# PROGRAMACIÓN ORIENTADA A OBJETOS INTERFAZ

**2021-2**

# Laboratorio 5/6

## OBJETIVOS

1. Desarrollar una mini aplicación gráfica
2. Experimentar el comportamiento de las ventanas JFrame, JDialog y JOptionPane
3. Seleccionar los lienzos más apropiados para un diseño: JPanel, JScrollPane, JTabbedPane
4. Revisar las posibilidades de los estilos:FlowLayout, BorderLayout y GridLayout
5. Ensayar algunos componentes básicos: JLabel, JTextField, JButton, JMenuBar,
6. Ensayar algunos componentes especiales: JFileChooser y JColorChooser
7. Implementar el esquema de manejo de eventos con clases anónimas
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 foro 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 **Senku** en versión usuario-computador.

El trabajo se debe hacer desde **CONSOLA**

|  |  |
| --- | --- |
| El propuesto por ustedes Senku | El acordado en laboratorio Senku |
| **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 **Senku**.

1. Preparen un directorio llamado **Senku** con los directorios src y bin y los subdirectorios presentación y dominio.

## Ciclo 0: Ventana vacía – Salir [En \*.java y lab05.doc]

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 **Senku** como un JFrame con su creador (que sólo coloca el título) y el método main que crea un objeto **Senku** 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)

Texto

Descripción generada automáticamente

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

Texto

Descripción generada automáticamente

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

Se cierra la ventana, aunque sigue en ejecución, la clase de SenkuGUI.

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

El setDefaultCloseOperation() es un método se utiliza para especificar una de las opciones que se t tiene para el botón de cierre.

Dentro de estas se encuentran:

JFrame.EXIT\_ON\_CLOSE: salir de la aplicación.

JFrame.HIDE\_ON\_CLOSE: oculta el marco, pero mantén la aplicación en ejecución.

JFrame.DO\_NOTHING\_ON\_CLOSE: ignora el clic.

JFrame.DISPOSE\_ON\_CLOSE - Elimina el objeto de marco, pero mantén la aplicación en ejecución.

Para este caso usaremos DO\_NOTHING\_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 cierren el programa. Capturen las pantallas.

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

## 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. 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 (prepareElementos - prepareElementosMenu) . Ejecuten. Capturen la pantalla.

Un conjunto de letras blancas en un fondo blanco

Descripción generada automáticamente con confianza media

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

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

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

## 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 de java para mostrar fácilmente una ventana para la selección de un fichero

ShowOpenDialog, ventana para abrir un archivo.

ShowSaveDialog, ventana para salvar el archivo.

getSelectedFile, selecciona el archivo dado.

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

los archivos indiquen que las funcionalidades están en construcción detallando la acción y el nombre del archivo seleccionado.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Ejecuten las dos opciones y capturen las pantallas más significativas.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

## 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 (vista – controlador) necesarios para implementar este caso de uso.
  2. Detalle el comportamiento de JColorChooser especialmente el método estático

showDialog

JColorChooser es una herramienta que presenta una ventana de varios colores, en sí es un JDialog, lo cual es un panel en la cual el usuario puede interactuar con cada uno de los colores, con respecto al comportamiento de showDialog, este método, carga cierto color definido en JDialog, para asignarlo a nuestra ventana.

* 1. Implementen los componentes necesarios para cambiar el color del tablero (inicialmente blanco-negro)
  2. Ejecuten el caso de uso y capture las pantallas más significativas.

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza baja

## Ciclo 5: Modelo Senku [En \*.java y lab05.doc]

El objetivo es implementar la capa de dominio 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

**[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
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?