**PROGRAMACIÓN ORIENTADA A OBJETOS**

**INTERFAZ**

**2023-2**

CRISTIAN ALVAREZ – JULIANA BRICEÑO

**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, JScrollPane, 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. En el espacio de entrega deben indicar el estado de avance de su laboratorio y los problemas pendientes por resolver.
3. Deben publicar el avance al final de la sesión y la versión definitiva en la fecha indicada, en los espacios preparados para tal fin.

**CONTEXTO**

El objetivo es implementar el juego  **Vintage** en versión usuario-computador.

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

|  |  |
| --- | --- |
| El propuesto por ustedes  VintageGUI | El acordado en laboratorio Vintage |
| **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 **Vintage**.

1. Preparen un directorio llamado **Vintage**  con los directorios src y bin y los subdirectorios para presentación, dominio y pruebas de unidad.

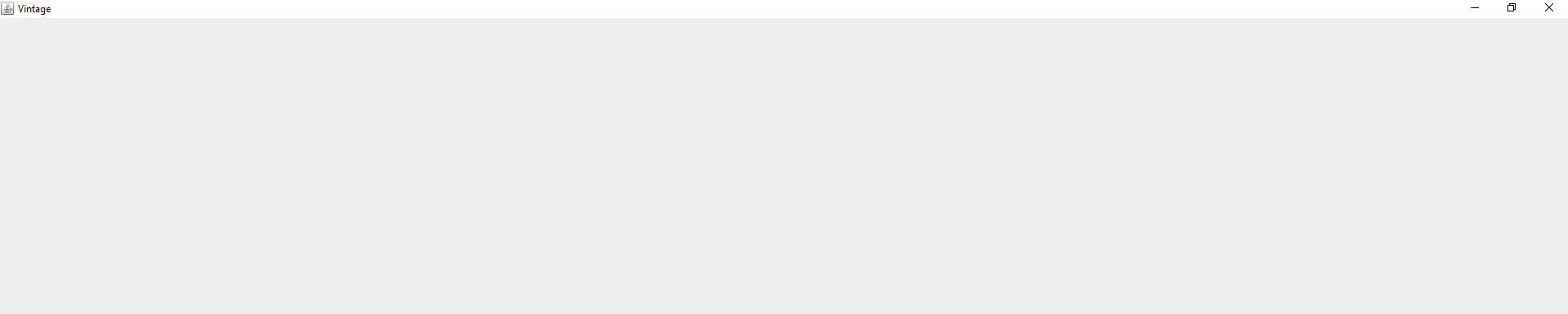
**Ciclo 0: Ventana vacía – Salir**

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

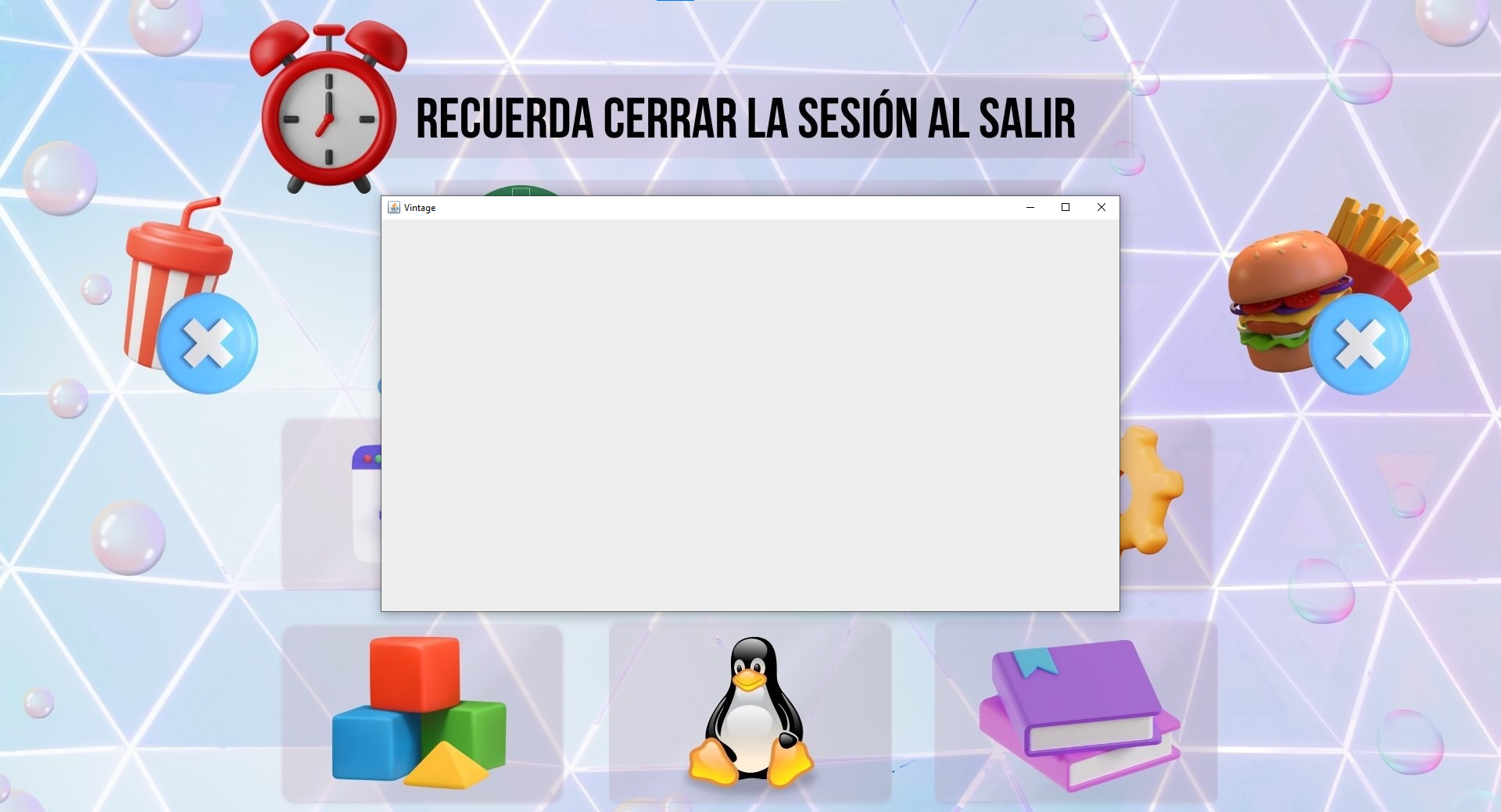
El objetivo es implementar la ventana principal de **Vintage** con un final adecuado desde el icono de cerrar. Utilizar el esquema de prepareElements-prepareActions.

1. Construyan el primer esquema de la ventana de **Vintage** únicamente con el título “**Vintage**”. Para esto cree la clase  **VintageGUI** como un JFrame con su creador (que sólo coloca el título) y el método main que crea un objeto **VintageGUI** 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.





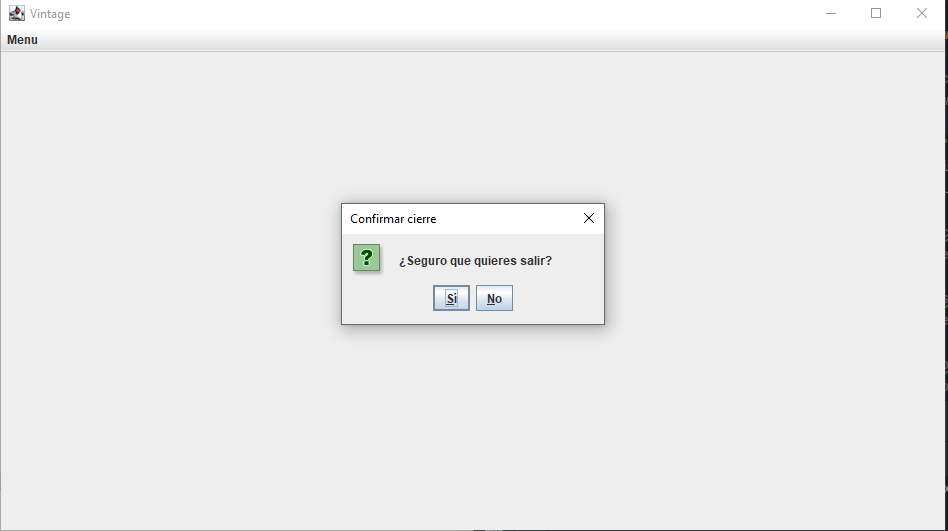
1. Traten de cerrar la ventana. ¿Termina la ejecución? ¿Qué deben hacer para terminar la ejecución?

R//: La ejecución no termina aunque se cierre la Ventana, debemos crear una accion en prepareAction para esto

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?

R//: Este método sirve para establecer la operación determinada cuando un usuario oprima el boton de la x. Lo usaríamos con DO\_NOTHING\_ON\_CLOSE y programaríamos un JOptionPane y en base a lo que responda el usuario se Cierra o no el programa.

1. 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**

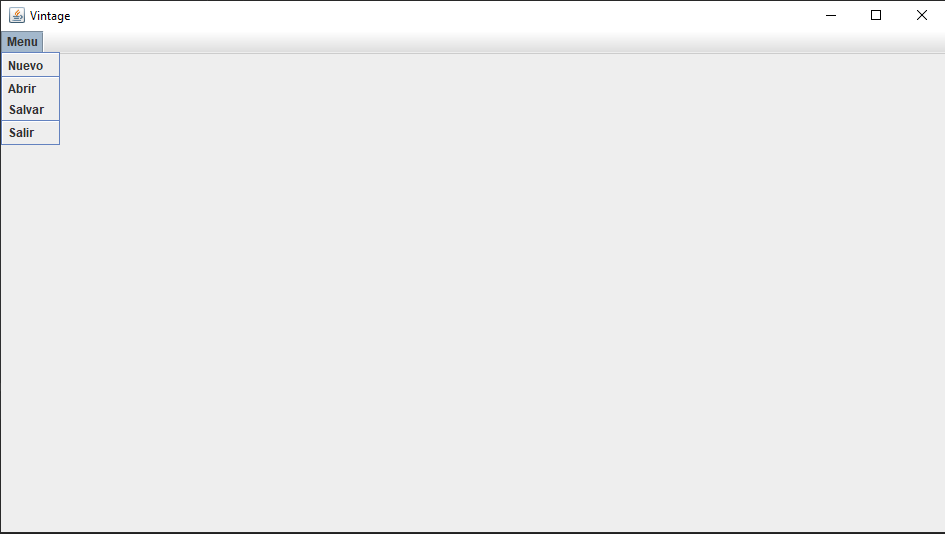
### [En \*.java y lab05.doc]

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.

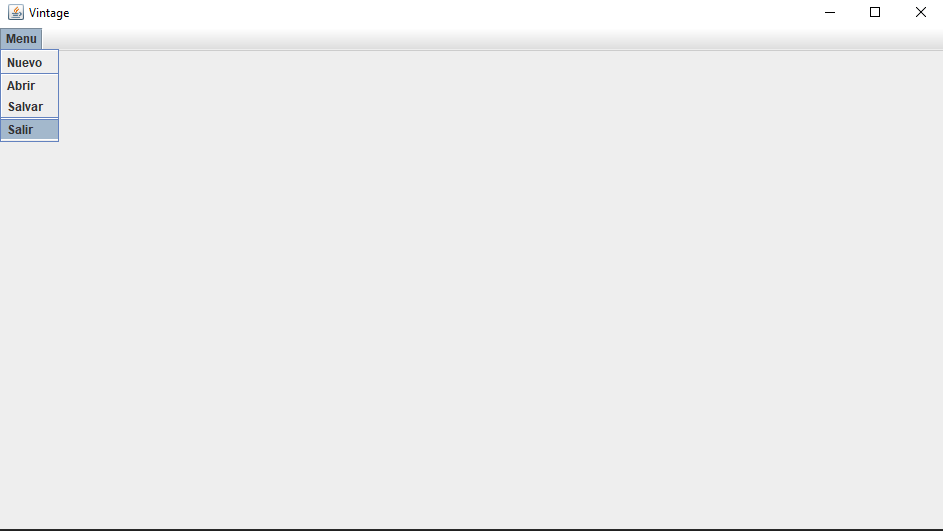
1. Expliquen los componentes visuales necesarios para este menú. ¿Cuáles serían atributos y cuáles podrían ser varibles del método prepareElements? Justifique.

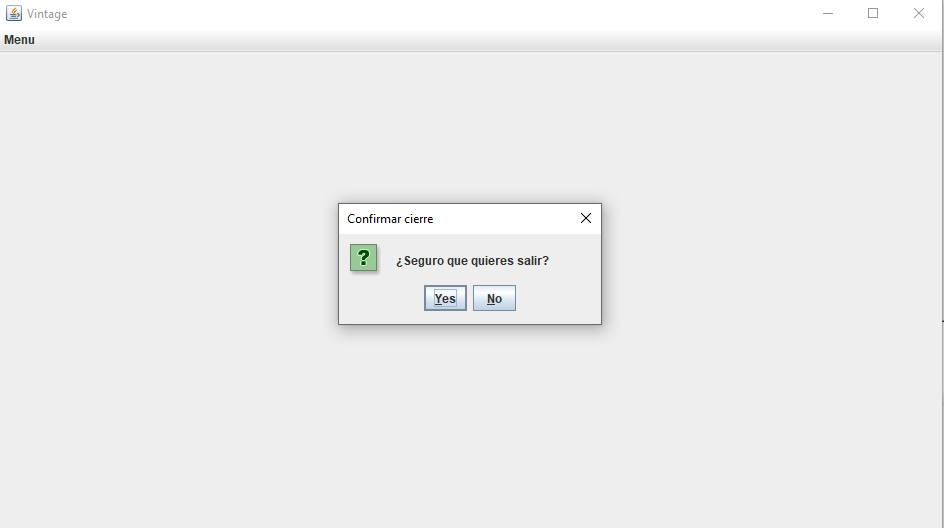
R//: Atributos serian en JMenuBar y el JMenu, como variables guardamos cada uno de los JMenuItem y los separadores JSeparator.

1. Construya la forma del menú propuesto (prepareElements - prepareElementsMenu) . Ejecuten. Capturen la pantalla.



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





**Ciclo 2: Salvar y abrir**

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

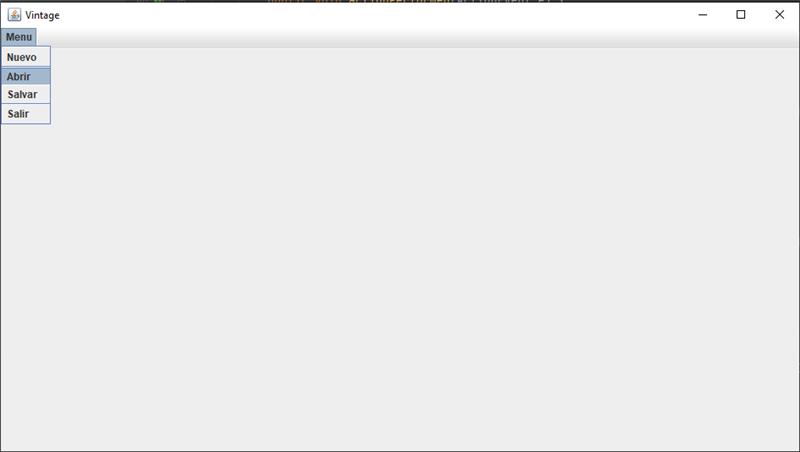
El objetivo es preparar la interfaz para las funciones de persistencia

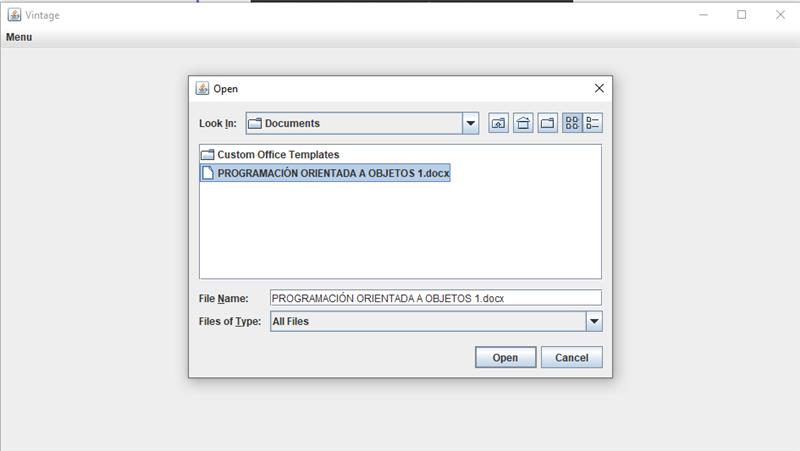
1. Detalle el componente JFileChooser especialmente los métodos : JFileChooser, showOpenDialog, showSaveDialog, getSelectedFile.

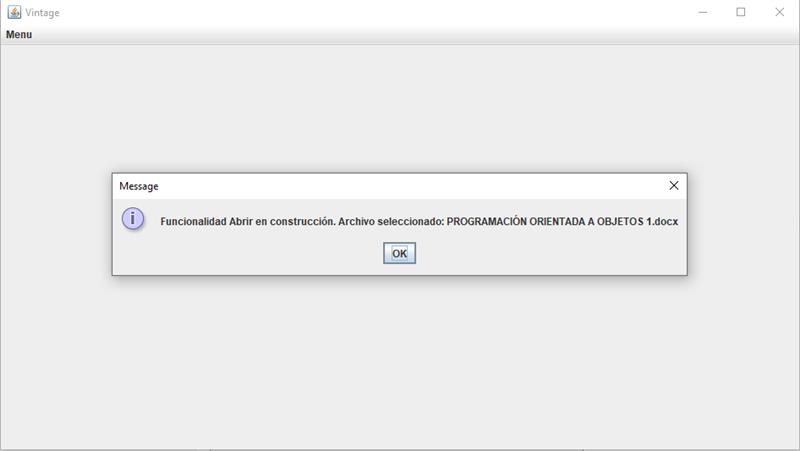
R//: JFileChooser proporciona un mecanismo para que el usuario elija un archivo, el constructor crea un fileChooser que apunta a un directorio predeterminado. El showOpenDialog muestra un cuadro de dialogo de apertura de archivos. ShowSaveDialog muestra un cuadro de dialogo de guardar un archivo y getSelectFile devuelve el archivo que el usuario a seleccionado en el cuadro de dialogo.

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 opciones y capturen las pantallas más significativas.

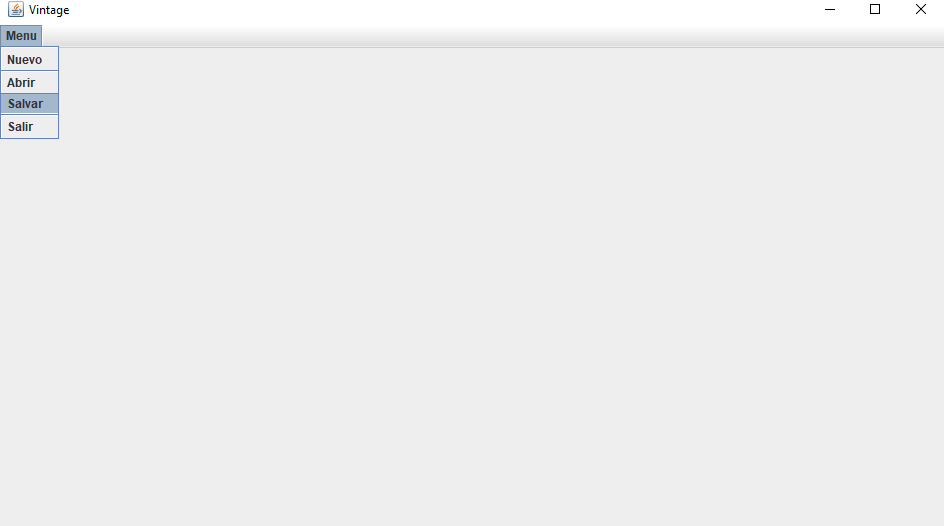
Abrir:

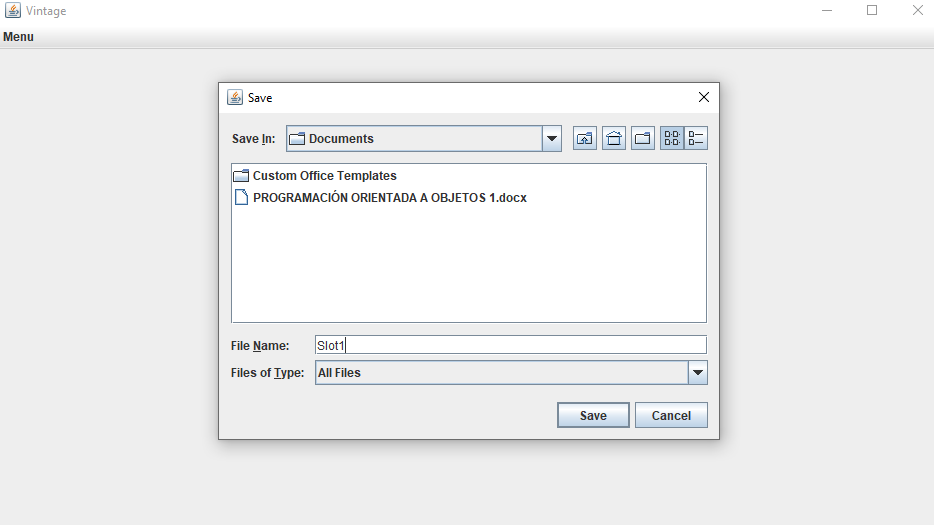


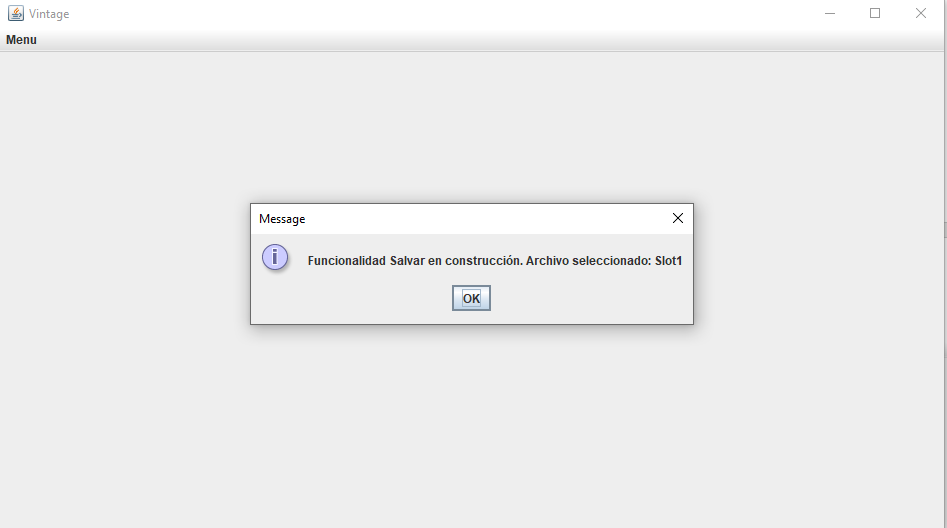




Salvar:







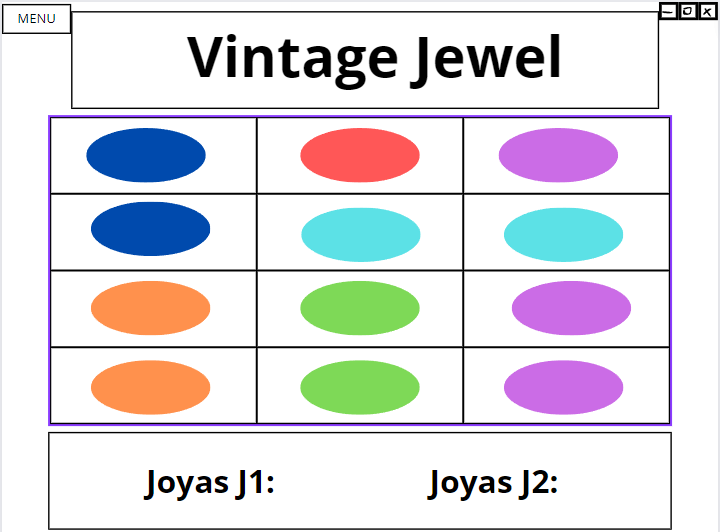
**Ciclo 3: Forma de la ventana principal**

### [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.

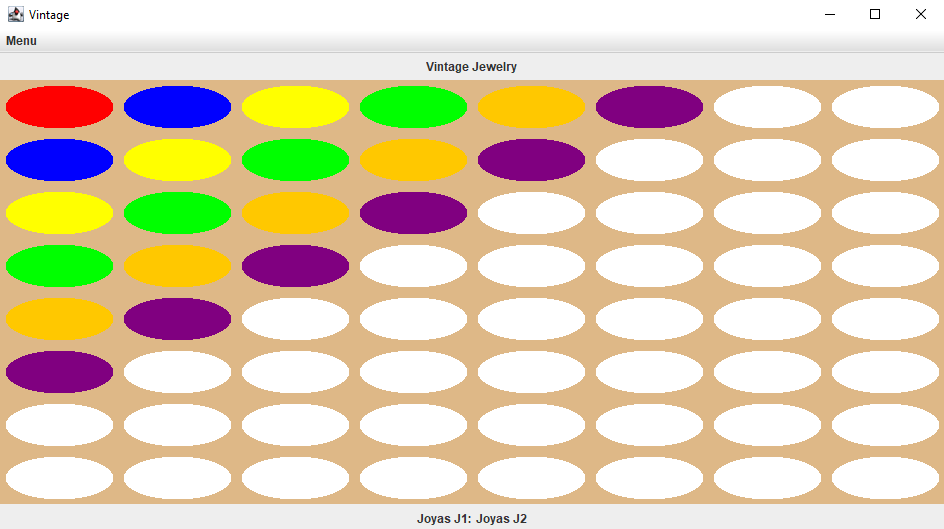
Para este primer bosquejo necesitamos de 3 JPanel, con una distribución de borderLayaot. El primer panel se ubicaría al norte y contaria con un JLabel que muestra el nombre de la aplicación, el segundo iría en el centro y tendría todo lo referente al tablero y el ultimo se ubicaría en el sur y tendría las puntuaciones de los jugadores.



1. Continúe con la implementación definiendo los atributos necesarios y extendiendo el método prepareElements().

Para la zona del tablero defina 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 esta pantalla.



**Ciclo 4: Cambiar color**

### [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 la interfaz grafica GUI, el botón que permita interactuar con el usuario para que elija el color y los eventos listener para manejar las interacciones del usuario y que implicaciones tiene en la interfaz grafica.

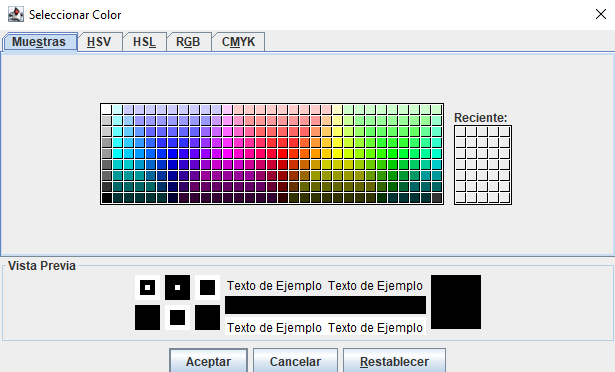
En el controlador tenemos que recibir ese evento de la vista y mediante métodos asignarlo a al color de la joya correspondiente.

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

R//:

JColorChooser es una clase en la biblioteca Swing de Java que proporciona una interfaz gráfica para que los usuarios elijan un color. El método estático showDialog se utiliza para mostrar un diálogo de selección de color.

1. Implementen los componentes necesarios para cambiar el color del tablero y de las fichas.
2. Ejecuten el caso de uso y capture las pantallas más significativas.

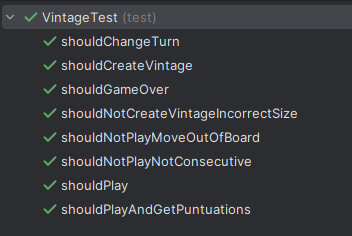


**Ciclo 5: Modelo Vintage**

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

El objetivo es implementar la capa de dominio para **Vintage**.

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.

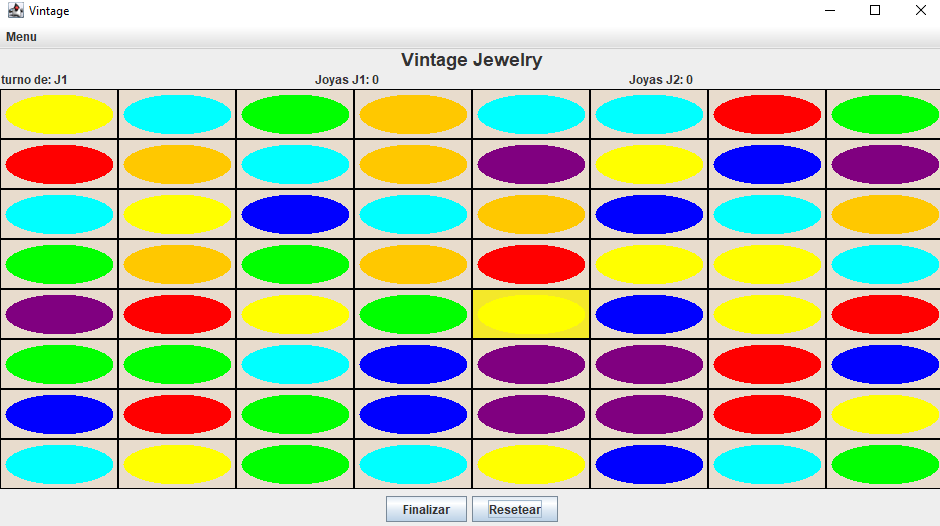
1. Adicione a la capa de presentación el atributo correspondiente al modelo.
2. Perfeccionen el método refresh() considerando la información del modelo de dominio.
3. Expliquen los elementos necesarios para implementar este caso de uso.

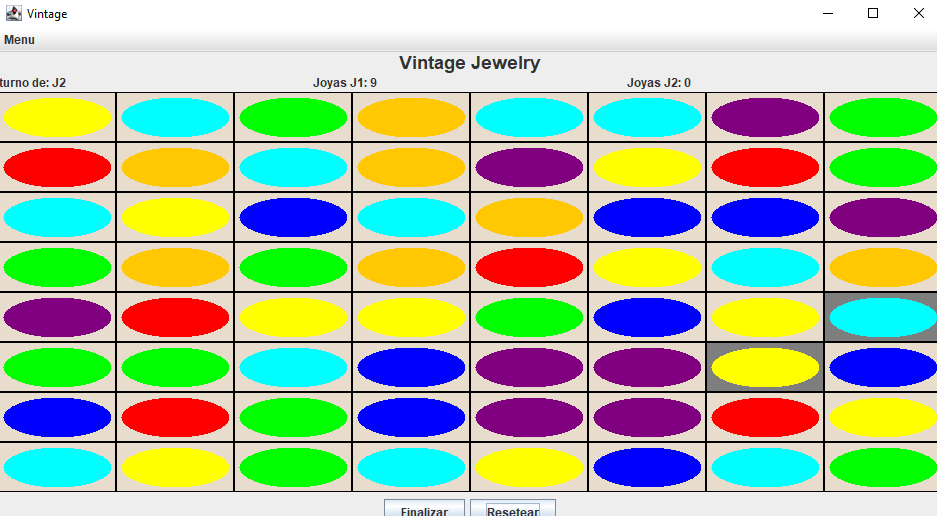
R//: Para realizar el refresh fue necesario solicitarle a vintage que nos de la matriz que nosotros interpretamos para pintar el tablero, luego recorremos los paneles con los nuevos componentes y repintamos el tablero.

1. Implementen los componentes necesarios para jugar .¿Cuántos oyentes necesitan?

Se necesitan dos oyentes, se realizan los dos clicks para elegir cuales casillas se van a intercambiar, no es necesario saber que usuario esta jugando porque el sistema cambia automáticamente de turno

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





**Ciclo 7: Reiniciar**

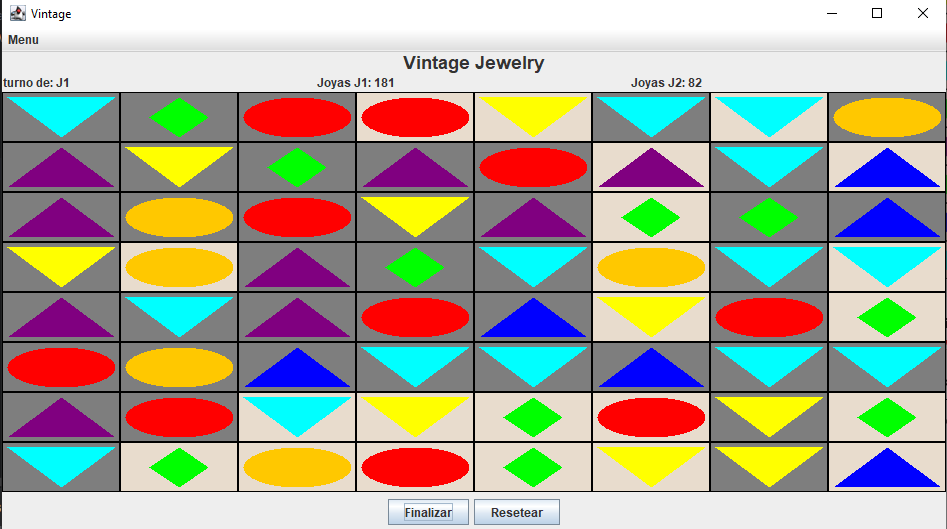
### [En \*.java y lab05.doc]

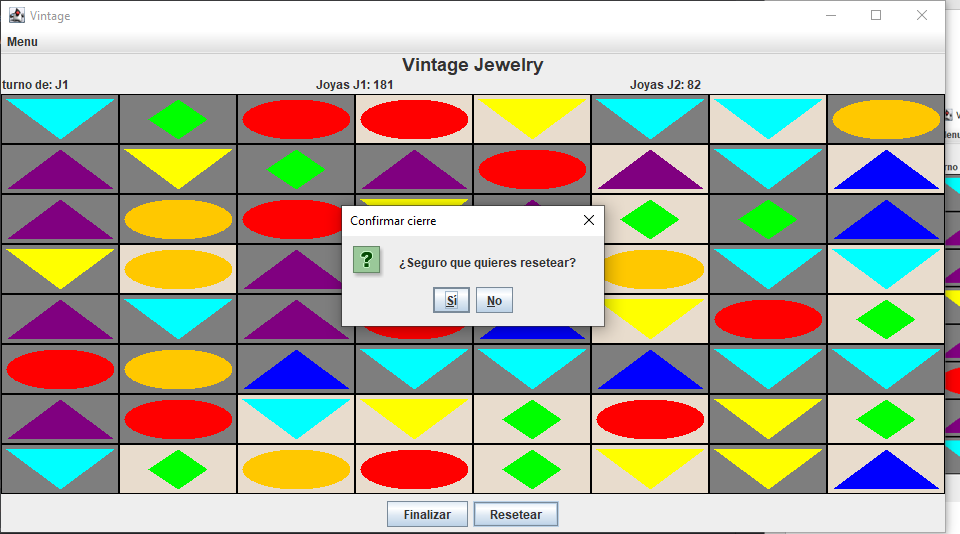
El objetivo es implementar este caso de uso.

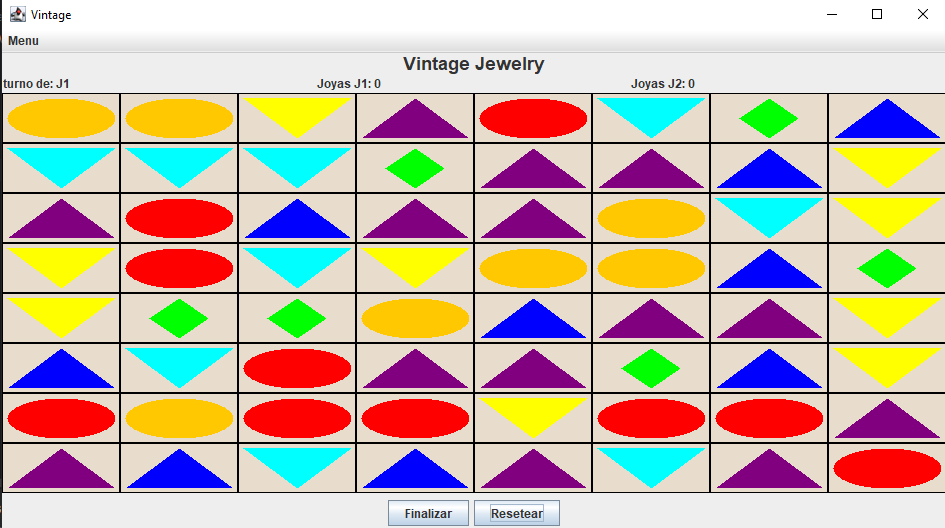
1. Expliquen los elementos a usar para implementar este caso de uso.

R//: Necesitamos un botón para realizar la opción de reiniciar, mandamos un mensaje de corfimacion que se quiera resetear, luego solo se pone como null los atributos de vintage y se crea nuevamente el vintage, por ultimo se refresca la pantalla.

1. Implementen los elementos necesarios para reiniciar
2. Ejecuten el caso de uso y capture las pantallas más significativas.







**Ciclo 8: Cambiar el tamaño**

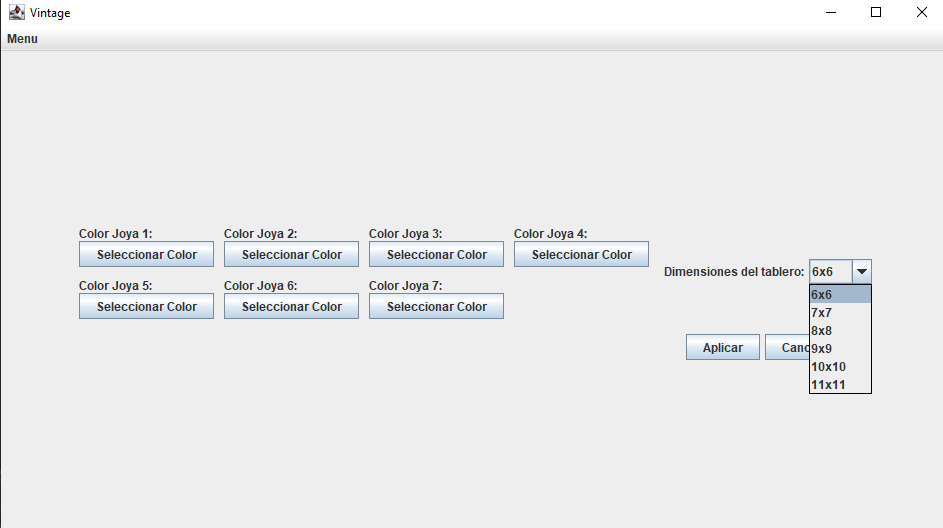
### [En \*.java y lab05.doc]

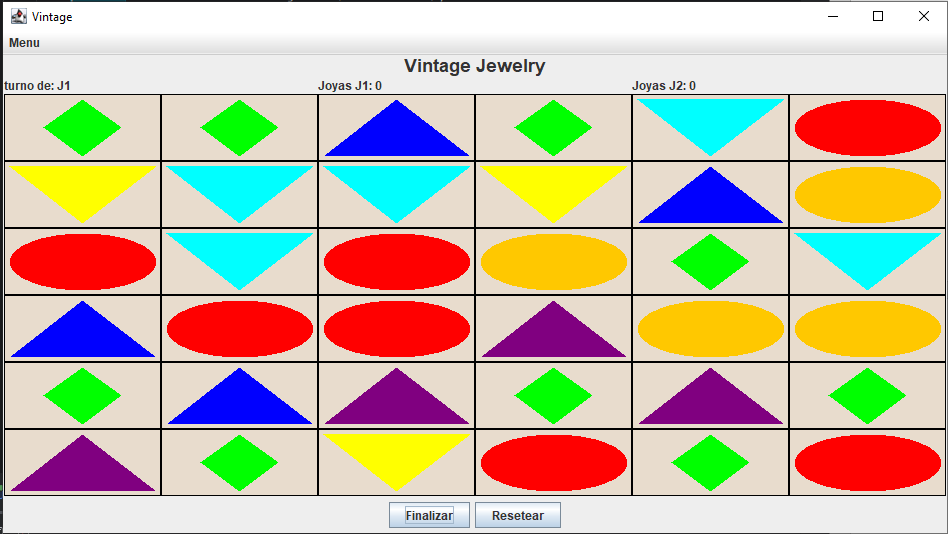
El objetivo es implementar este caso de uso.

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

R//: Nosotros decidimos que el usuario puede modificar el tamaño desde una lista de opciones, por lo cual es necesario usar un desplegable con los posibles tamaños, esta elección se le asigna al atributo de row y columna que se usan para instanciar el vintage

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

R//:(Cristian Javier Alvarez Baquero/35), (Juliana Briceño/ 35)

1. ¿Cuál es el estado actual de laboratorio? ¿Por qué? (Para cada método incluya su estado)

R//: Abarcamos todos los puntos y los desarrollamos en su totalidad.

1. Considerando las prácticas XP del laboratorio de hoy ¿por qué consideran que son importe?

R//: All code must have unit tests, ya que aseguran la calidad del codigo al detectar errores a medida que se codifica.

1. ¿Cuál consideran fue su mayor logro? ¿Por qué? ¿Cuál consideran que fue su mayor problema? ¿Qué hicieron para resolverlo?

R//: Haber logrado que los métodos funcionaran

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

R//: Trabajamos a la par, resolvimos dudas entre nosotros y buscamos ayuda cuando lo llegamos a necesitar. Nos comprometemos a seguir trabajando en equipo.