**Daniel Gomez**

**Juan Camilo Ortiz**

**Ciclo 0: Ventana vacía – Salir**

[En \*.java y lab05.doc]

El objetivo es implementar la ventana principal de Replicate con un final adecuado a la

aplicación desde el icono de cerrar. Utilizar el esquema de prepareElementosprepareAcciones.

1. Construyan el primer esquema de la ventana de Replicate únicamente con el título

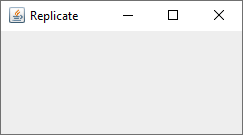
“ Replicate”. Para esto cree la clase ReplicateGUI como un JFrame con su

creador, que sólo coloca el título, y el método main que crea un objeto

ReplicateGUI y lo hace visible. Ejecútenlo. Capturen la pantalla.

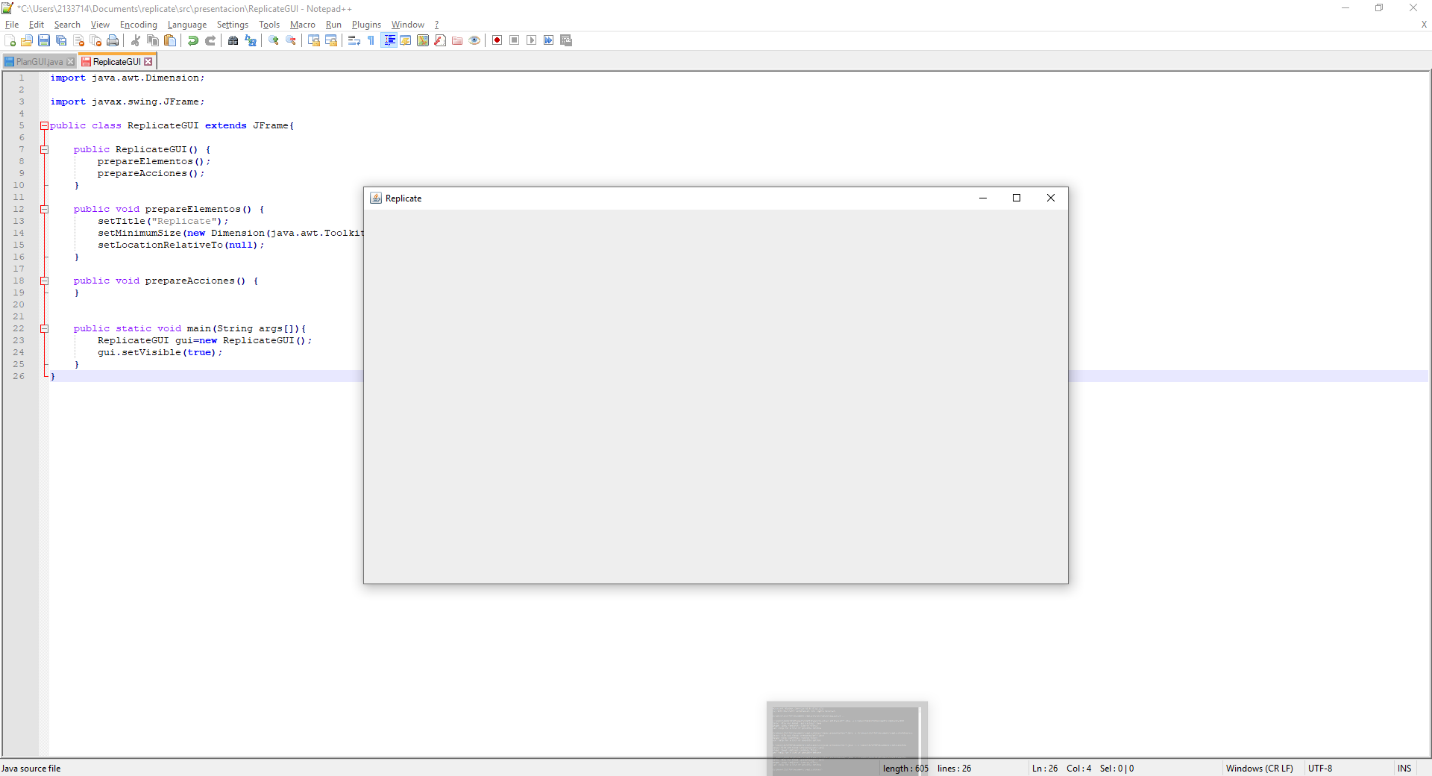
(Si la ventana principal no es la inicial en su diseño, después moverá el main al

componente visual correspondiente)



2. Modifiquen el tamaño de la ventana para que ocupe un cuarto de la pantalla y

ubíquenla en el centro (prepareElementos). Capturen esa pantalla.



3. Traten de cerrar la ventana. ¿Termina la ejecución? ¿Qué deben hacer para terminar

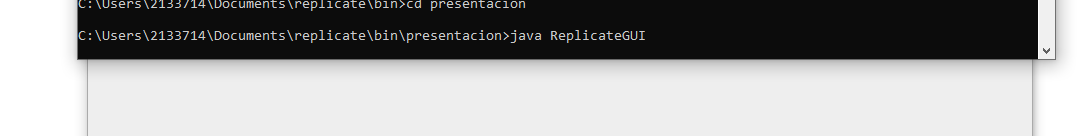
la ejecución? ¿Por qué?



No termina la ejecucion al cerrar el Jframe , por lo que toca implementar el metodo que lo detiene.

4. Estudien en JFrame el método setDefaultCloseOperation. ¿Para qué sirve? ¿Cómo

lo usarían en este caso?





se utiliza para especificar una de varias opciones para el botón de cierre. Use una de las siguientes constantes para especificar su elección:

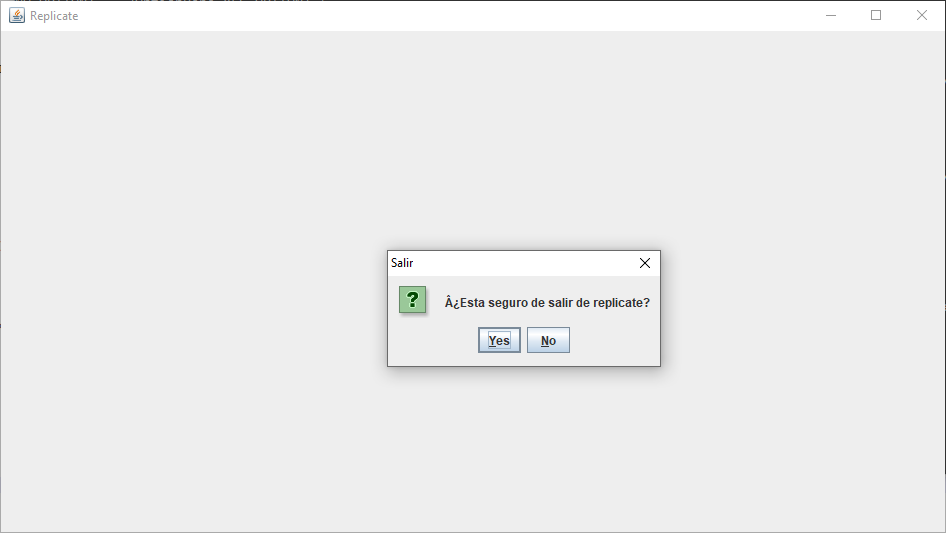
* JFrame.EXIT\_ON\_CLOSE: sale de la aplicación.
* JFrame.HIDE\_ON\_CLOSE: oculta el marco, pero mantiene la aplicación en ejecución.
* JFrame.DISPOSE\_ON\_CLOSE: elimine el objeto de marco, pero mantenga la aplicación en ejecución.
* JFrame.DO\_NOTHING\_ON\_CLOSE - Ignora el clic.

5. 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**

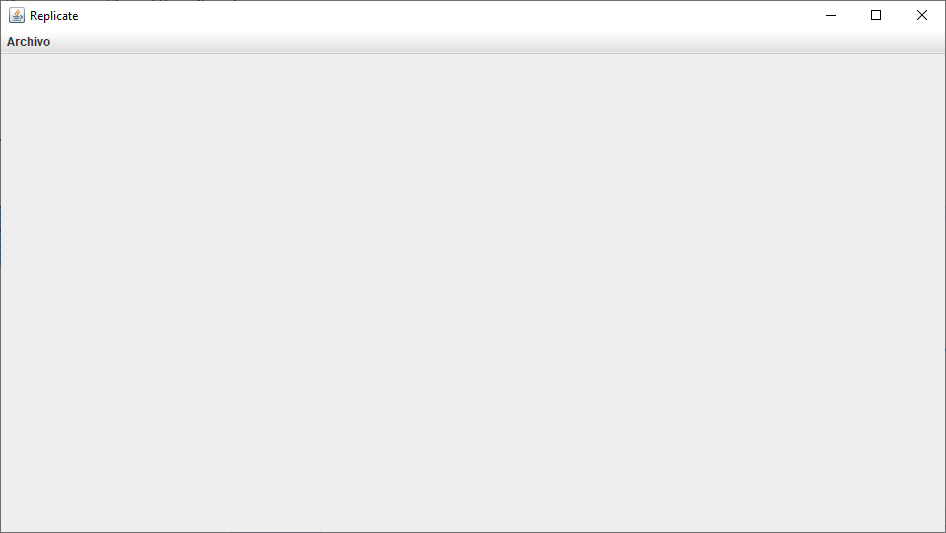
[En \*.java y lab05.doc]

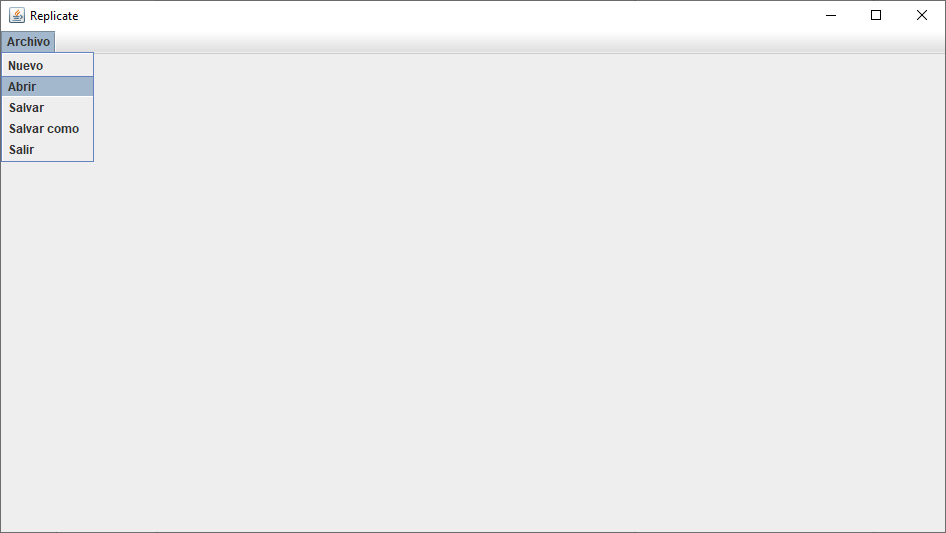
El objetivo es implementar un menú clásico para la aplicación con un final adecuado desde la opción del menú para salir. El menú debe ofrecer mínimo las siguientes opciones (Nuevo, Abrir – Salvar, Salvar Como – Salir) - No olviden los criterios de calidad del código.

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

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

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





**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.

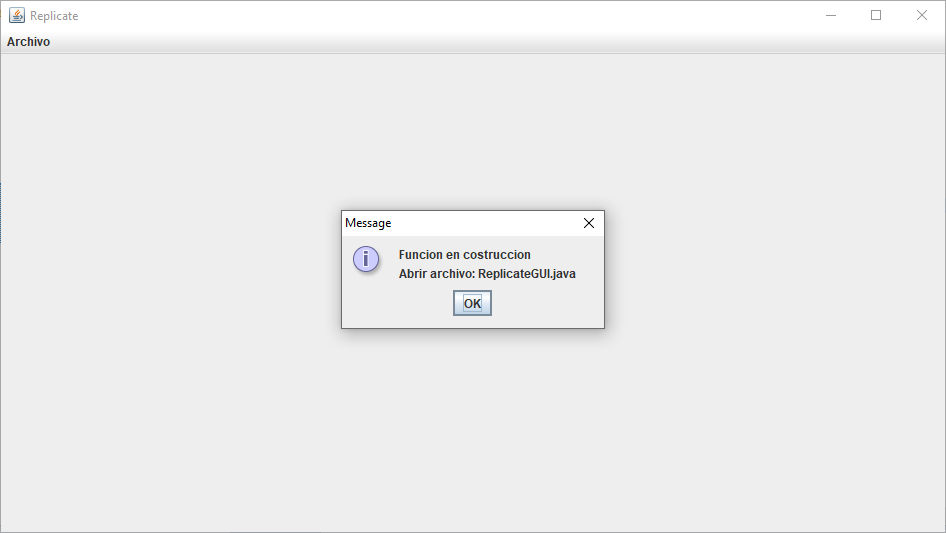
2. Implementen parcialmente los elementos necesarios para salvar y abrir. Al

seleccionar los archivos indique que las funcionalidades están en construcción

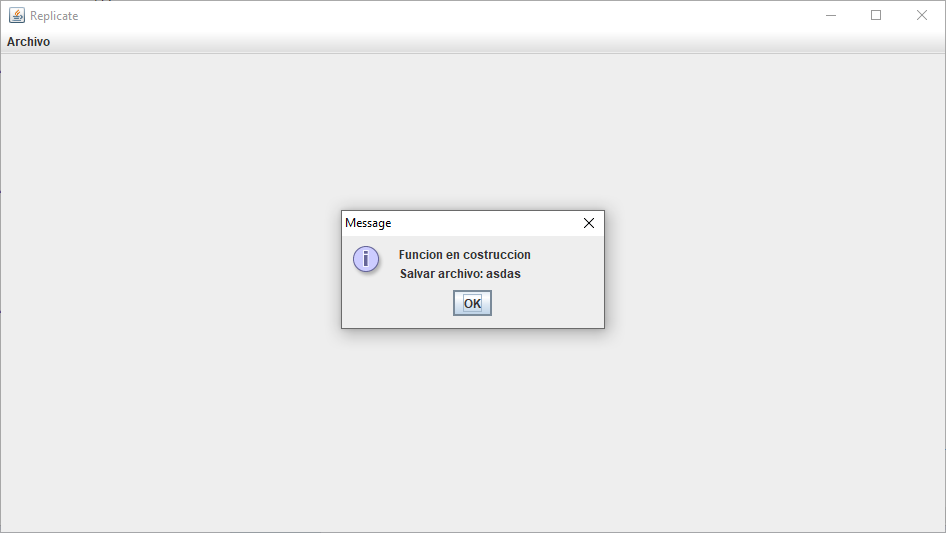
detallando la acción y el nombre del archivo seleccionado.

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

Salvar:



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.

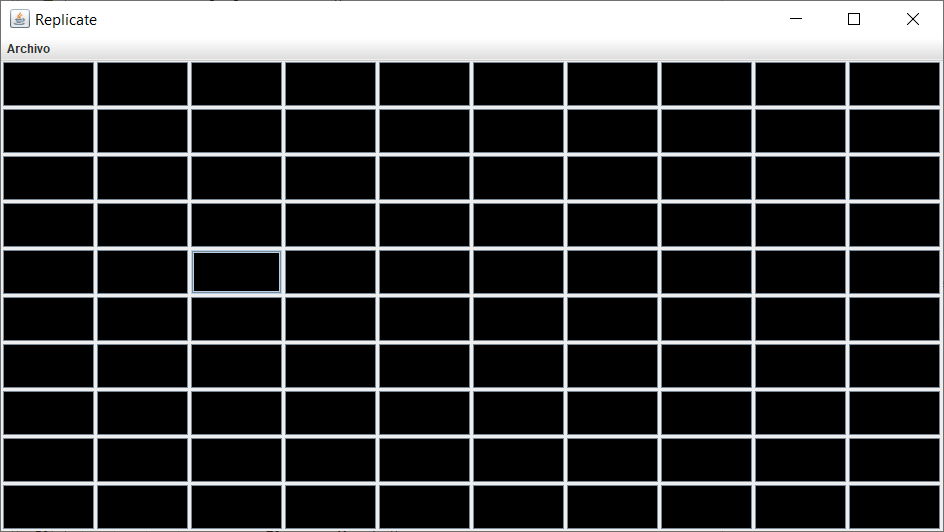
2. Continúe con la implementación del método prepareElementos().

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 (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 necesario para implementar este caso de uso.

BotonColor

JColorChooser

showDialog

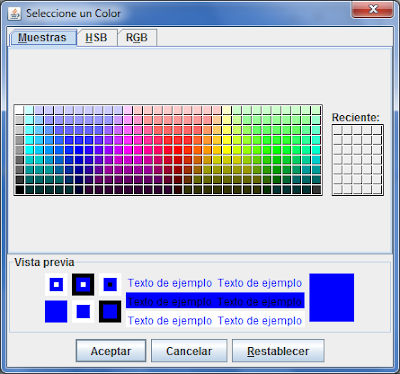
variables para capturar el color.

2. Detalle el comportamiento de JColorChooser especialmente el método estático

showDialog

JColorChooser : Este componente nos ayuda a seleccionar un color de una forma mas atractiva para el usuario.

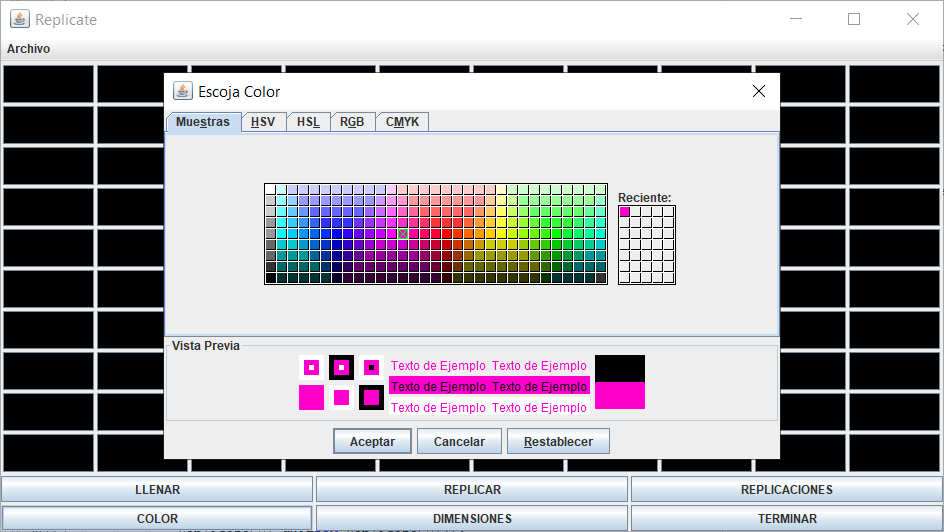
showDialog : Abre la paleta de colores y se selecciona el color que se quiere usar.



3. Implementen los componentes necesarios para cambiar el color de las celdas llenas y

vacías.

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





**Ciclo 5: Modelo Replicate**

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

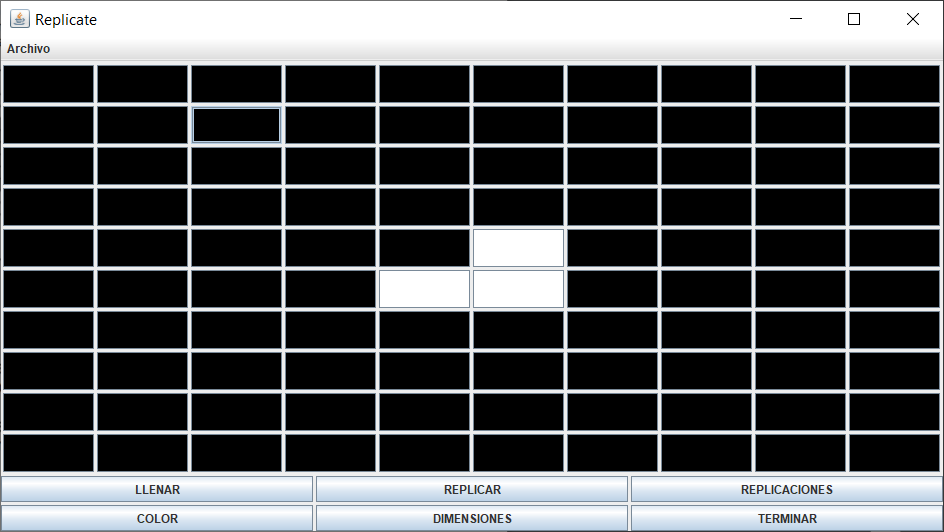
El objetivo es implementar la capa de aplicación para Replicate.

1. Construya los métodos básicos del juego (No olvide MDD y TDD) Clase replicate , metodos : contarCelAlrededores (boolean[][] mat,int i, int j)

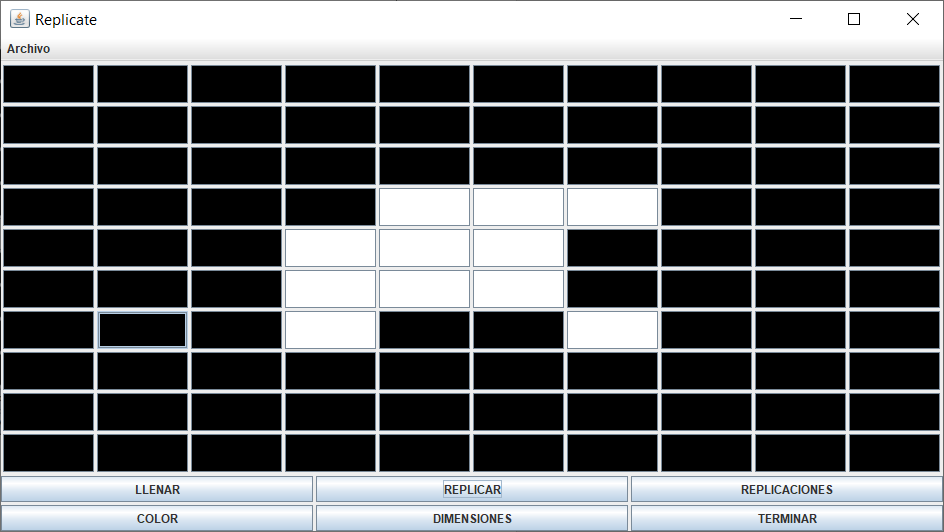
replicar()

2. Ejecuten las pruebas y capturen el resultado.

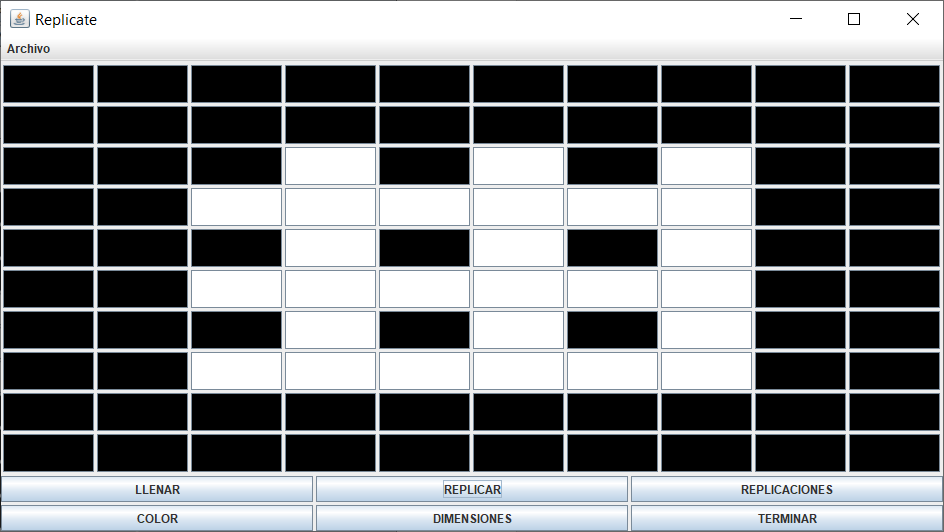
Estado 1:



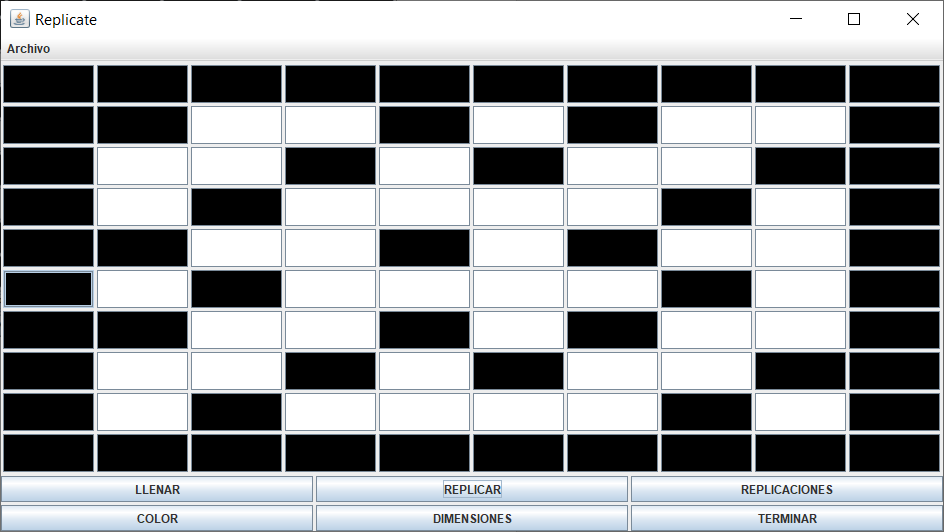
Estado 2:



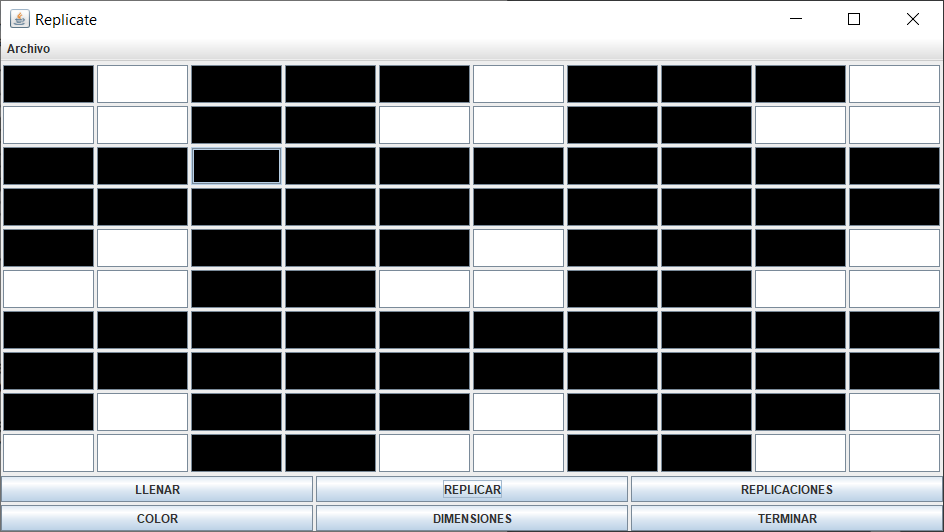
Estado 3:



Estado 4:



Estado 5:



**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.

seleccionarCasilla(int i, int j,boolean estado)

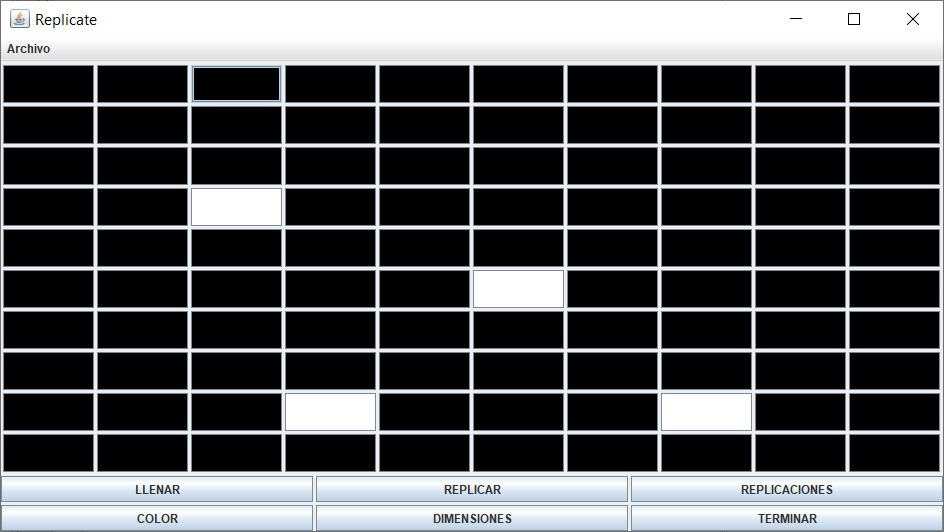
getMatriz()

llenarMatriz()

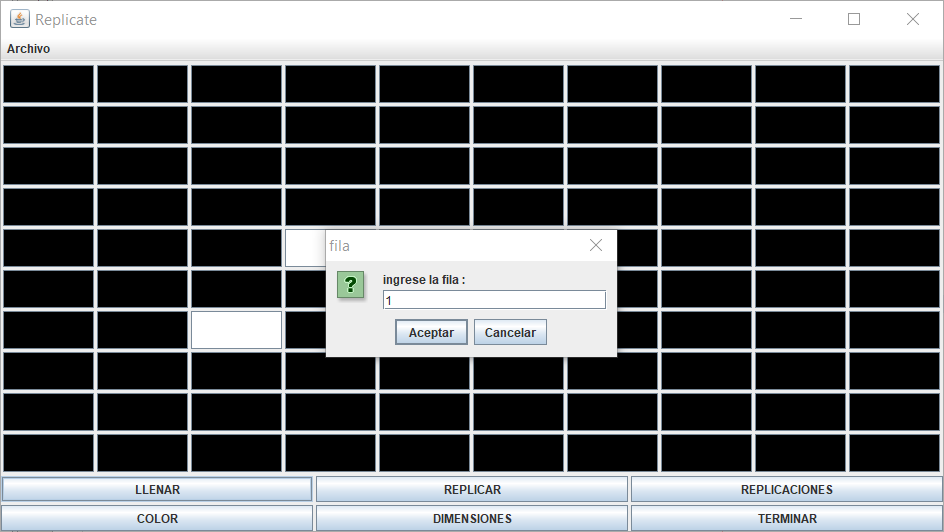
4. Implementen los componentes necesarios para jugar .

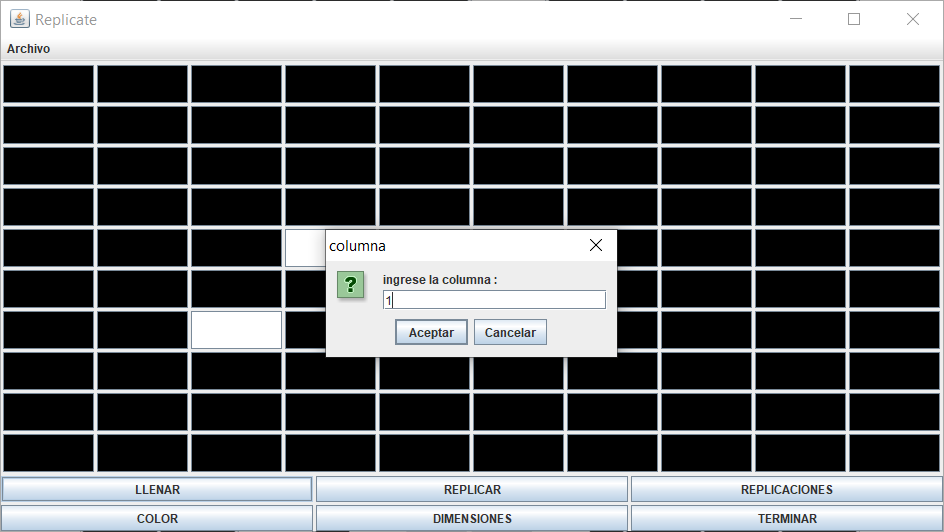
5. Ejecuten el caso de uso y capture las pantallas más significativas.

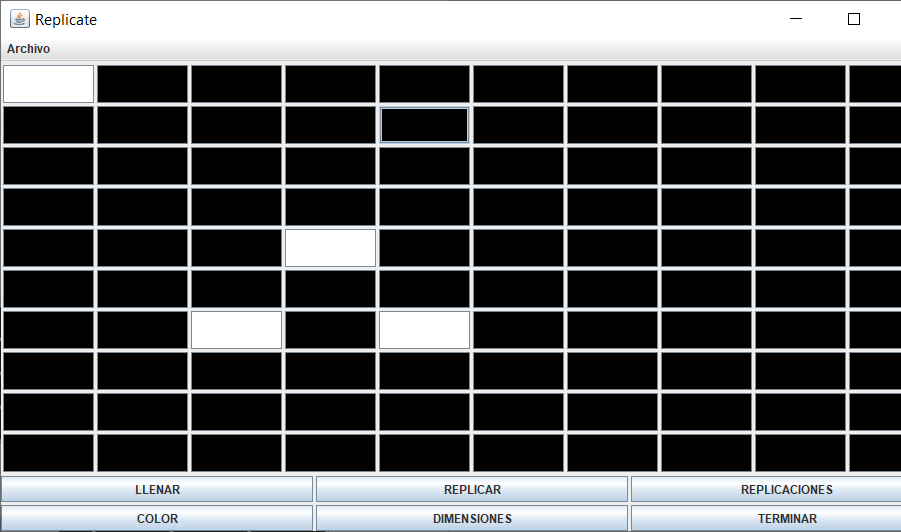
Seleccionar Casilla :



Llenar :







**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.

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:**

**[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

getDimension()

resize (int dim)

llenarMatriz()

salvar(File doc)

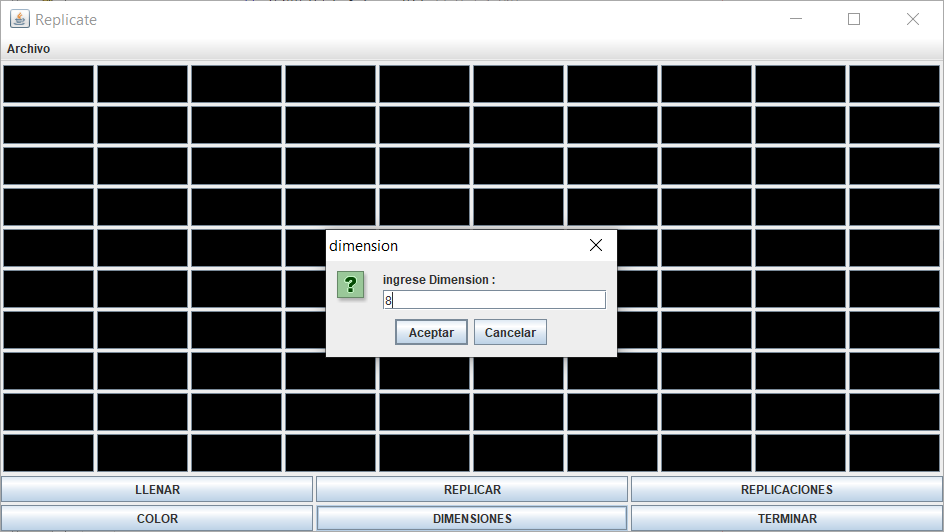
abrir(File doc)

replicaciones()

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

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

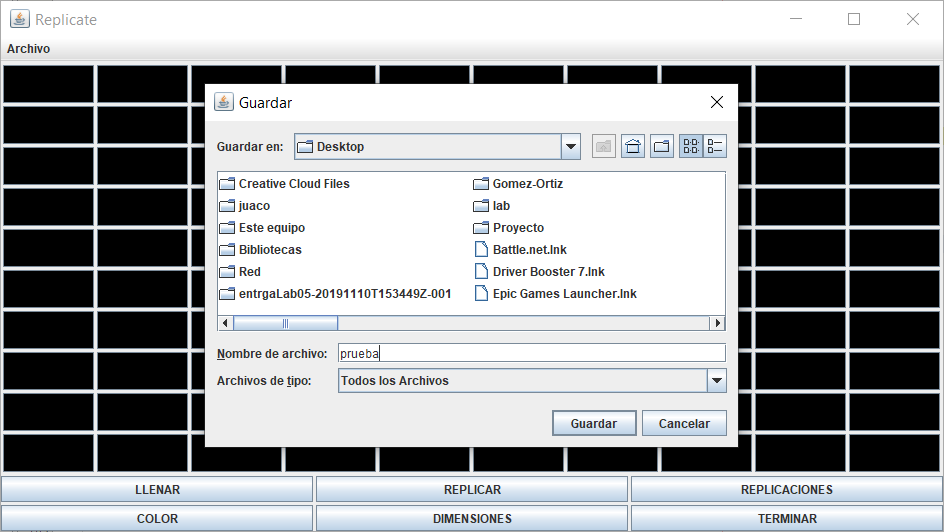
Dimensiones



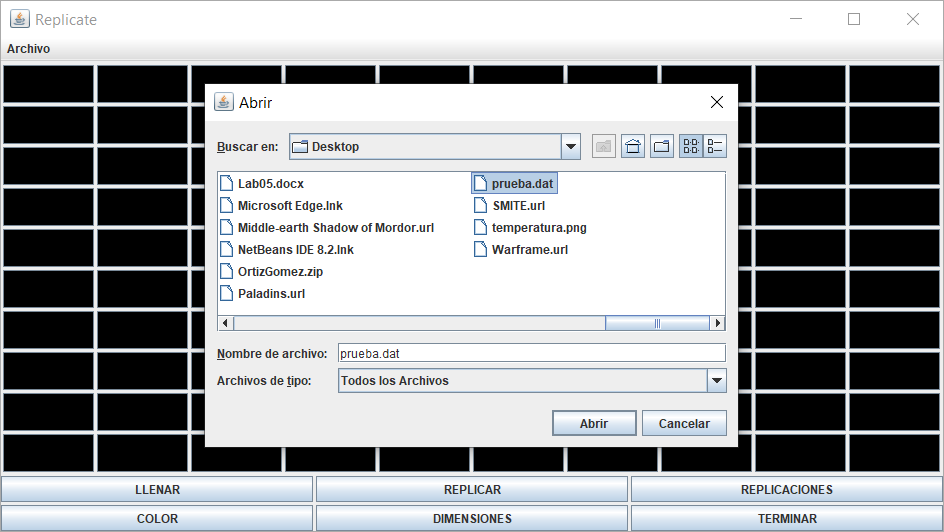
Matrix 8 x 8 :



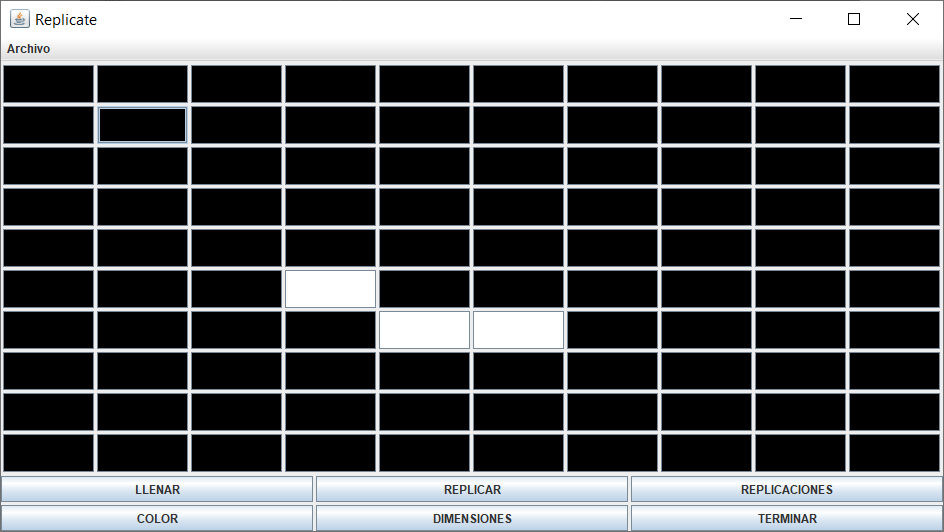
Metodo salvar :

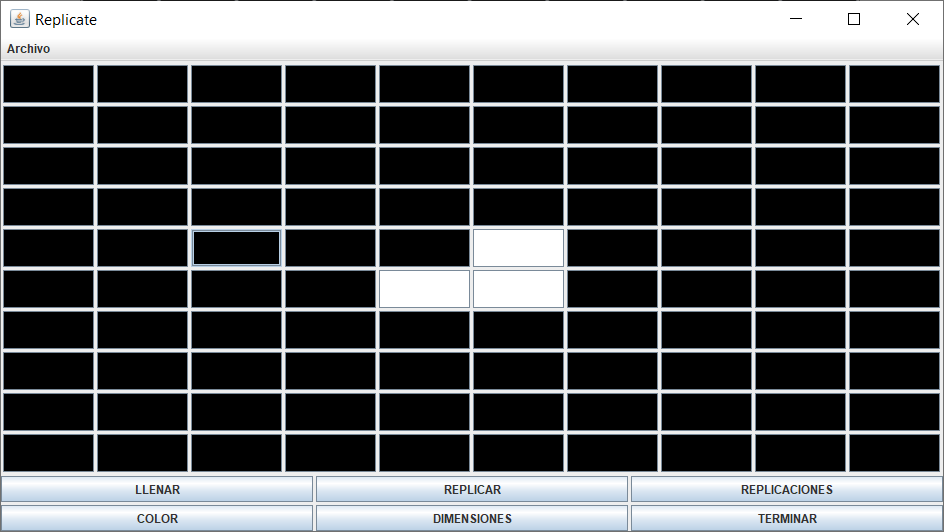


Metodo abrir :

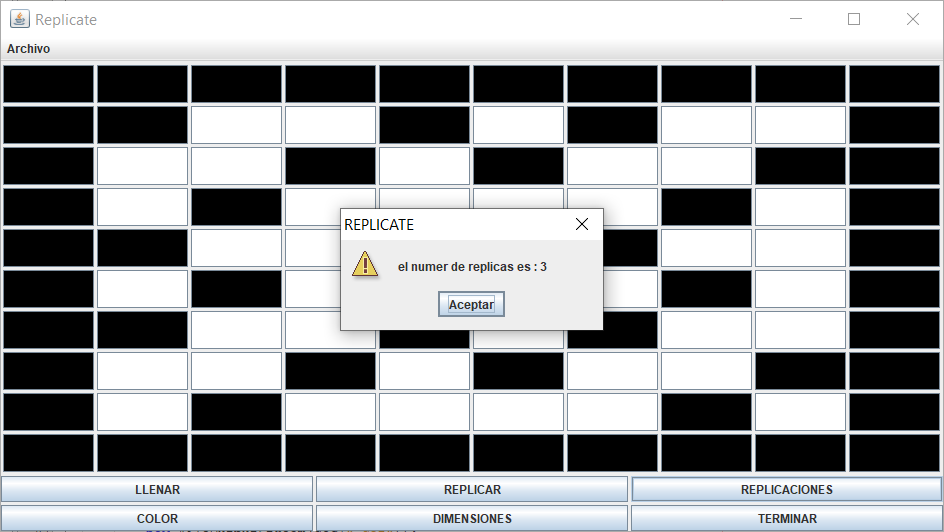


Resultado final :





Consultar replicaciones despues de 3 replicates()



**RETROSPECTIVA**

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

Nombre)

10 hr 🡪 Daniel Gomez

10 hr 🡪Juan Camilo Ortiz

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

100% funcional con todos los requerimientos solicitados .

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

Pares, porque consideramos que al turnarnos tener el mando en la máquina, la persona que esta observando no ve lo mismo que el que está ahí frente al computador.

4. ¿Cuál consideran fue su mayor logro? Sacar el replicate ¿Por qué? Era el metodo mas complejo de este laboratorio, despues de que lo sacamos , fue mas sencillo hacer las otras funcionalidades . ¿Cuál consideran que fue su mayor problema? Uso de interfaz , botones y persistencia ¿Qué hicieron para resolverlo? Investigar como funcionaba lo de exportar el archivo y serializar objetos.

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

Nos entendemos muy bien, ambos aportamos buenas ideas y es importante aceptar nuestros propios errores. Aparte de eso, nos costó repartir bien el tiempo , sin embargo, esperamos solucionarlo en el trascurso del tiempo.