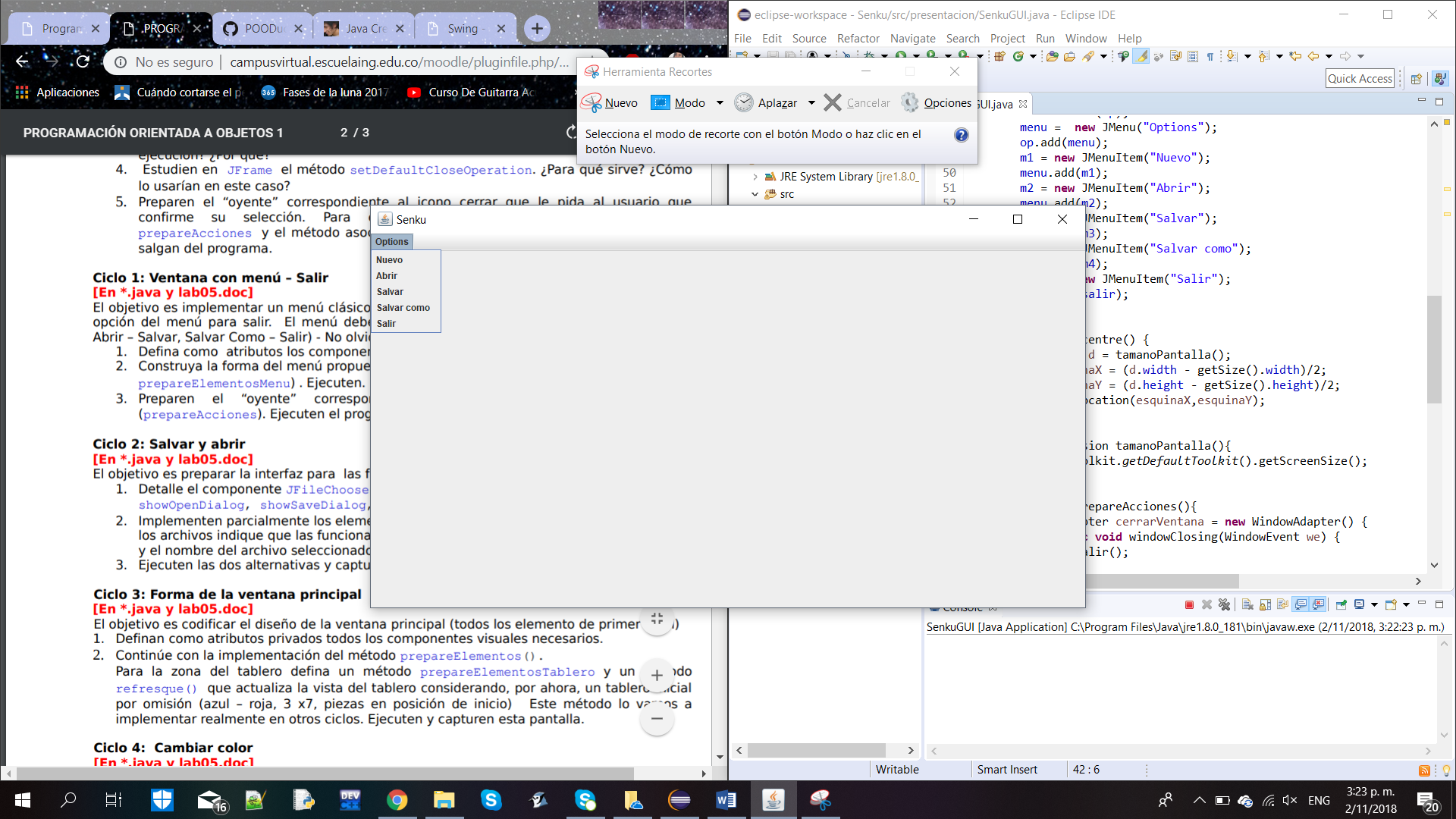
**Ciclo 1: Ventana con menú – Salir**

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.

Se definieron JMenuBar, JMenu y JMenuItem para los atributos.

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



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

**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.

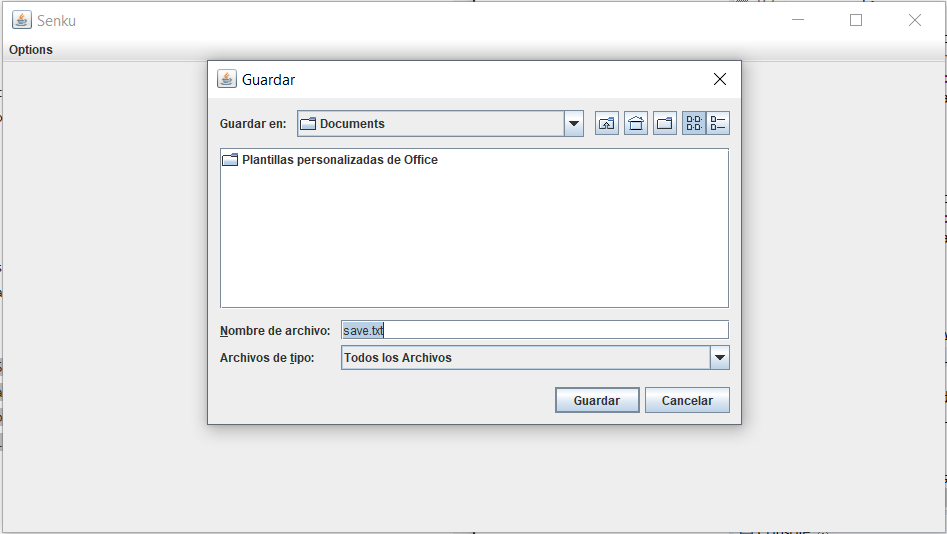
JFileChooser: Provee un mecanismo simple para el usuario para escoger un archivo.

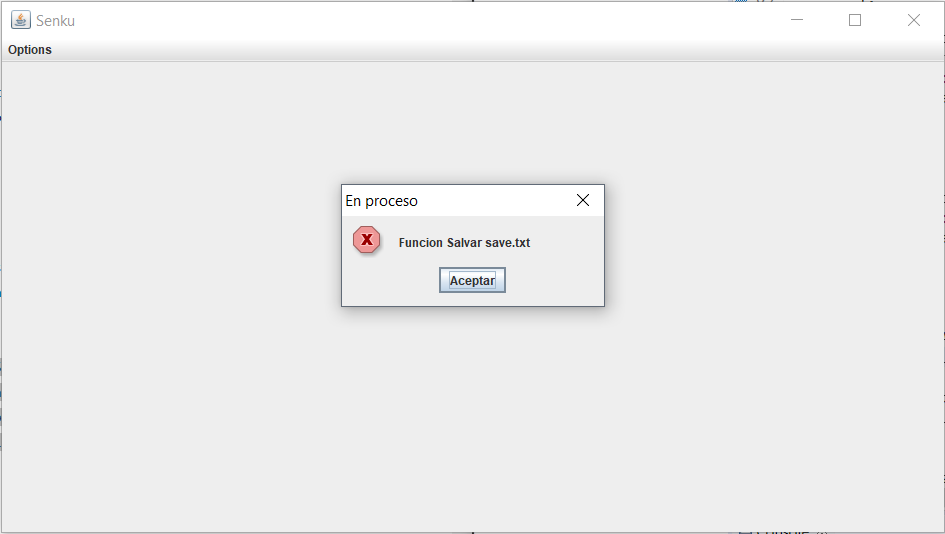
showOpenDialog: Abre una ventana para abrir un archivo.

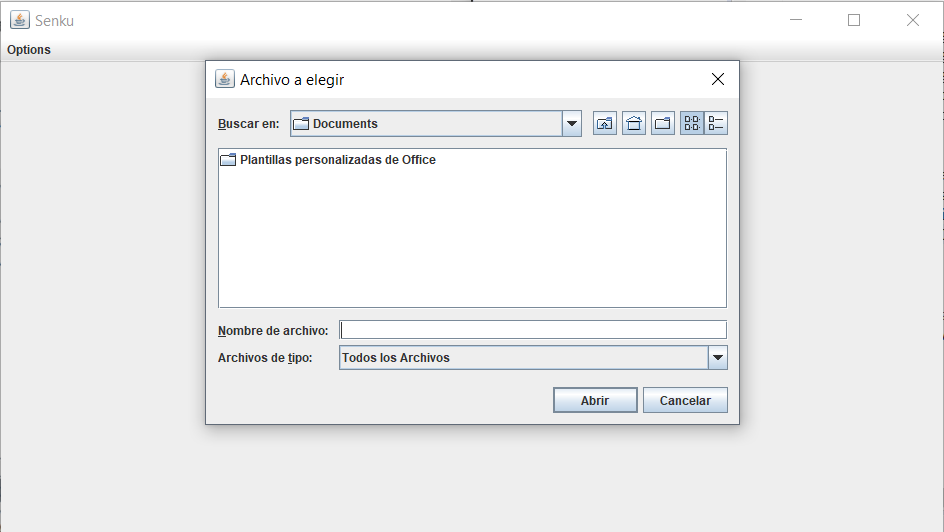
showSaveDialog: Abre una ventana para guardar un archivo.

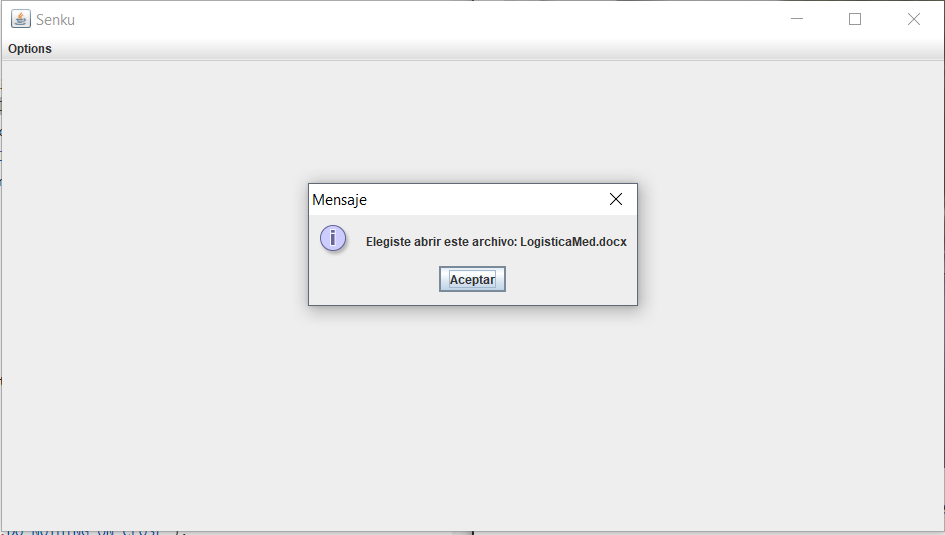
getSelectedFile: Agarra el archivo elegido.

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.
2. Ejecuten las dos alternativas y capture las pantallas más significativas.







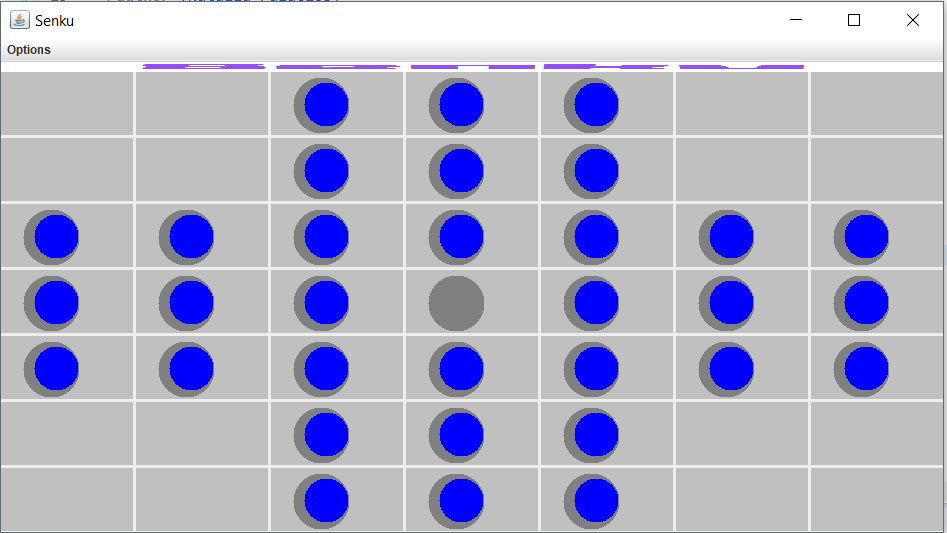


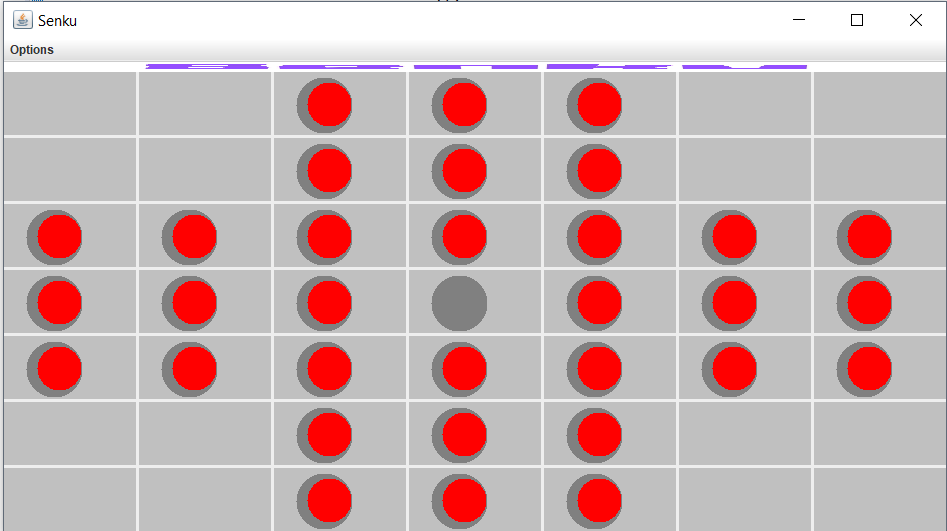
**Ciclo 3: Forma de la ventana principal**

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

1. Definan como atributos privados todos los componentes visuales necesarios.
2. 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 (azul – roja, 3 x7, piezas en posición de inicio) Este método lo vamos a implementar realmente en otros ciclos. Ejecuten y capturen esta pantalla.





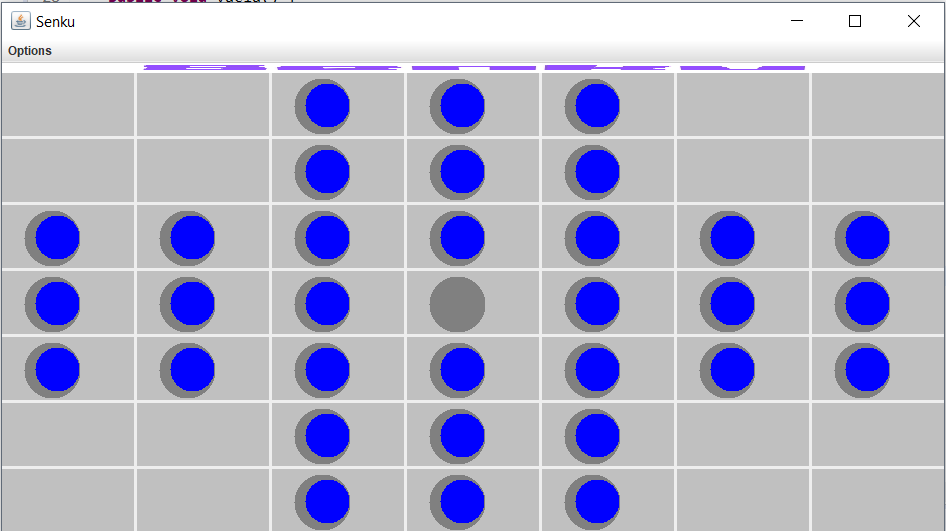
**Ciclo 4: Cambiar color**

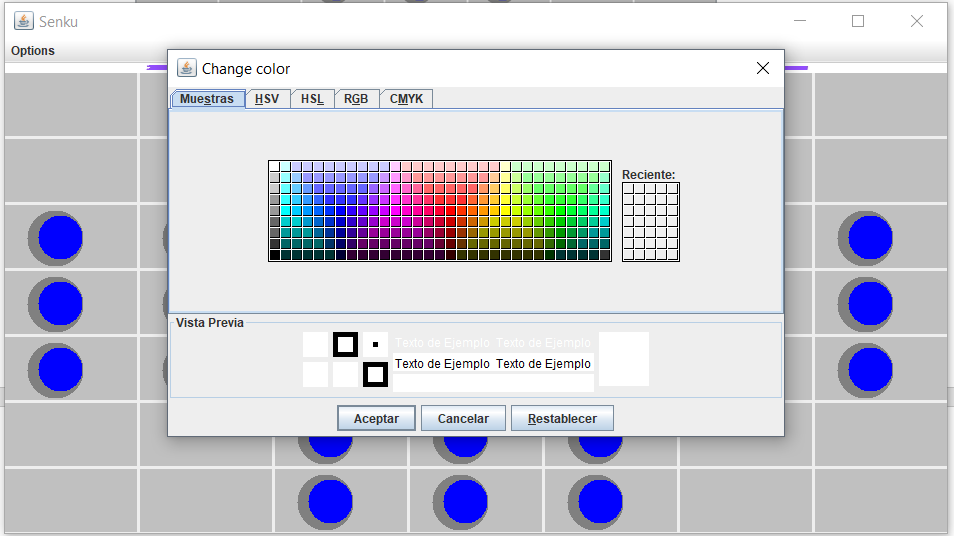
El objetivo es implementar este caso de uso.

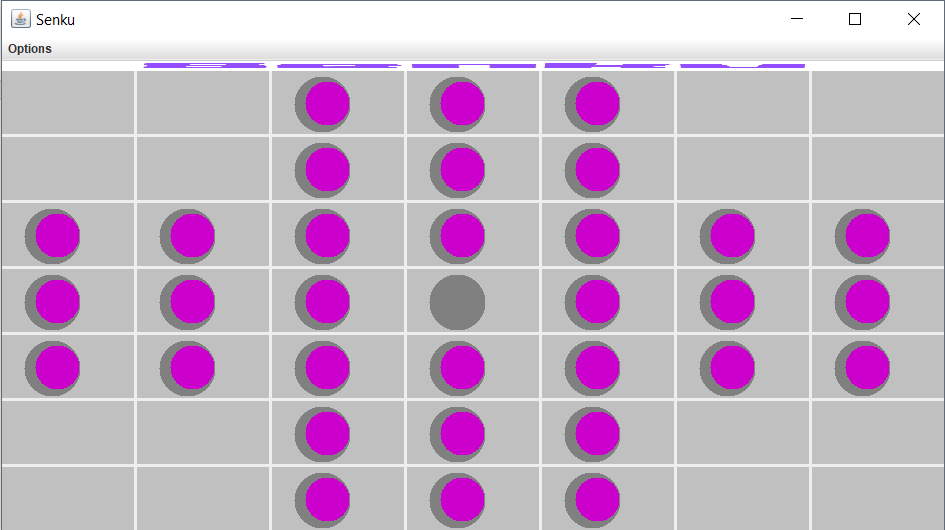
1. Expliquen los elementos necesarios para implementar este caso de uso.

Se usa JColorChooser, se agregan nuevos métodos de cambiar color en la clase Casilla y en la clase SenkuGUI, se crean atributos de tipo Color en ambas clases. La opción de cambiar color se encuentra en Options (menú)

1. Detalle el comportamiento de JColorChooser especialmente el método estático showDialog.
2. Implementen los componentes necesarios para cambiar el color de las canicas.
3. Ejecuten el caso de uso y capture las pantallas más significativas.







**Ciclo 5: Modelo Senku**

El objetivo es implementar la capa de aplicación para Senku.

1. Construya los métodos básicos del juego **(No olvide MDD y TDD)**
2. 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.
2. Perfeccionen el método refresque() 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.
5. 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.
2. Implementen los elementos necesarios para reiniciar.
3. . Ejecuten el caso de uso y capture las pantallas más significativas.

**Ciclo 8: Cambiar el tamaño del juego**

El objetivo es implementar este caso de uso.

1. Expliquen los elementos a usar para implementar este caso de uso.
2. Implementen los elementos necesarios para cambiar el tamaño del juego.
3. Ejecuten el caso de uso y capture las pantallas más significativas.

**RETROSPECTIVA**

1. ¿Cuál fue el tiempo total invertido en el laboratorio por cada uno de ustedes? (Horas/Hombre)
2. ¿Cuál es el estado actual del laboratorio? ¿Por qué?
3. Considerando la práctica XP del laboratorio ¿por qué consideran que es importante?
4. ¿Cuál consideran fue su mayor logro? ¿Por qué? ¿Cuál consideran que fue su mayor problema? ¿Qué hicieron para resolverlo?
5. ¿Qué hicieron bien como equipo? ¿Qué se comprometen a hacer para mejorar los resultados?