**PROGRAMACIÓN ORIENTADA A OBJETOS**

**INTERFAZ**

**Laboratorio 05**

**Javier Esteban Lopez Peña**

**Andres Felipe Davila Gutierrez**

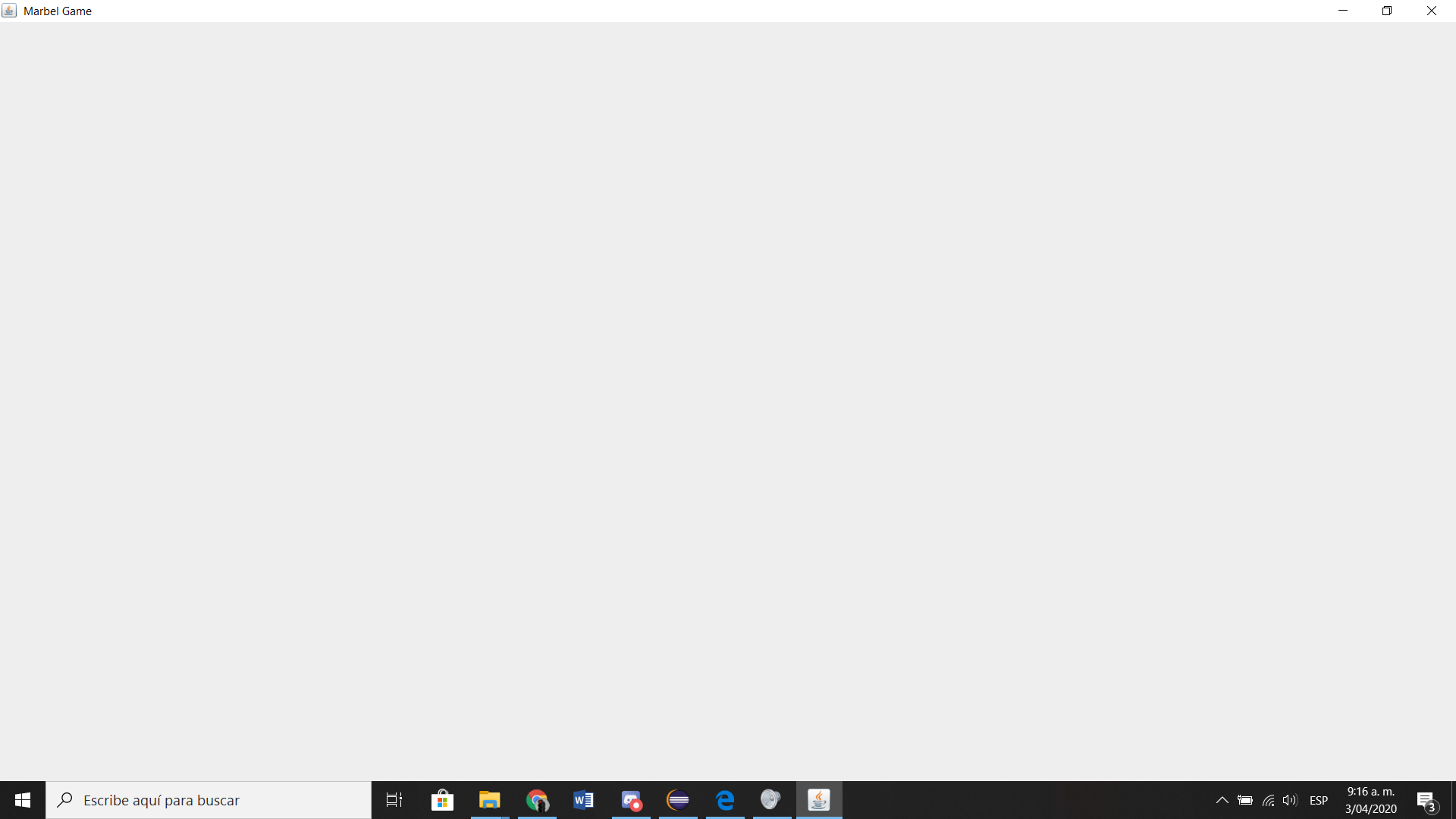
**2020-1**

**Marbel Game**

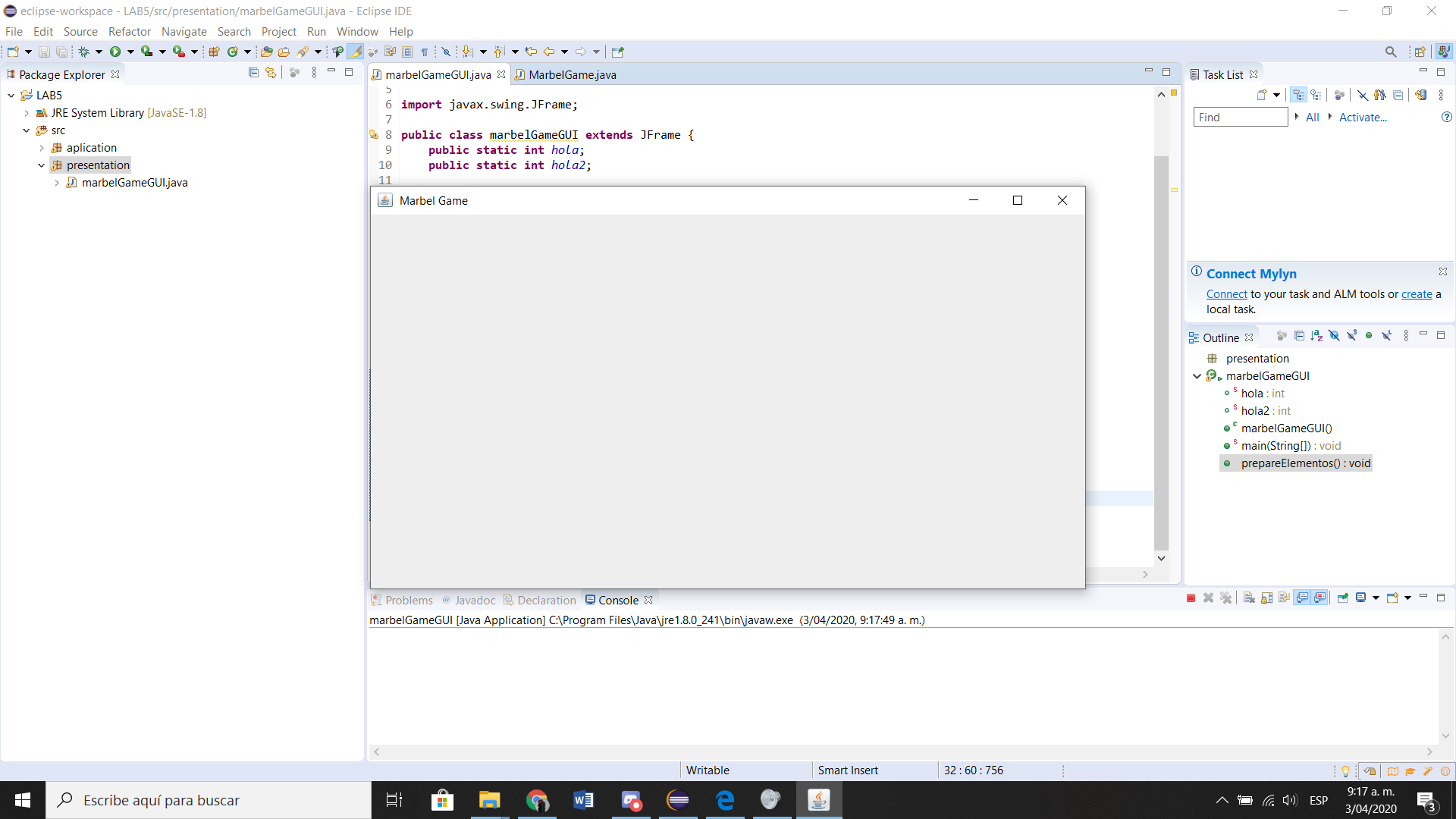
**Ciclo 0: Ventana vacía – Salir**

**El objetivo es implementar la ventana principal de MarbelGame 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 MarbelGame únicamente con el título “ Marbel Game”. Para esto cree la clase MarbelGameGUI como un JFrame con su creador, que sólo coloca el título, y el método main que crea un objeto MarbelGameGUI 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?**

Al oprimir el botón de salir no termina la ejecución, solo se esconde el frame debido a que es la opción por defecto de salir cuando no se utiliza el método setDefaultCloseOperation

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

**El método setDefaultCloseOperation contiene varias opciones de cerrar el programa y son las siguientes:**

**JFrame.EXIT\_ON\_CLOSE:** cierra el frame y termina la ejecución

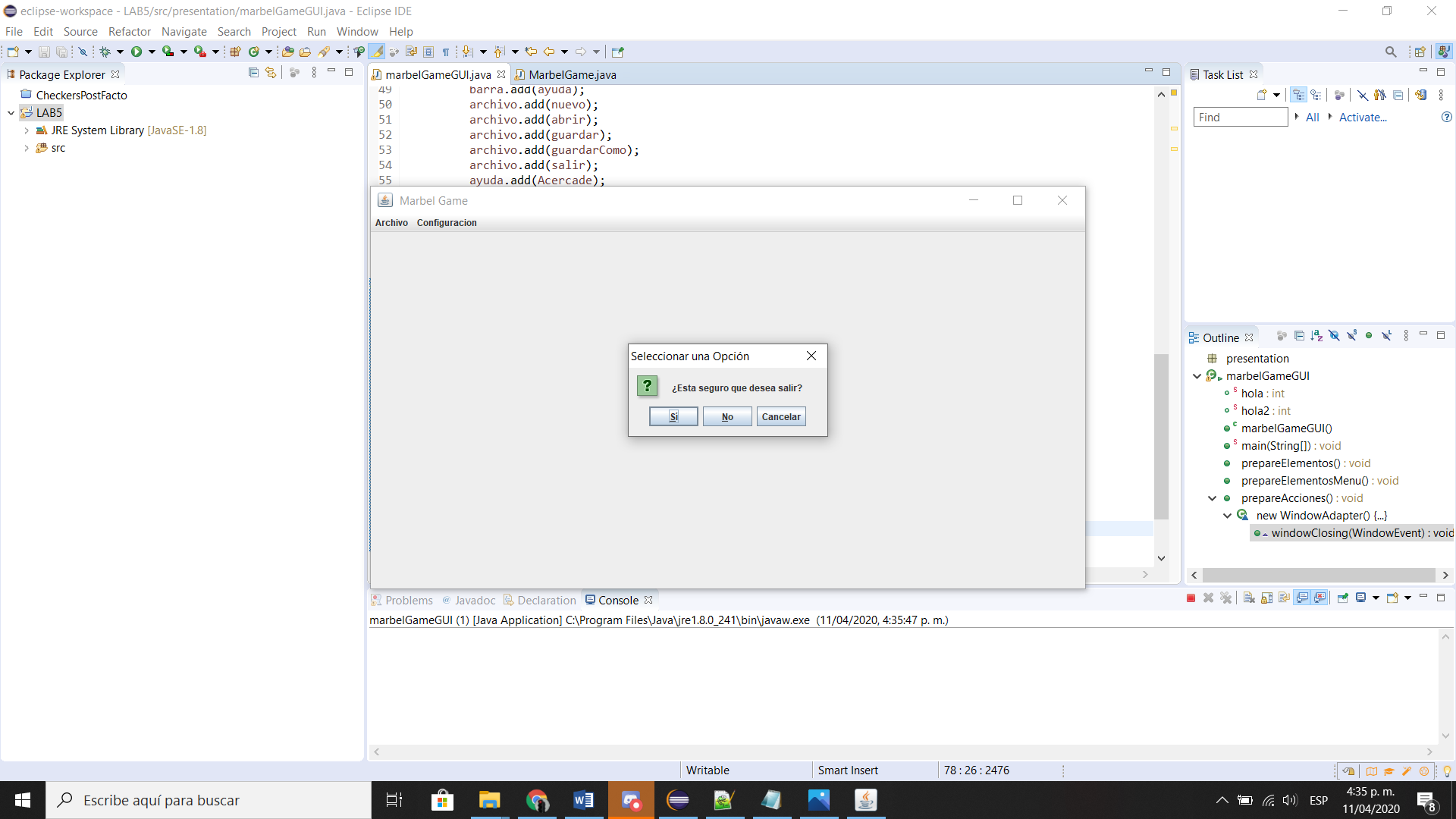
**JFrame.HIDE\_ON\_CLOSE:** esconde el frame pero no termina la ejecucion

**JFrame.DISPOSE\_ON\_CLOSE:** desecha el marco frame pero mantiene la aplicación corriendo.

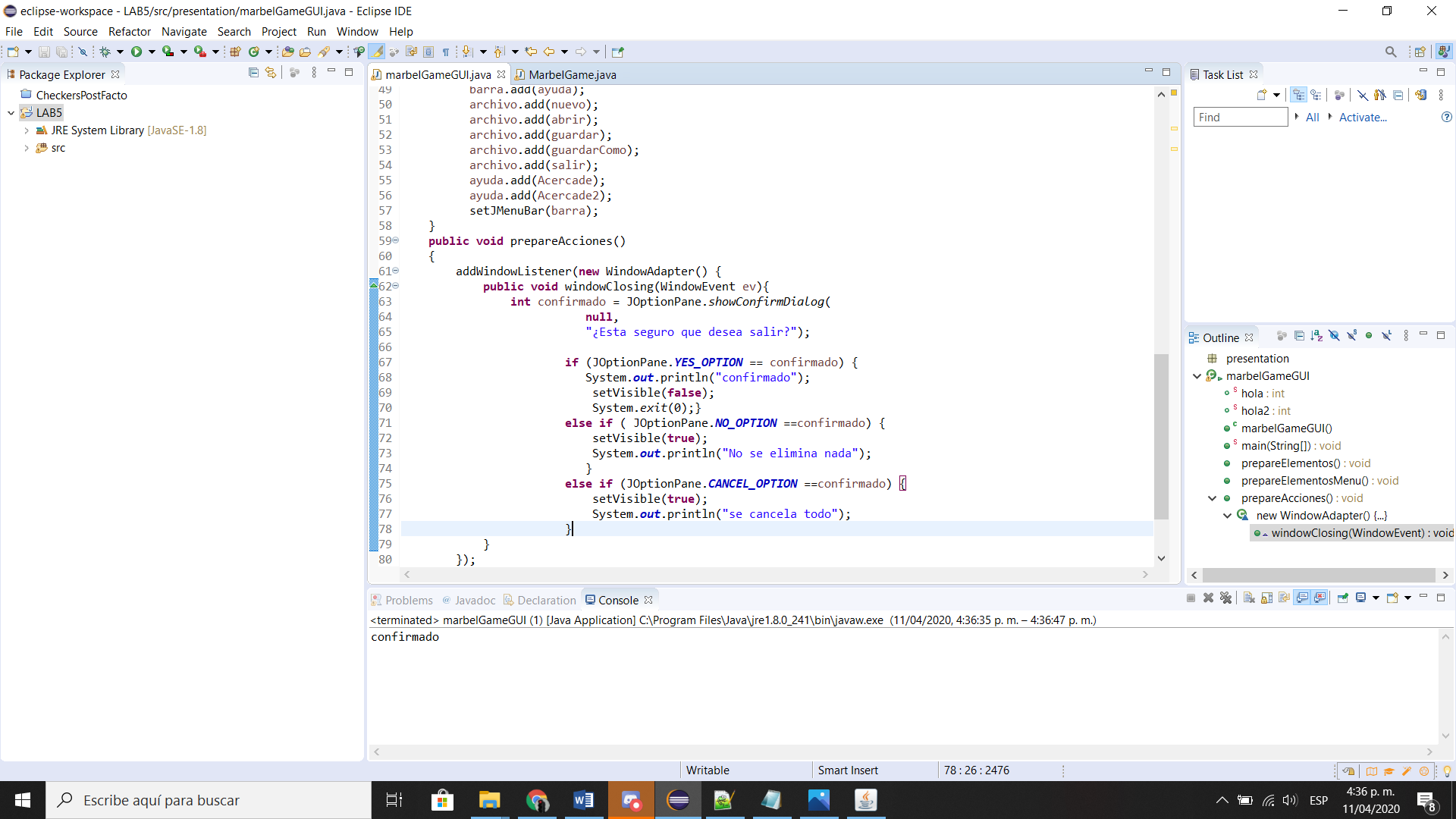
**JFrame.DO\_NOTHING\_ON\_CLOSE:** al oprimir el botón no hace nada

El que vamos a utilizar para este caso es el método **EXIT\_ON\_CLOSE** para que termine la ejecución del programa correctamente

**5. Preparen el “oyente” correspondiente al icono cerrar que le pida al usuario que confirme su selección. Para esto Implementan parcialmente el método prepareAcciones y el método asociado a la acción (salga). Ejecuten el programa y salgan del programa. Capturen las pantallas.**



**Al OPRIMIR SI**



**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**

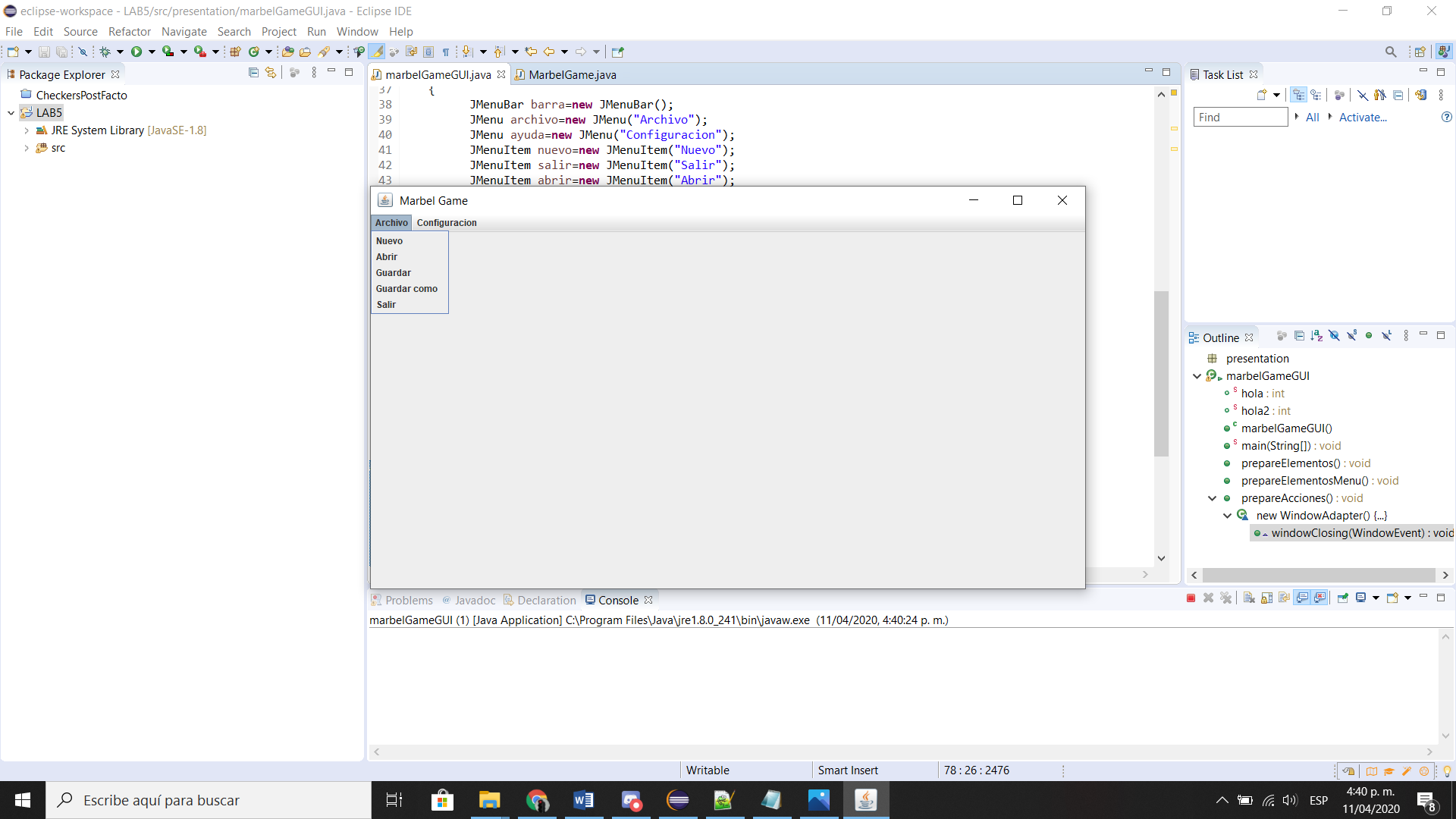
**opción del menú para salir. El menú debe ofrecer mínimo las siguientes opciones (Nuevo,**

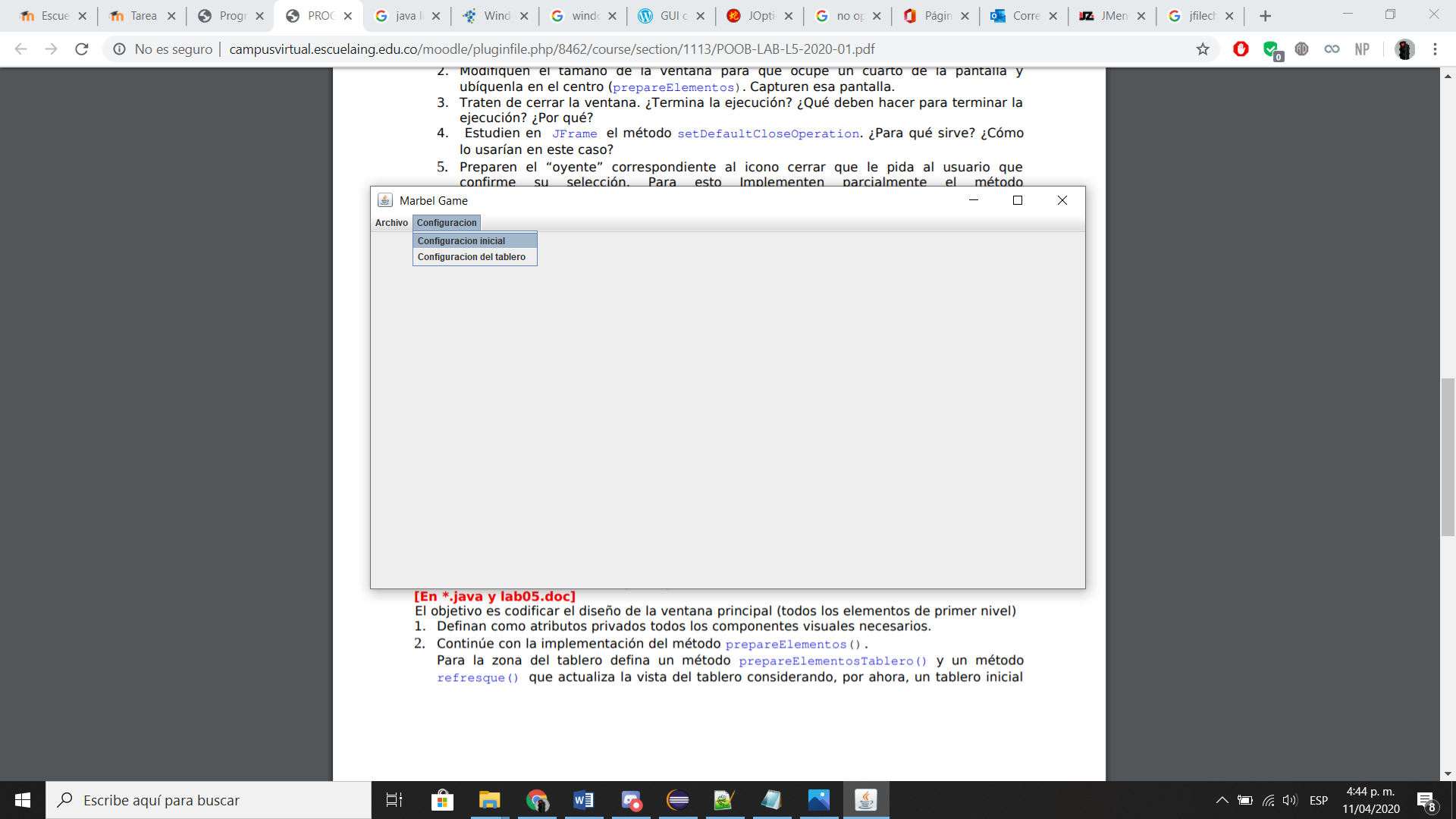
**Abrir – Salvar, Salvar Como – Salir) . No olviden los criterios de calidad del código.**

**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. Capturen la pantalla.**

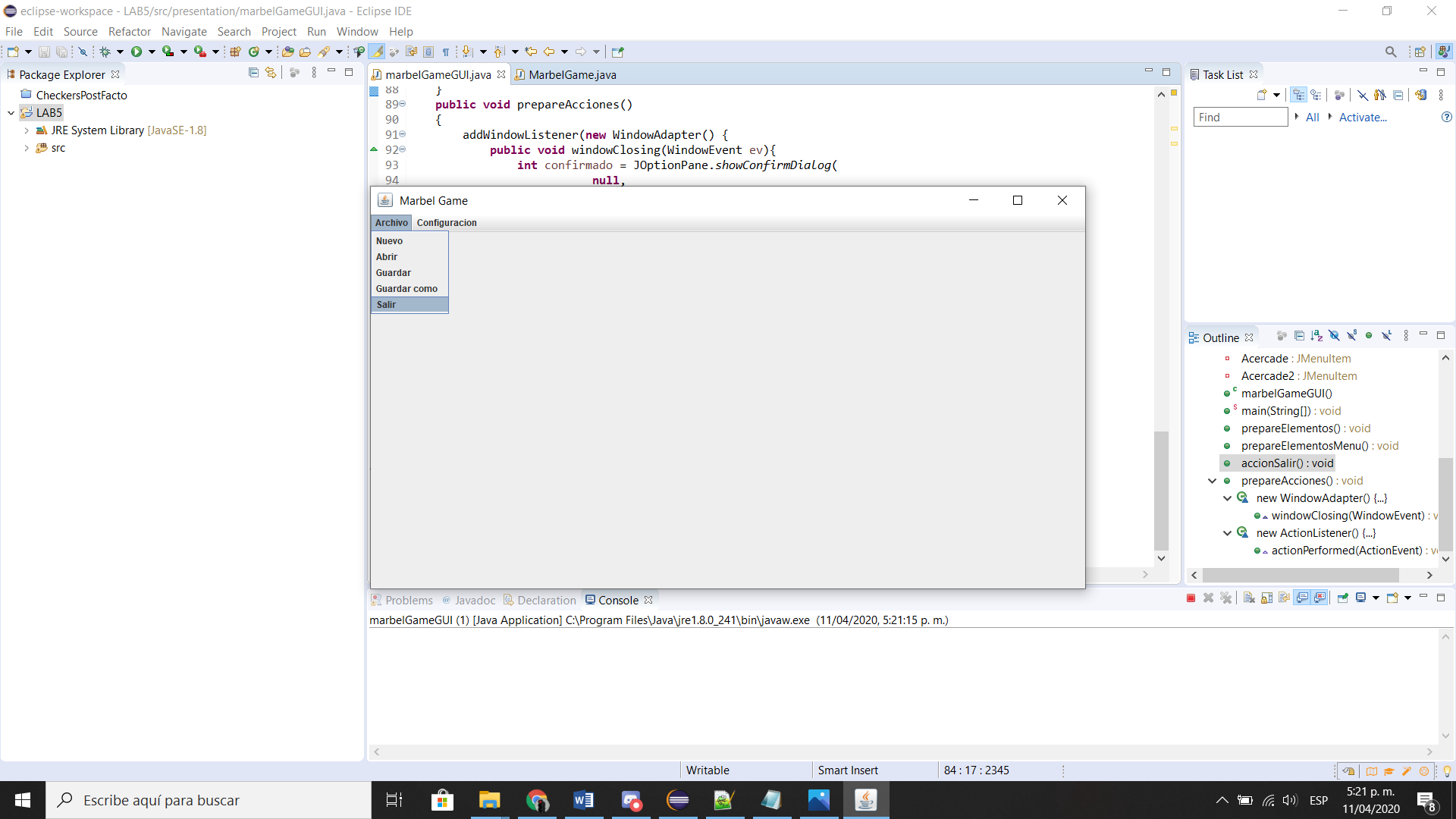


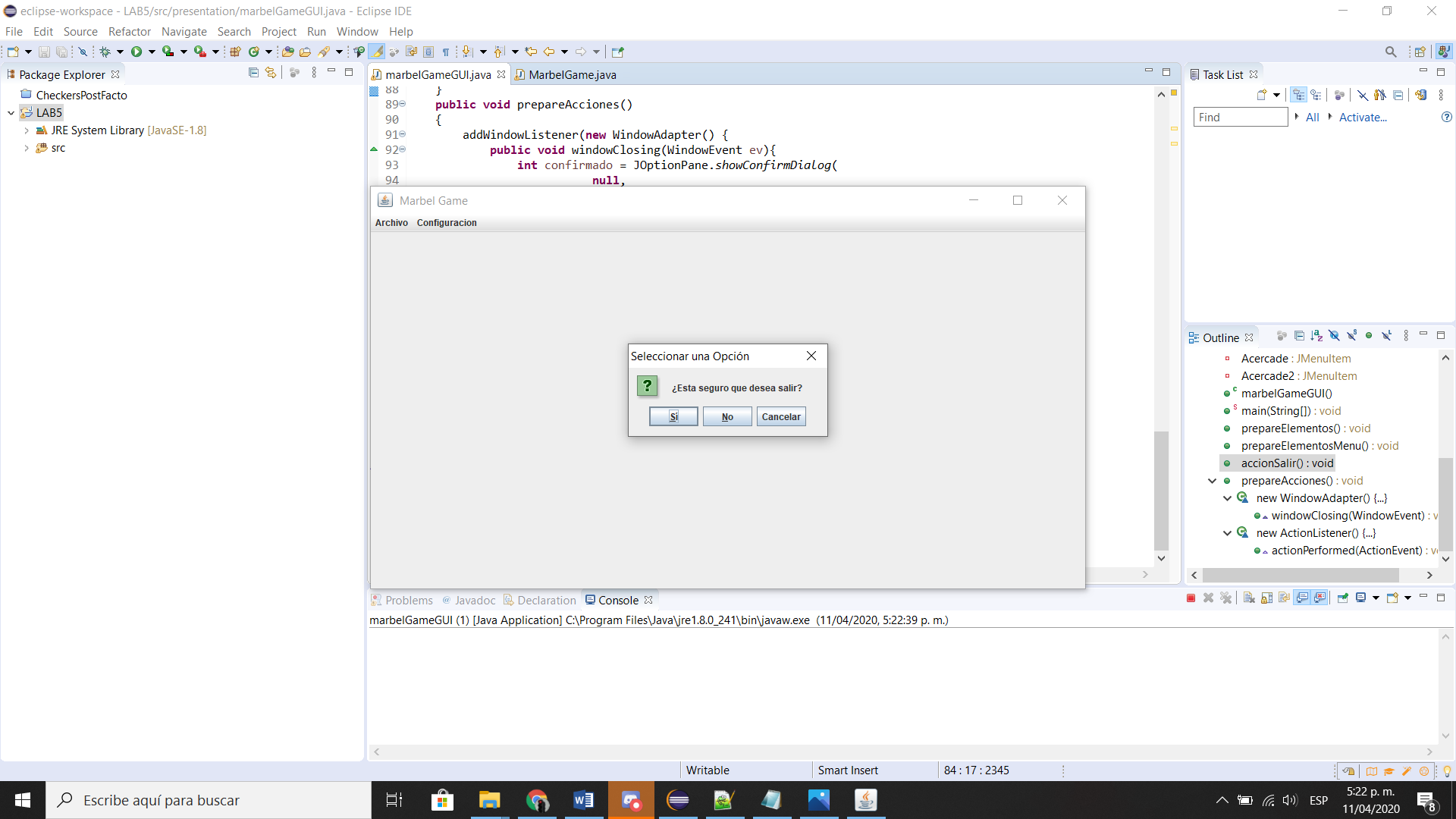


**3. Preparen el “oyente” correspondiente al icono cerrar con confirmación**

**(prepareAcciones). Ejecuten el programa y salgan del programa. Capturen las**

**pantallas.**





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.**

**JfileChooser:** es una clase java que nos permite mostrar fácilmente una ventana para la selección de un fichero.

**showOpenDialog:** La Clase JFileChooser nos permite cargar la ventana anterior usando su método showOpenDialog(this)

**showSaveDialog:** nos permite cargar la ventana de dialogo y encontrar la ruta donde almacenaremos un archivo

**getSelectedFile:** agregamos a un objeto de tipo File por medio del método getSelectedFile()

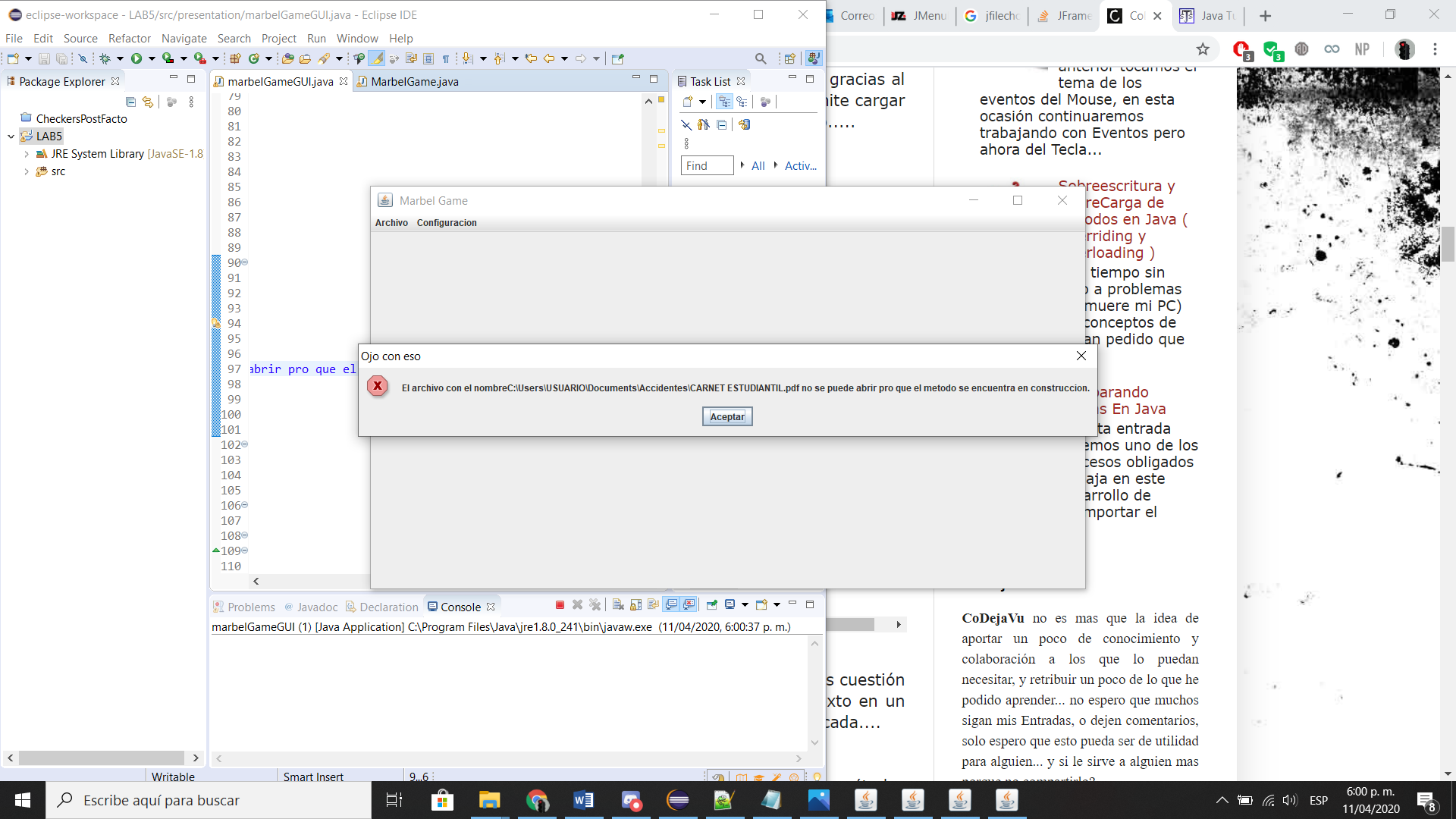
**2. Implementen parcialmente los elementos necesarios para salvar y abrir. Al seleccionar**

**los archivos indique que las funcionalidades están en construcción detallando la acción**

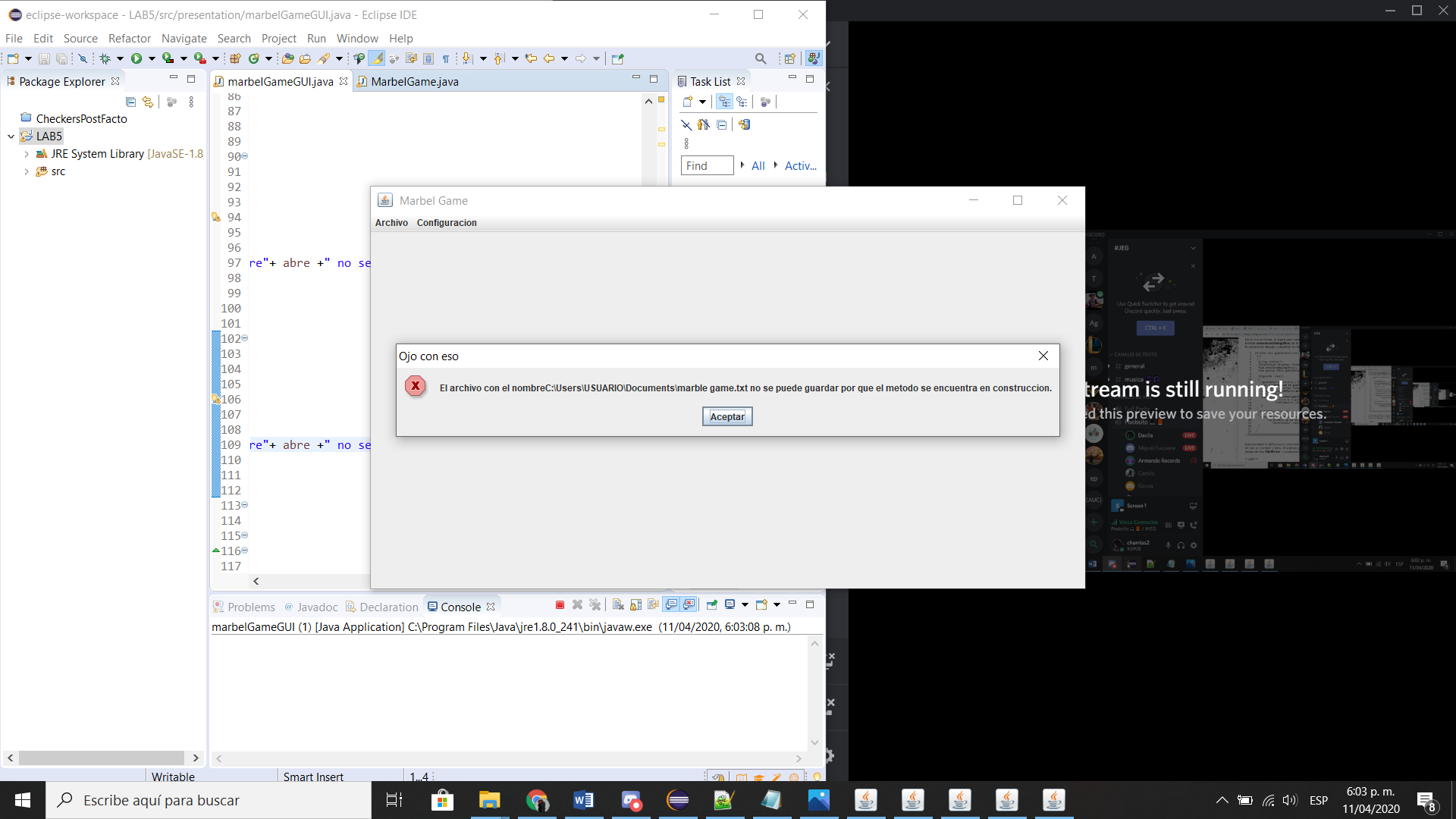
**y el nombre del archivo seleccionado.**

**3. Ejecuten las dos alternativas y capturen las pantallas más significativas.**

ABRIR



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.**

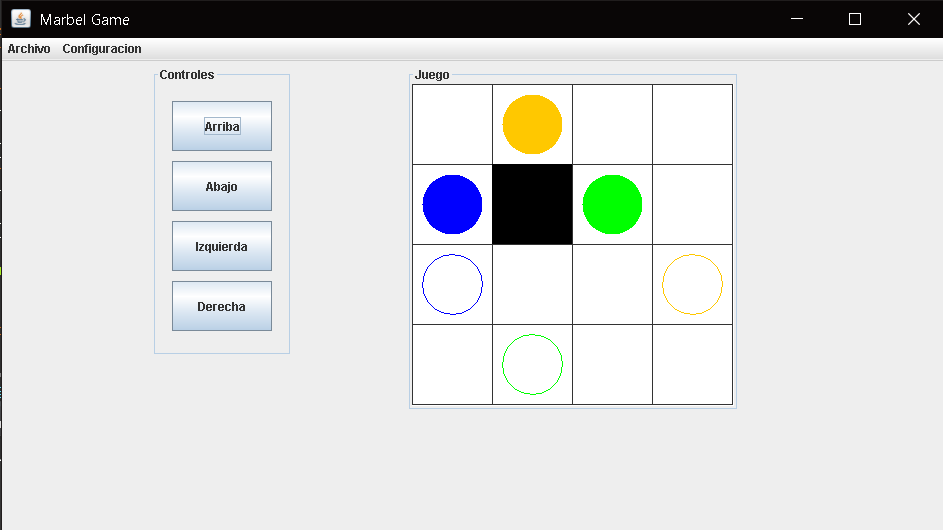
**2. Continúe con la implementación del método prepareElementos().**

**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.**

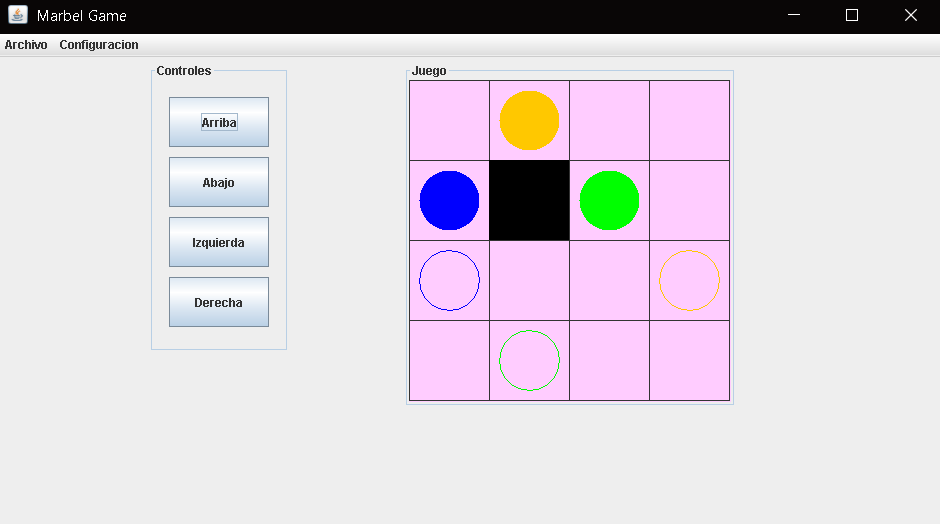
**2. Detalle el comportamiento de JColorChooser especialmente el método estático**

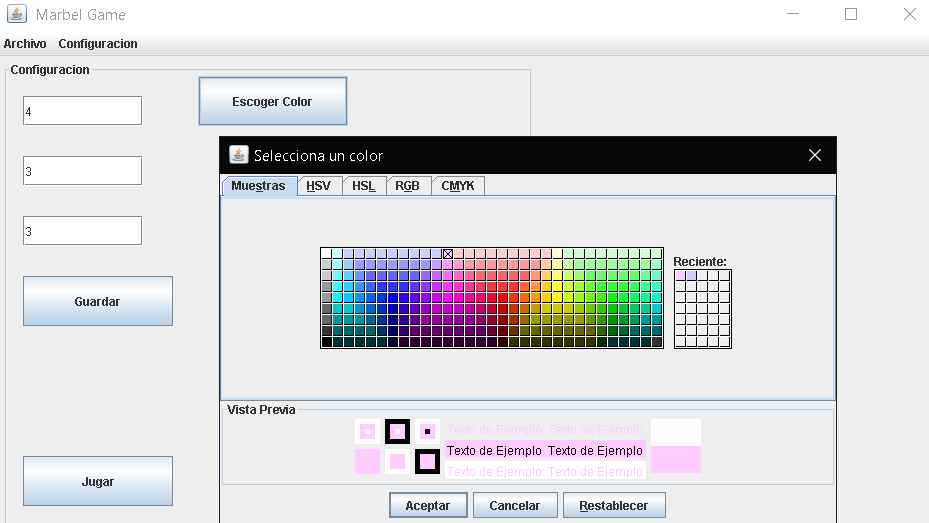
**showDialog**

**3. Implementen los componentes necesarios para cambiar el color del tablero**

**(inicialmente blanco-negro)**

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

****

****

**Ciclo 5: Modelo MarbelGame**

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

**El objetivo es implementar la capa de aplicación para MarbelGame .**

**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.**

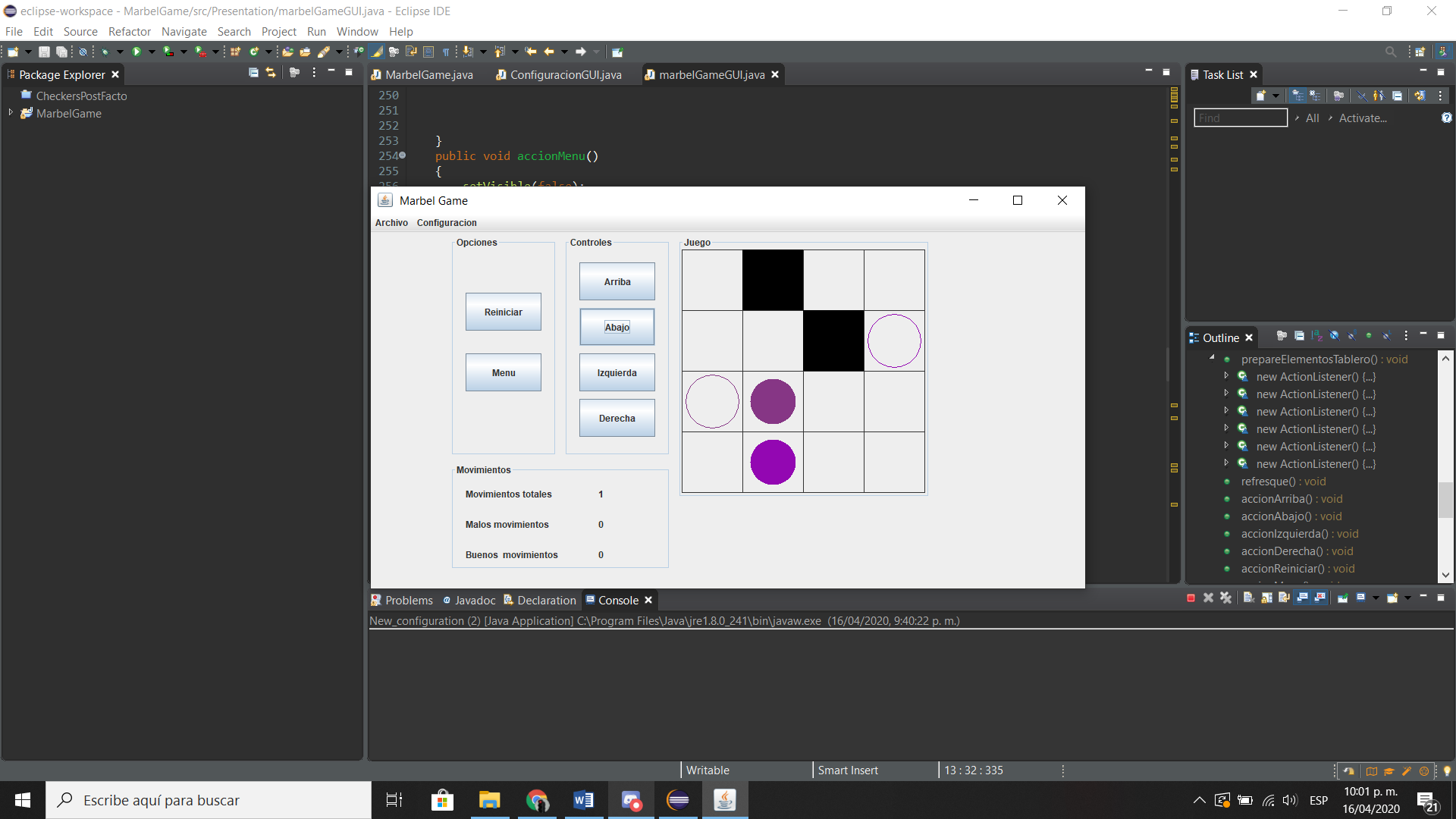
**2. Perfeccionen el método refresque() considerando la información del modelo de**

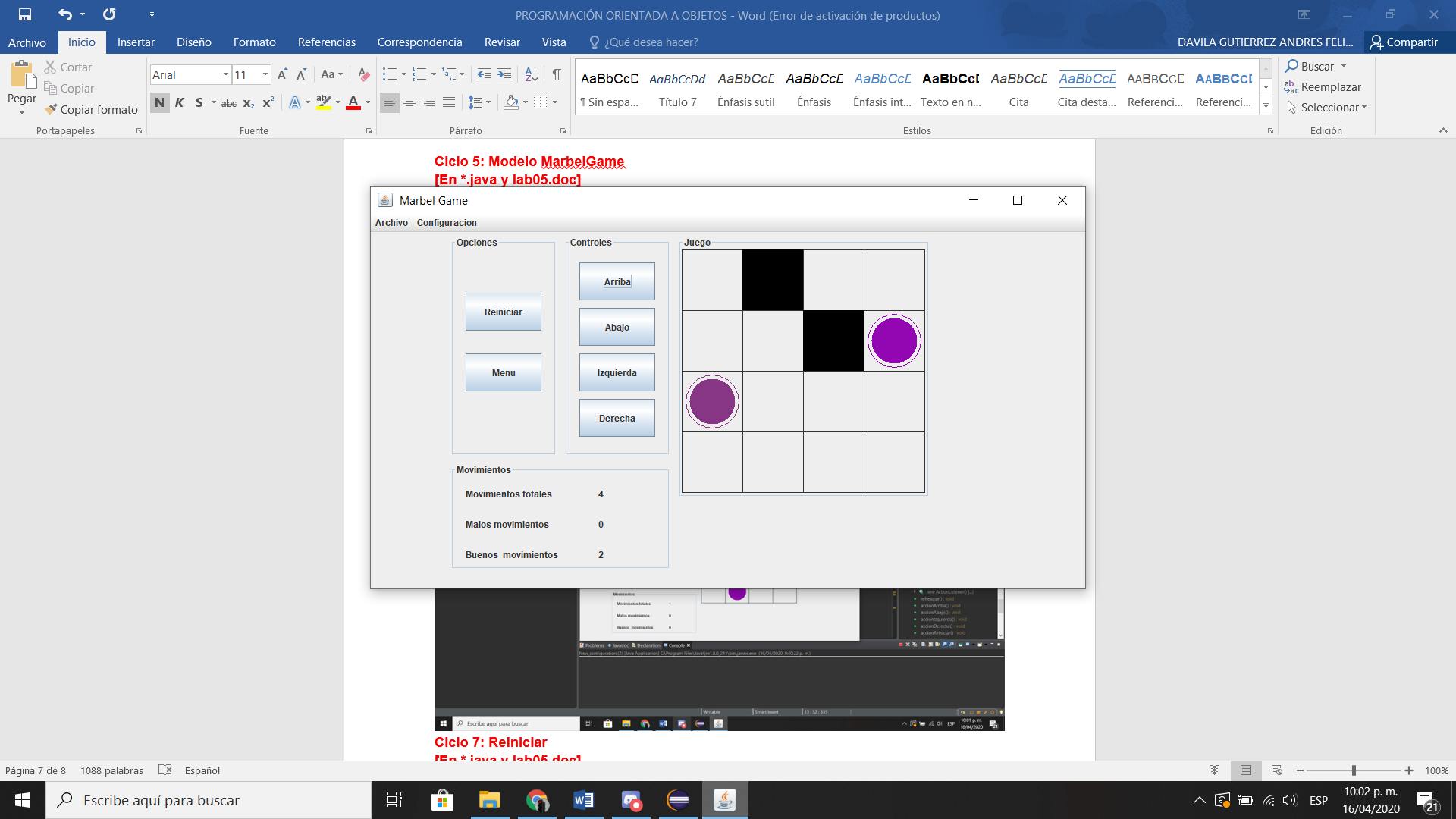
**dominio.**

**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**

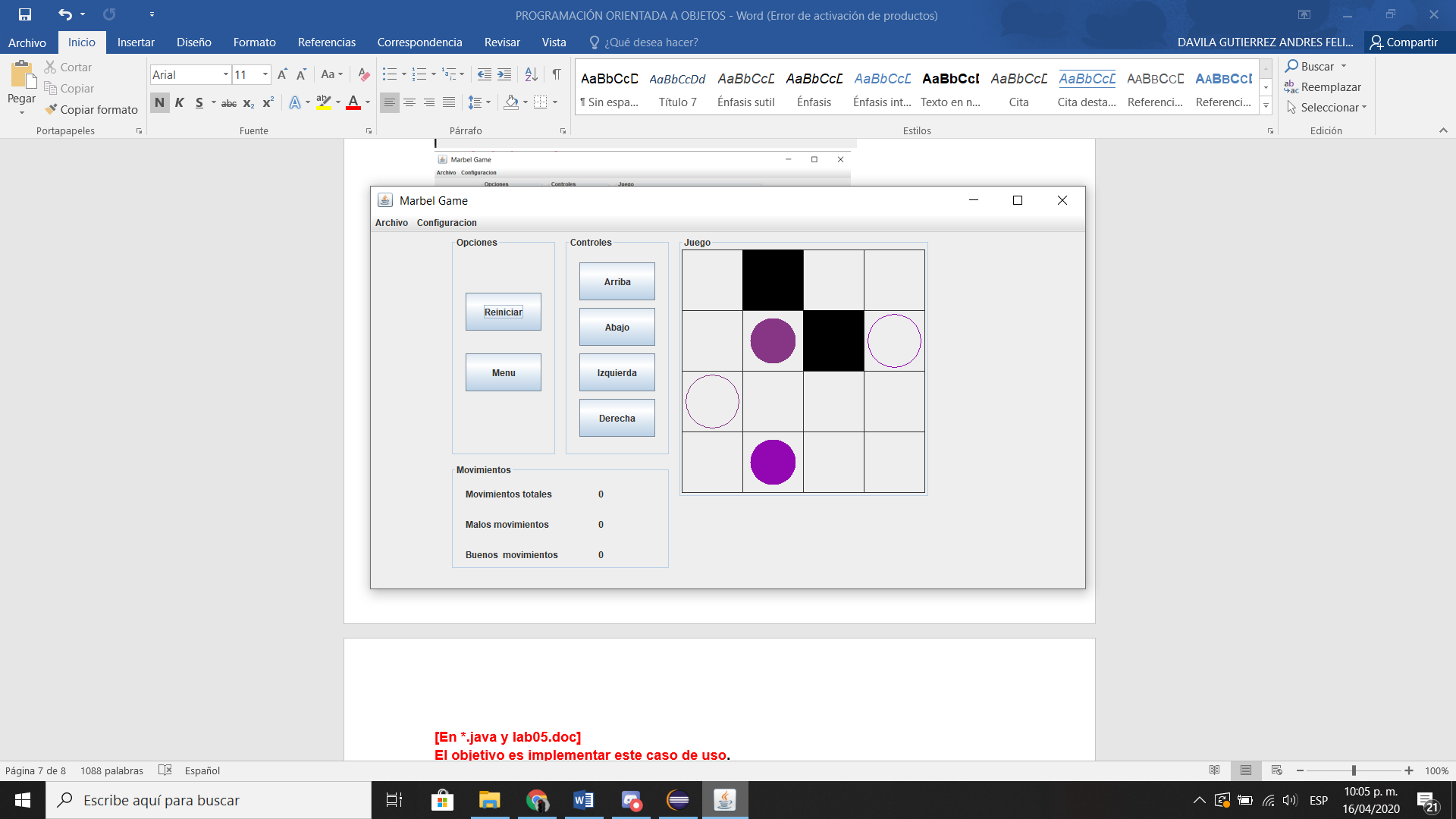
**[En \*.java y lab05.doc]**

**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 la configuración del juego**

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

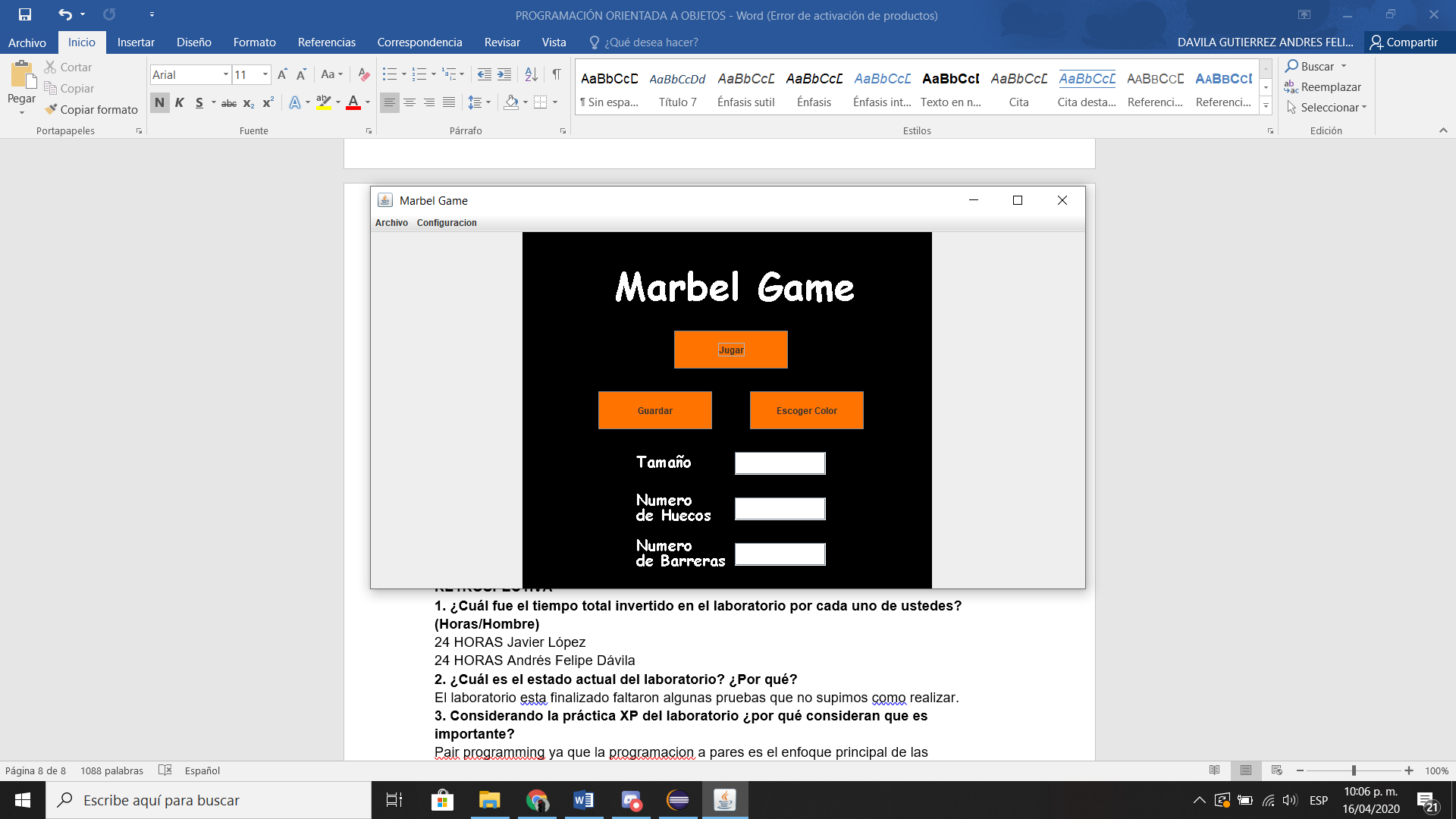
**El objetivo es implementar este caso de uso.**

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

**2. Implementen los elementos necesarios para cambiar la configuración del juego:**

**tamaño, número de canicas y número de barreras.**

**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)**

24 HORAS Javier López

24 HORAS Andrés Felipe Dávila

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

El laboratorio esta finalizado faltaron algunas pruebas que no supimos como realizar.

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

Pair programming ya que la programacion a pares es el enfoque principal de las programaciones que vamos a ver de ahora en adelante

**4. ¿Cuál consideran fue su mayor logro? ¿Por qué? ¿Cuál consideran que fue su mayor**

Entender todo el tema interfaces fue todo un reto pero fue algo que nos gusto bastante

**problema? ¿Qué hicieron para resolverlo?**

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

**resultados?**

Le dedicamos bastante tiempo al laboratorio y mejoramos en los temas de la documentacion