Sobre los Componentes SWING

1. Label + Separator
Uso:
Label: Se utiliza para mostrar texto o imágenes en la interfaz, proporcionando información o instrucciones al usuario.
Separator: Sirve para dividir secciones en un formulario o ventana, mejorando la organización visual.

Propiedades Principales:
Label:
text: Texto que se mostrará.
icon: Imagen asociada al label.

Separator:
orientation: Define si el separador es horizontal o vertical.

Métodos:
Label:
setText(String text): Cambia el texto del label.
setIcon(Icon icon): Establece un ícono.

Separator:
setOrientation(int orientation): Cambia la orientación del separador.

Eventos:
No suele tener eventos.

2. Button + Toggle Button
Uso:
Button: Un botón que ejecuta una acción cuando se presiona.
Toggle Button: Un botón que puede estar en un estado activado o desactivado.

Propiedades Principales:
Button:
text: Texto visible en el botón.
actionCommand: Comando asociado a la acción.

Toggle Button:
selected: Indica si el botón está seleccionado.

Métodos:
Button:
addActionListener(ActionListener l): Agrega un oyente para la acción.

Toggle Button:
setSelected(boolean selected): Establece el estado del botón.

Eventos:
ActionEvent: Se genera cuando el botón es presionado.

3. Check Box
Uso:
Se utiliza para permitir al usuario seleccionar múltiples opciones de una lista.

Propiedades Principales:
selected: Indica si el checkbox está seleccionado.
text: Texto asociado al checkbox.

Métodos:
addItemListener(ItemListener l): Agrega un oyente para cambios en el estado.

Eventos:
ItemEvent: Se genera cuando el estado del checkbox cambia.

4. Radio Button + Button Group
Uso:
Radio Button: Permite seleccionar solo una opción de un grupo.
Button Group: Agrupa varios radio buttons para asegurar que solo uno esté seleccionado.

Propiedades Principales:
selected: Indica si el radio button está seleccionado.

Métodos:
addItemListener(ItemListener l): Agrega un oyente para cambios en el estado.

Eventos:
ItemEvent: Se genera cuando se selecciona un radio button.

5. Combo Box
Uso:
Permite seleccionar una opción de una lista desplegable.

Propiedades Principales:
selectedItem: Elemento actualmente seleccionado.

Métodos:
addActionListener(ActionListener l): Agrega un oyente para la acción.

Eventos:
ActionEvent: Se genera cuando se selecciona un elemento.

6. Spinner
Uso:
Permite al usuario seleccionar un valor de un rango determinado, utilizando botones de incremento y decremento.

Propiedades Principales:
value: Valor actualmente seleccionado.
model: Modelo que define el rango de valores.

Métodos:
addChangeListener(ChangeListener l): Agrega un oyente para cambios en el valor.

Eventos:
ChangeEvent: Se genera cuando el valor cambia.

7. List
Uso:
Muestra una lista de elementos donde el usuario puede seleccionar uno o varios.

Propiedades Principales:
selectedValuesList(): Devuelve la lista de elementos seleccionados.

Métodos:
addListSelectionListener(ListSelectionListener l): Agrega un oyente para cambios en la selección.

Eventos:
ListSelectionEvent: Se genera cuando la selección cambia.

8. Text Field
Uso:
Permite al usuario ingresar texto en una línea.

Propiedades Principales:
text: Texto actualmente en el campo.

Métodos:
addActionListener(ActionListener l): Agrega un oyente para la acción.

Eventos:
ActionEvent: Se genera cuando se presiona "Enter".

9. Formatted Field + Password Field
Uso:
Formatted Field: Permite entrada de texto con un formato específico.
Password Field: Oculta el texto ingresado, usado para contraseñas.

Propiedades Principales:
text: Texto ingresado.

Métodos:
Formatted Field:
setFormatterFactory(FormatterFactory factory): Establece el formato.
Password Field:
setEchoChar(char echoChar): Define el carácter que oculta la entrada.

Eventos:
ActionEvent: Se genera al presionar "Enter".

10. Text Area
Uso:
Permite la entrada de texto en múltiples líneas.

Propiedades Principales:
text: Texto en el área.
lineWrap: Define si el texto se ajusta a la línea.

Métodos:
addKeyListener(KeyListener l): Agrega un oyente para eventos de teclado.

Eventos:
KeyEvent: Se genera al presionar una tecla.

11. Scroll Bar + Slider
Uso:
Scroll Bar: Permite desplazarse a través de un contenido extenso.
Slider: Permite seleccionar un valor de un rango con un control deslizante.

Propiedades Principales:
value: Valor actualmente seleccionado.

Métodos:
addChangeListener(ChangeListener l): Agrega un oyente para cambios en el valor.

Eventos:
ChangeEvent: Se genera cuando el valor cambia.

12. Progress Bar
Uso:
Muestra el progreso de una tarea en ejecución.

Propiedades Principales:
value: Valor actual del progreso.

Métodos:
setIndeterminate(boolean indeterminate): Establece si es indeterminado.

Eventos:
No suele tener eventos asociados.

13. Text Pane
Uso:
Permite mostrar texto con diferentes estilos y formatos.

Propiedades Principales:
text: Texto en el pane.

Métodos:
setStyledDocument(StyledDocument doc): Establece el documento estilizado.

Eventos:
No suele tener eventos asociados directamente.

14. Editor Pane
Uso:
Permite mostrar contenido HTML o texto enriquecido.

Propiedades Principales:
contentType: Tipo de contenido (HTML, texto plano).

Métodos:
setPage(URL url): Carga una página web.

Eventos:
No suele tener eventos asociados.

15. Tree + Table
Uso:
Tree: Muestra datos en forma jerárquica.
Table: Muestra datos en formato de cuadrícula.

Propiedades Principales:
Tree:
model: Modelo que representa la estructura del árbol.

Table:
model: Modelo que contiene los datos de la tabla.

Métodos:
Tree:
addTreeSelectionListener(TreeSelectionListener l): Agrega un oyente para selección.

Table:
addTableModelListener(TableModelListener l): Agrega un oyente para cambios en el modelo.

Eventos:
Tree: TreeSelectionEvent: Se genera cuando cambia la selección en un componente de tipo árbol (JTree). Este evento se dispara cada vez que un nodo del árbol es seleccionado o deseleccionado.
Table: TableModelEvent: Se genera en un componente de tabla (JTable) cuando se produce un cambio en el modelo de datos subyacente, como la adición, eliminación o modificación de datos en la tabla.