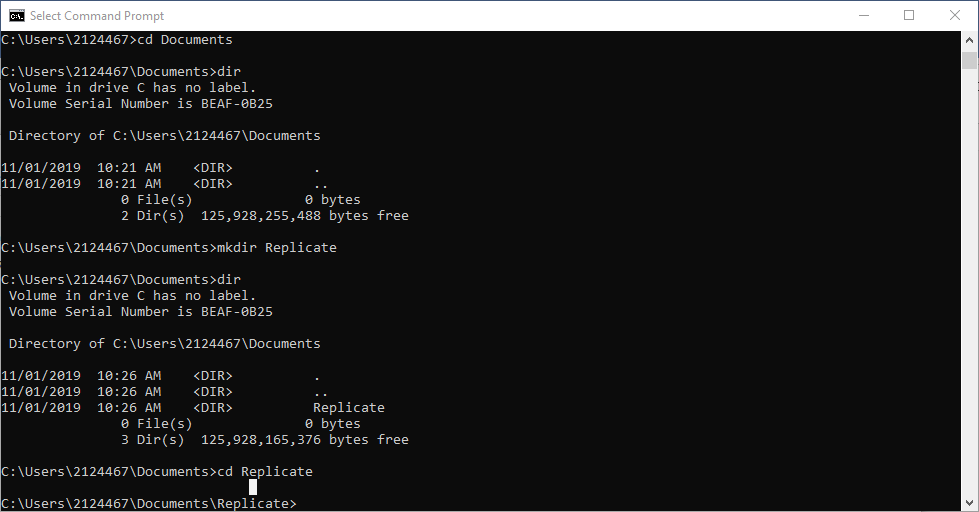
# PROGRAMACIÓN ORIENTADA A OBJETOS INTERFAZ 2019-02 Daniel Alfonso – Brayan Burgos Laboratorio 5/6

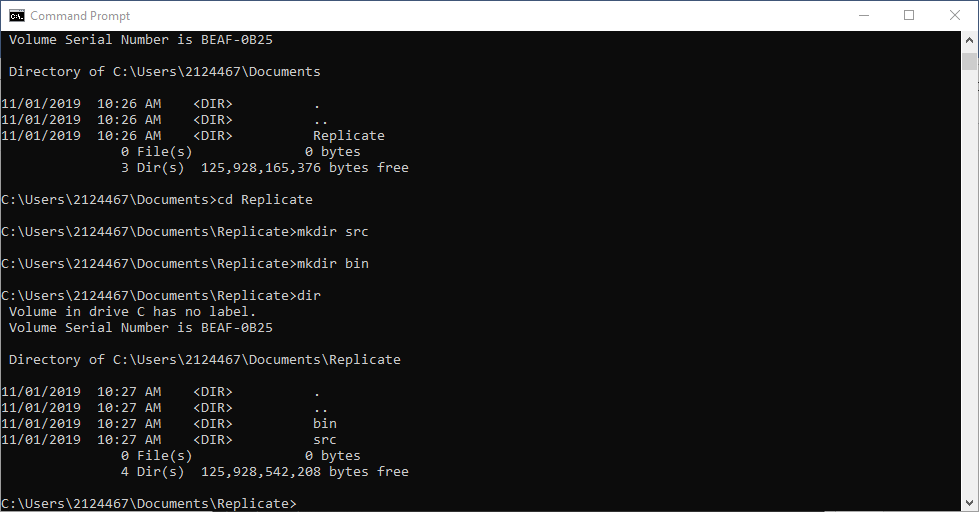
## DESARROLLO

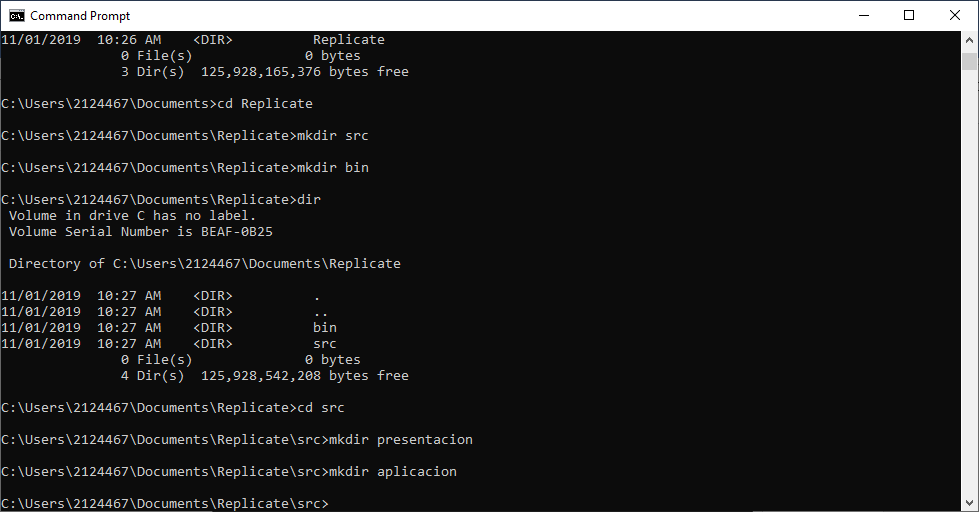
Directorios

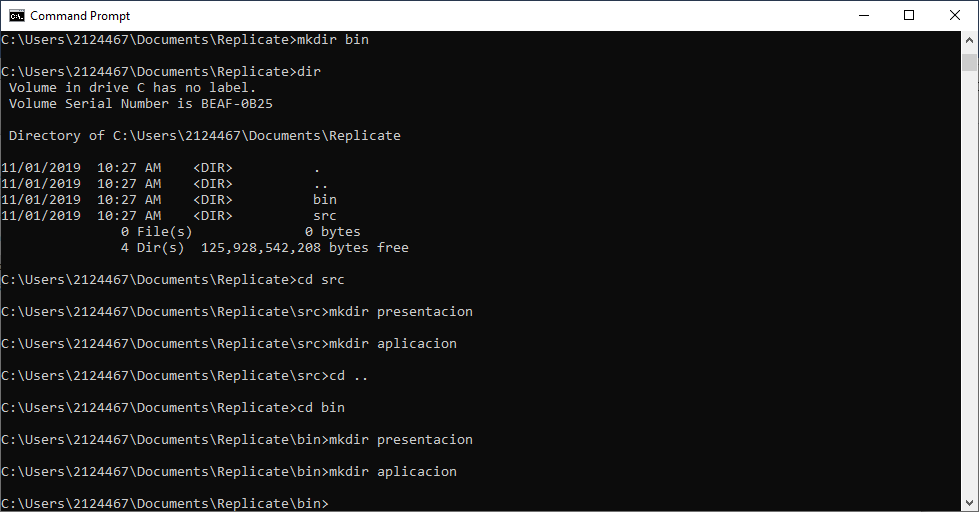
El objetivo de este punto es construir un primer esquema para el juego Replicate .

1. Preparen un directorio llamado Replicate con los directorios src y bin y los subdirectorios presentación y aplicación.





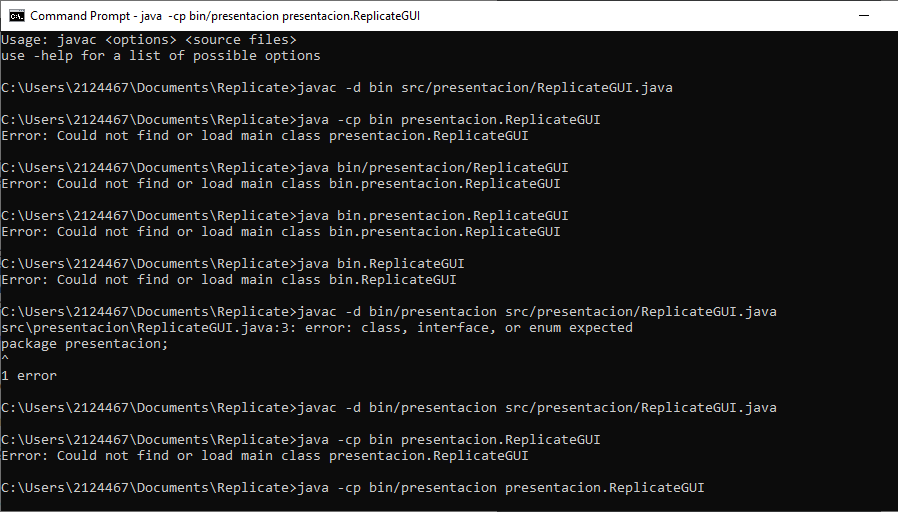


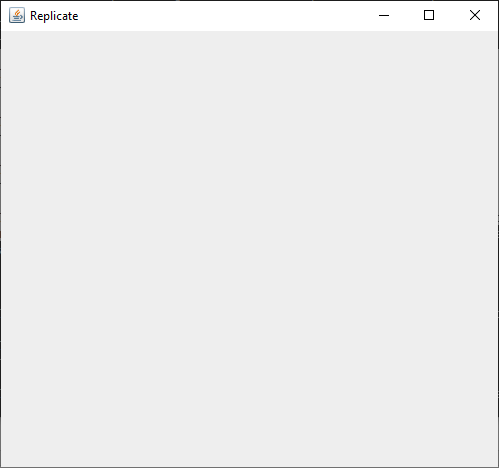


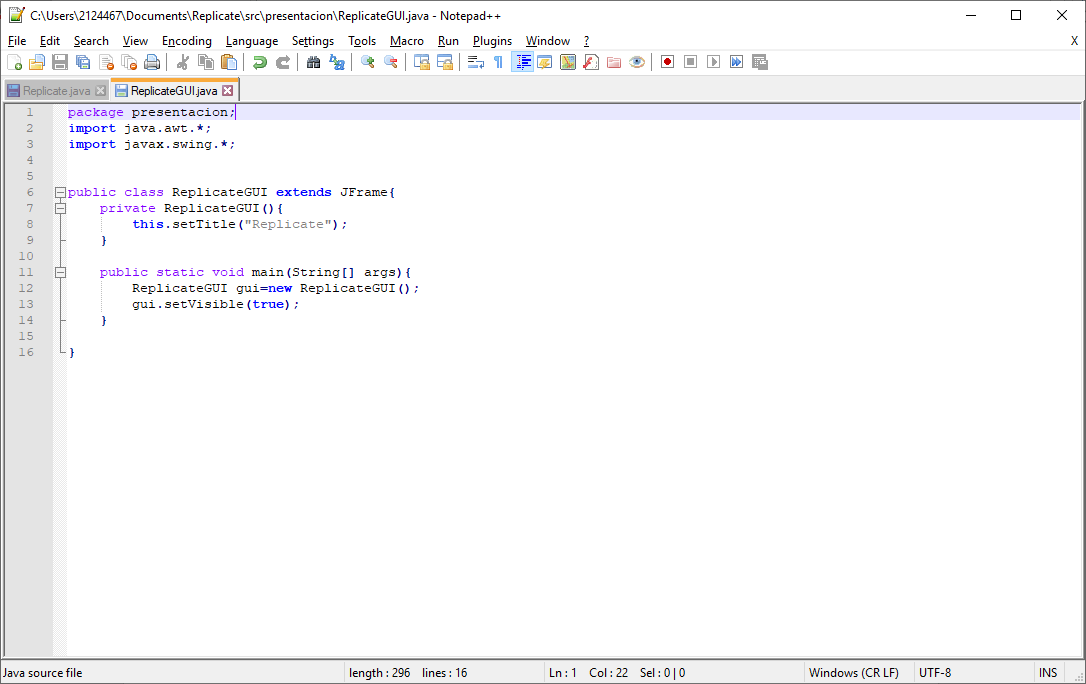
# Ciclo 0: Ventana vacía – Salir

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 prepareElementos prepareAcciones.

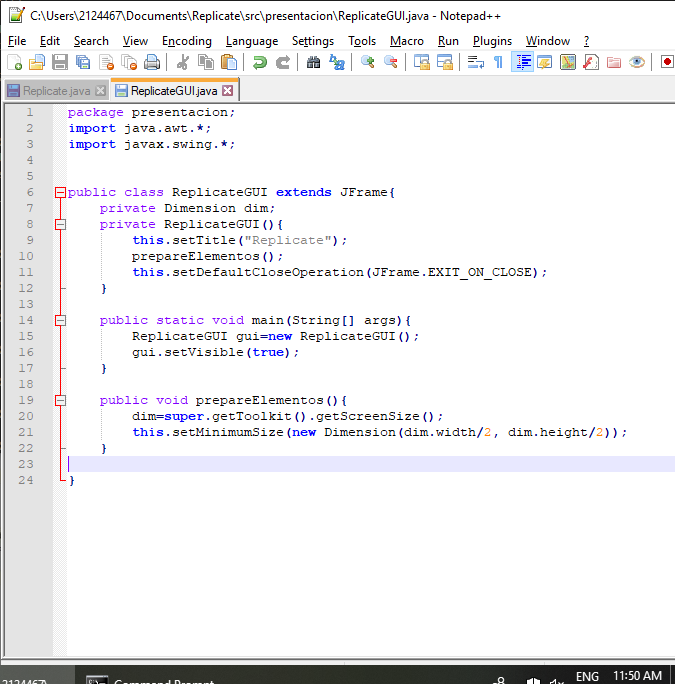
1. Construyan el primer esquema de la ventana de Replicate únicamente con el título  
   “ Replicate”. Para esto cree la clase ReplicateGUI como un JFrame con su creador, que sólo coloca el título, y el método main que crea un objeto ReplicateGUI y lo hace visible. Ejecútenlo. Capturen la pantalla. (Si la ventana principal no es la inicial en su diseño, después moverá el main al componente visual correspondiente)

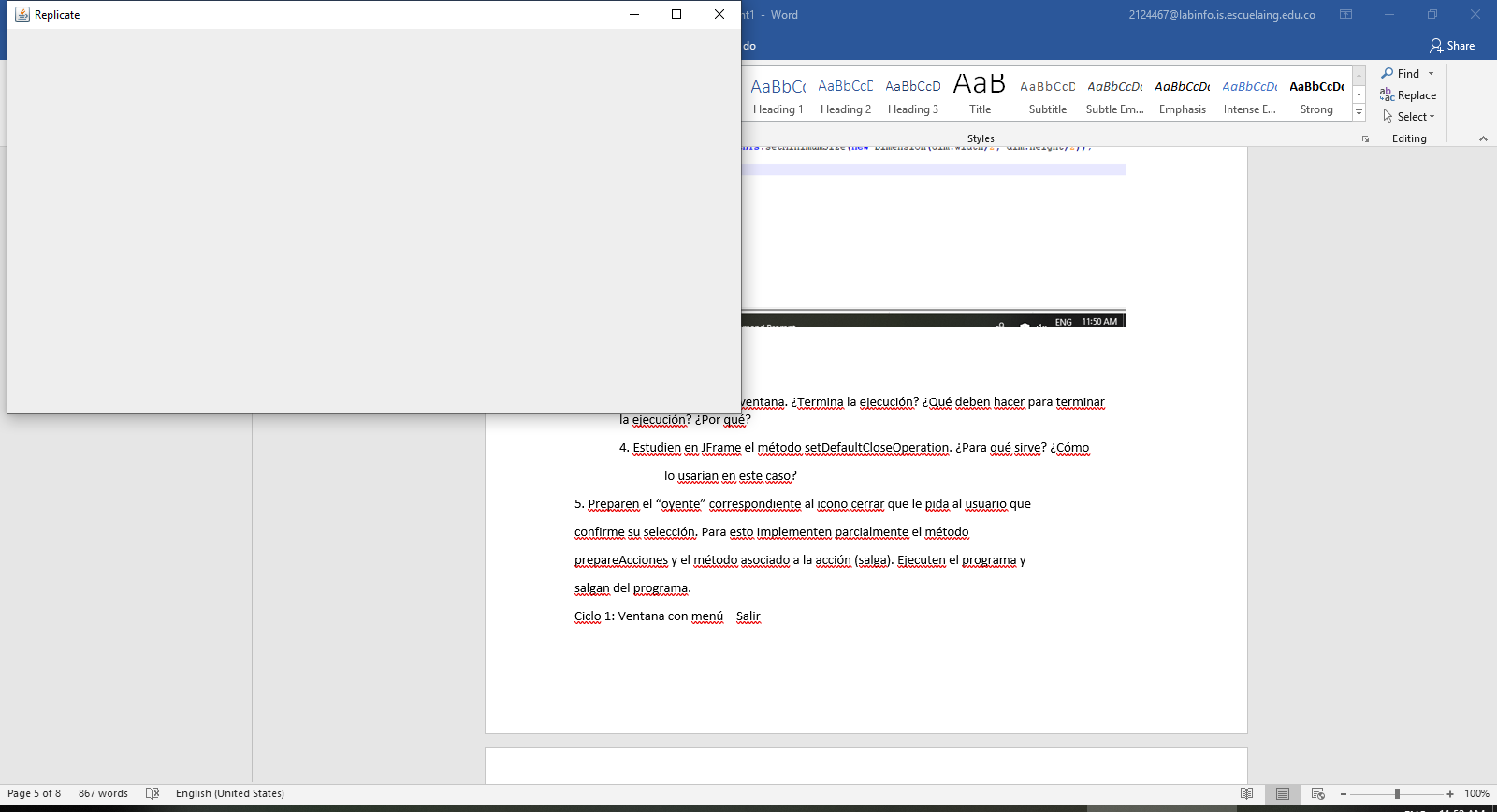






1. Modifiquen el tamaño de la ventana para que ocupe un cuarto de la pantalla y  
   ubíquenla en el centro (prepareElementos). Capturen esa pantalla.





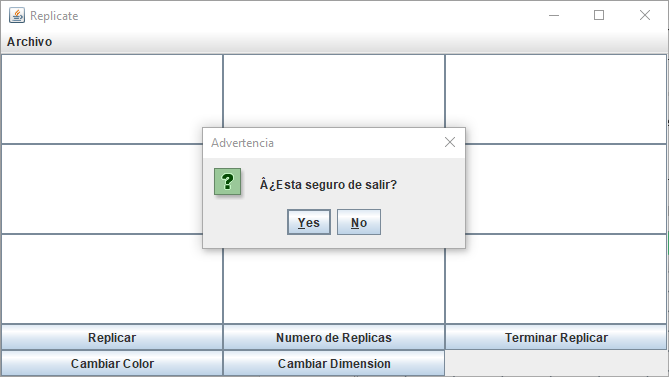
1. Traten de cerrar la ventana. ¿Termina la ejecución? ¿Qué deben hacer para terminar  
   la ejecución? ¿Por qué?

No termina la ejecución, porque falta el método setDefaultCloseOperation para que cuando demos click en cerrar termine la ejecución

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

El método setDefaultCloseOpération sirve para terminar la ejecución de un proceso

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

[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.



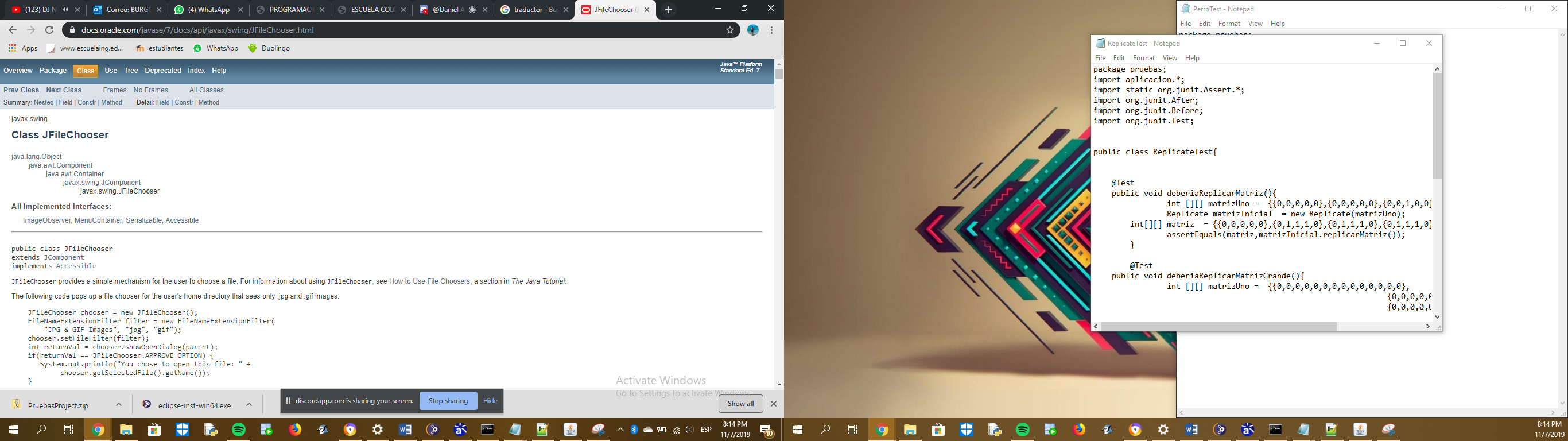
# Ciclo 2: Salvar y abrir

[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.



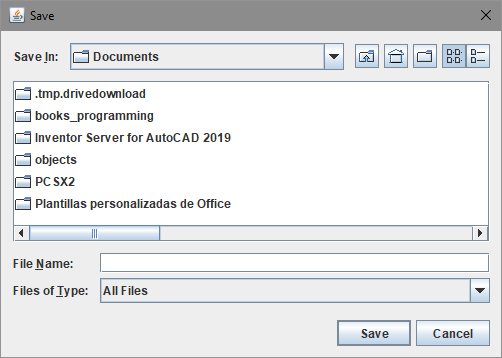
2. Implementen parcialmente los elementos necesarios para salvar y abrir. Al

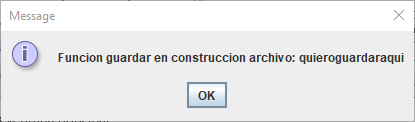
seleccionar los archivos indique que las funcionalidades están en construcción

detallando la acción y el nombre del archivo seleccionado.

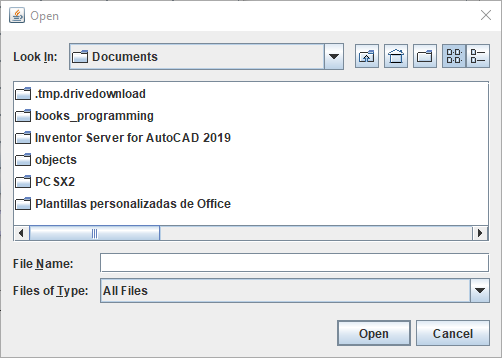
1. Ejecuten las dos alternativas y capture las pantallas más significativas.

Save





Open



# 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.

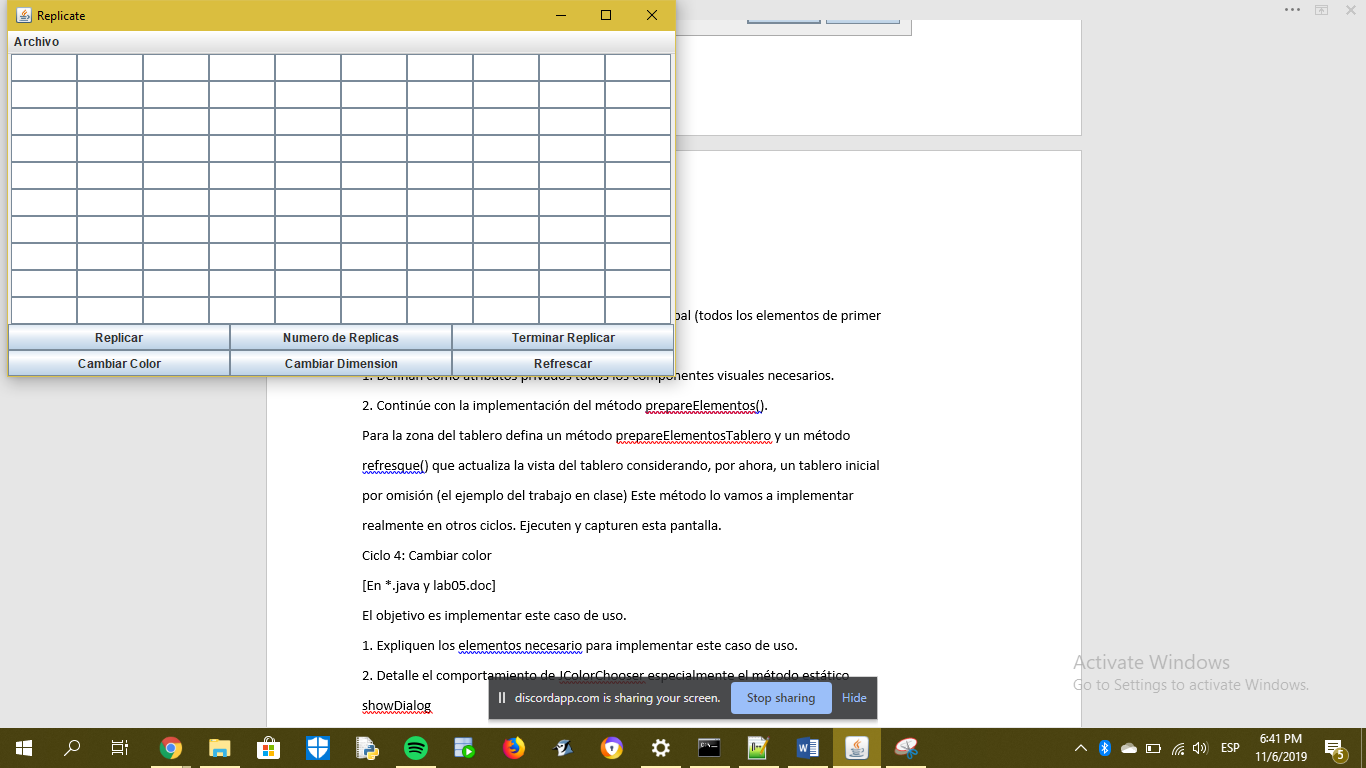
2. Continúe con la implementación del método prepareElementos().

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

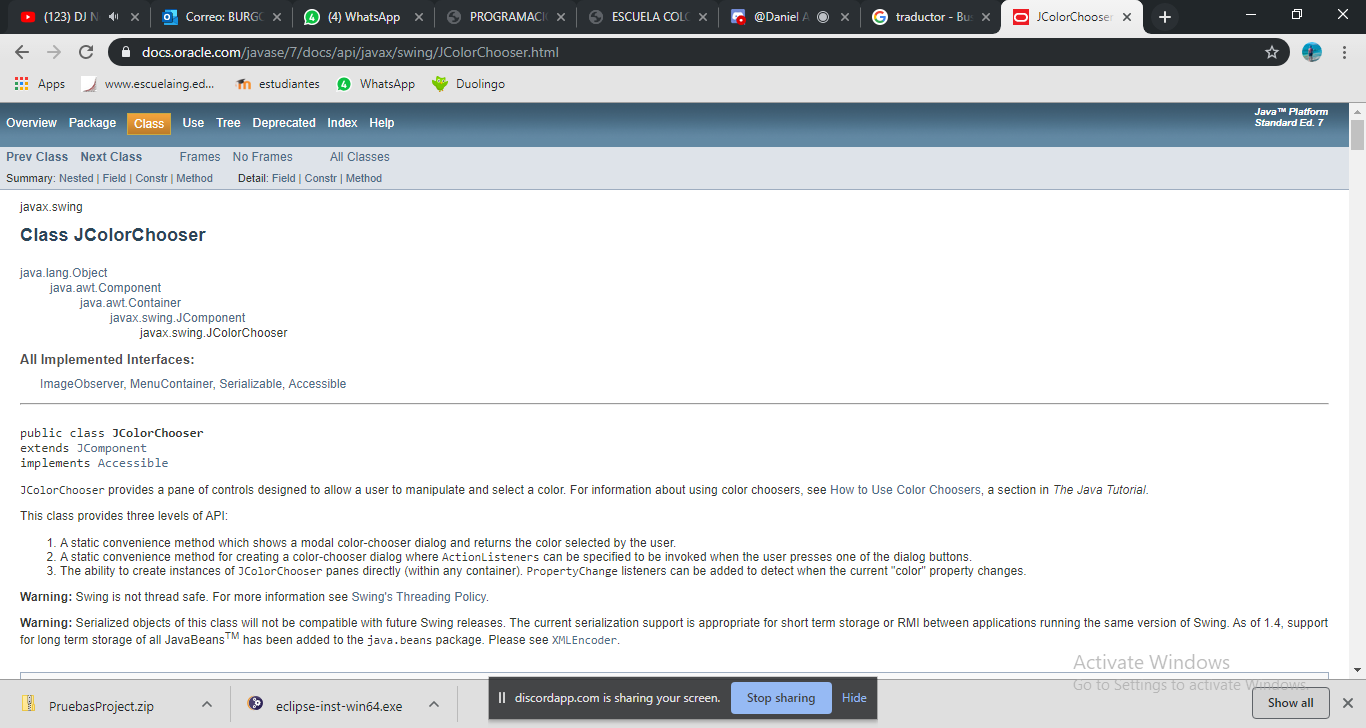
[En \*.java y lab05.doc]

El objetivo es implementar este caso de uso.

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

2. Detalle el comportamiento de JColorChooser especialmente el método estático

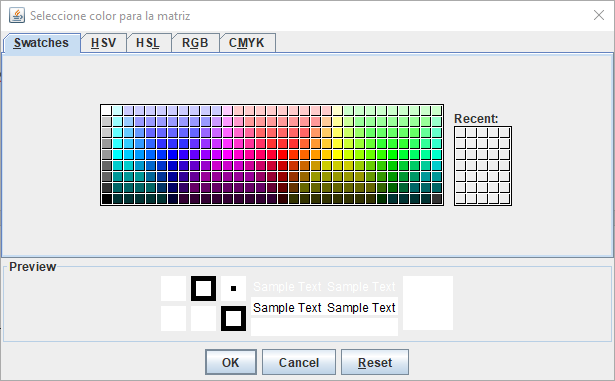
showDialog

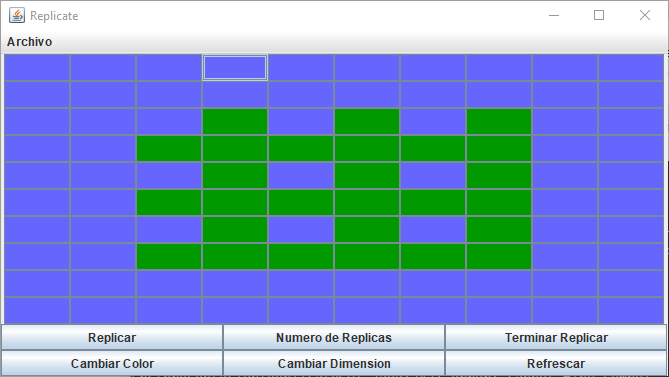


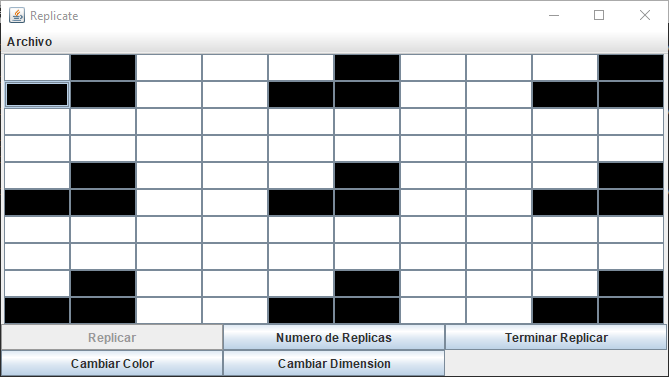
3. Implementen los componentes necesarios para cambiar el color de las casas y

almacenes.

1. 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)

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 refresque() considerando la información del modelo de

dominio.

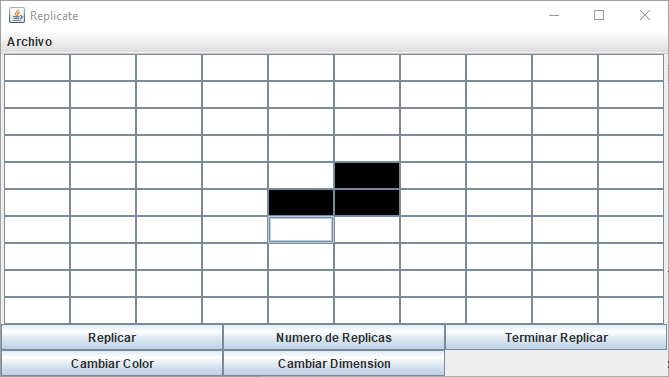
1. Expliquen los elementos necesarios para implementar este caso de uso.

-Los elementos necesarios son: una matriz que tiene el control mediante unos y ceros

-Luego, encapsula la colección para poder replicar la matriz anterior

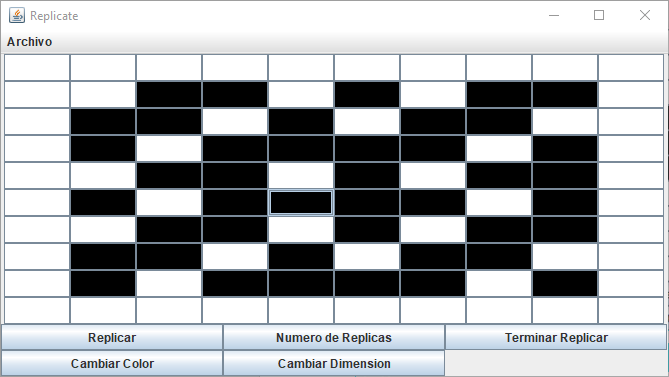
4. Implementen los componentes necesarios para jugar.

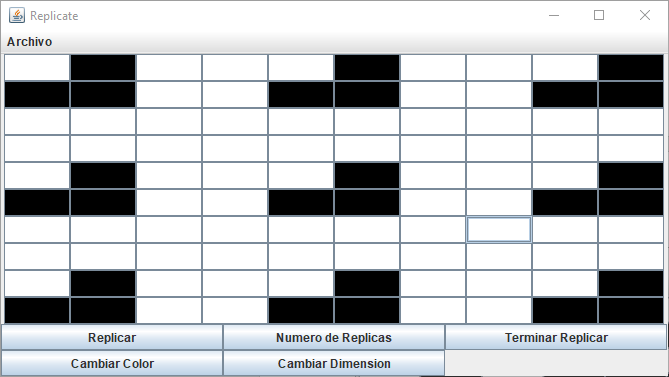
5. Ejecuten el caso de uso y capture las pantallas más significativas.











CASO Planteado EN LA GUIA

# Ciclo 7: Reiniciar

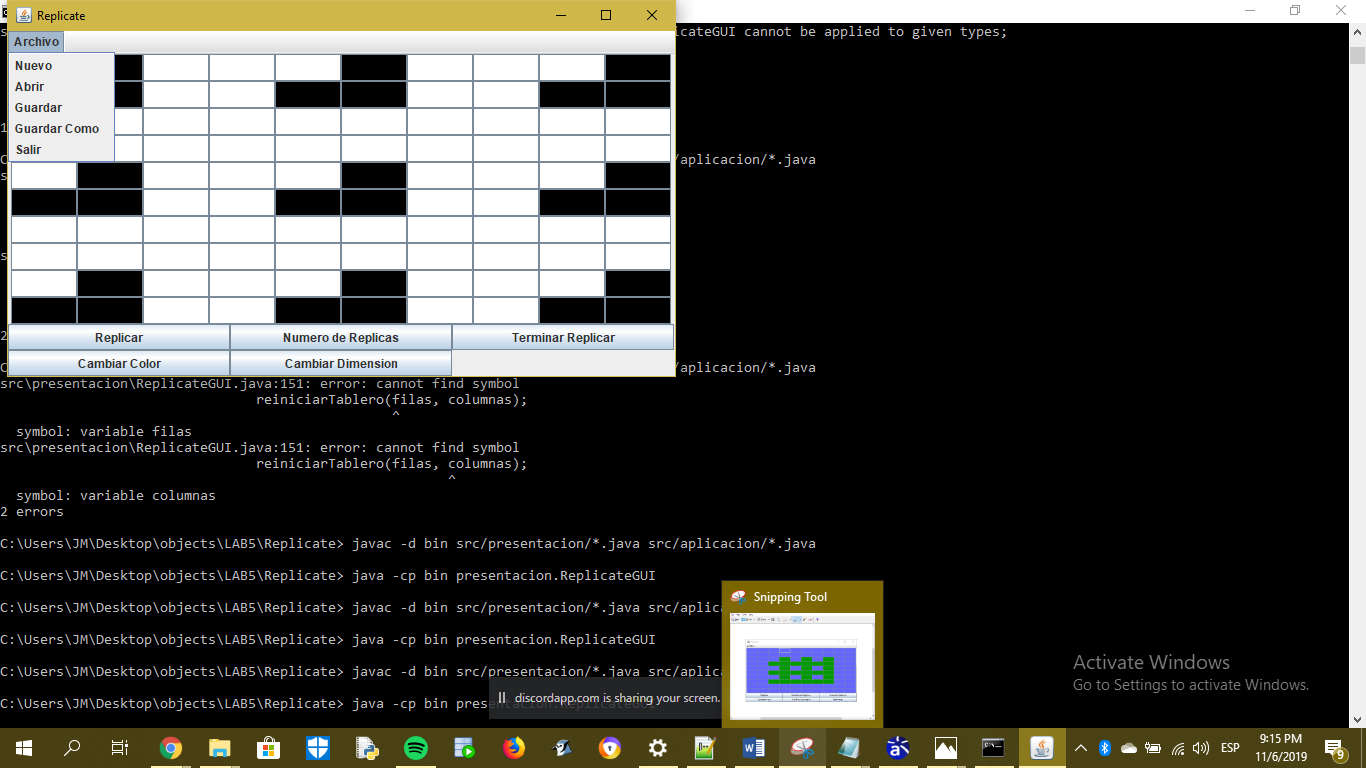
[En \*.java y lab05.doc]

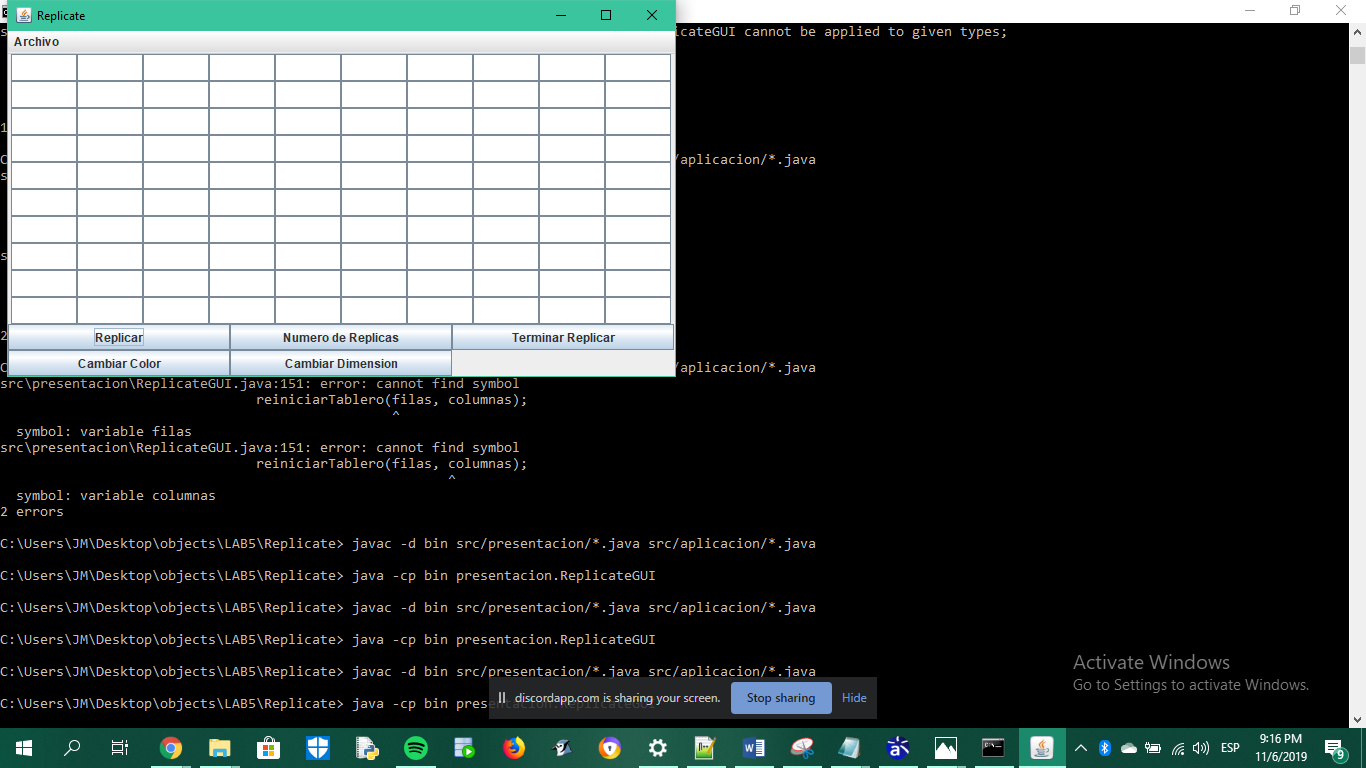
El objetivo es implementar este caso de uso.

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: casas y semillas

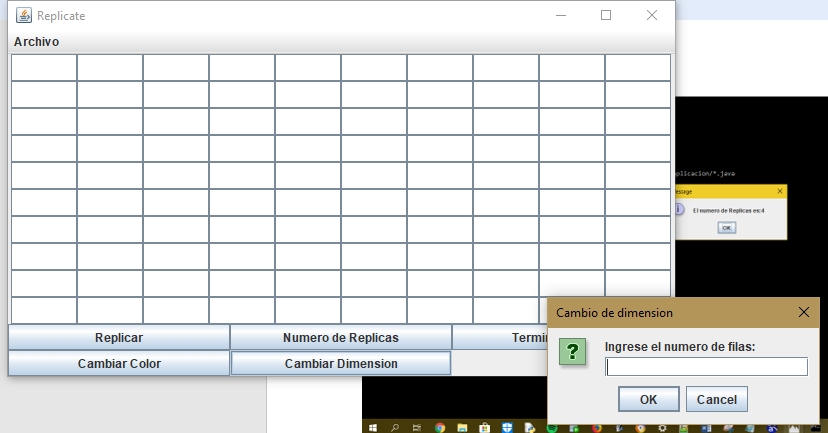
[En \*.java y lab05.doc]

El objetivo es implementar este caso de uso.

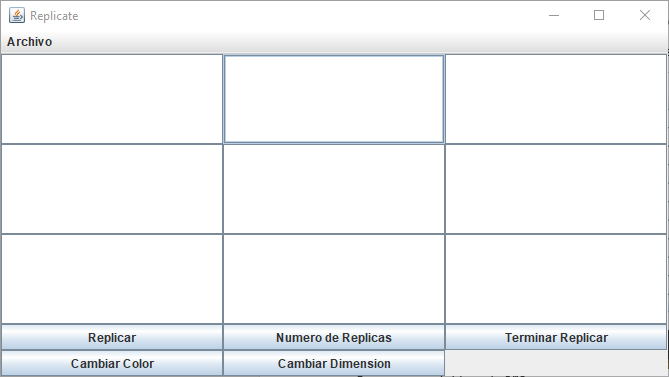
1. Expliquen los elementos a usar para implementar este caso de uso

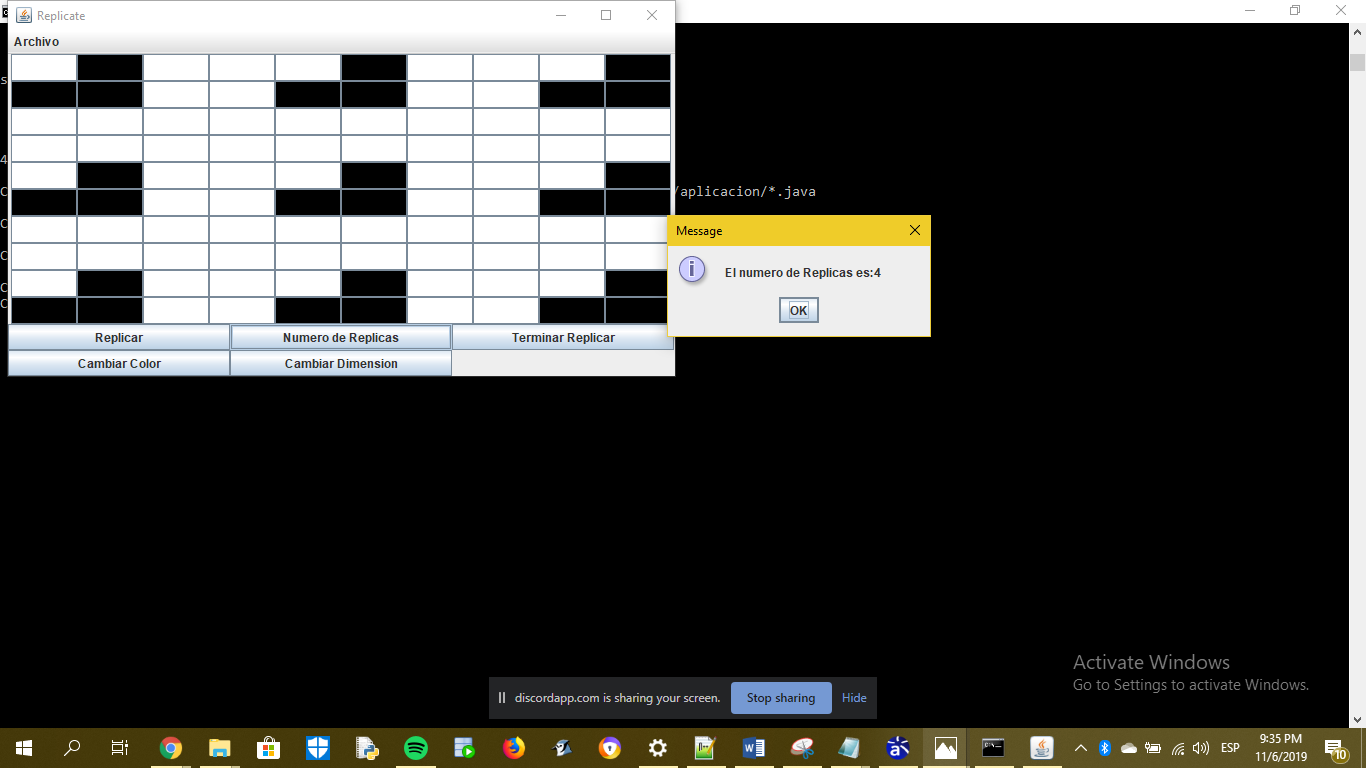
2. Implementen los elementos necesarios para cambiar el tamaño del juego

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



Queremos un tablero de 3\*3





# RETROSPECTIVA

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

El tiempo invertido fueron 12 horas por cada uno

2. ¿Cuál consideran fue el mayor logro? ¿Por qué?

El mayor logro para nosotros es, que cada vez vamos entendiendo más lo que es la programación orientada a objetos y logramos aplicar, herencias, superclases, extensibilidad, paquetes y demás

3. ¿Cuál consideran que fue el mayor problema técnico? ¿Qué hicieron para resolverlo?

El tiempo de entrega a pesar de haber sido extendido, tuvimos bastante problema con el tiempo que pudimos invertir, además del manejo de la sintaxis del lenguaje que aún no es la mejor para avanzar de manera óptima el proceso del proyecto

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

Como equipo siempre escuchamos la opinión del otro y tratamos de llegar a un acuerdo sobre lo acordado. La utilización de la programación en parejas. Nos comprometemos a esforzarnos y seguir con la disciplina que logramos en este proyecto para lograr los objetivos satisfactoriamente

5. Considerando las prácticas XP del laboratorio. ¿Cuál fue la más útil? ¿Por qué?

Las dos prácticas que conocemos las utilizamos, las iteraciones las utilizamos para dividir en partes más pequeñas los problemas y las más útil fue la programación a pares que ayuda a entenderse y lograr mejores trabajos de calidad en pareja