CLASES:

Los componentes swing son objetos de clases derivadas de la clase **JComponent** que deriva de la clase **java.awt.Component**. Para crear interfaces gráficas de usuario, swing combina los componentes de la clase JComponent en contenedores de nivel alto JWindow, JFrame, JDialog y JApplet.

* **JWindow** es una ventana sin barra de título y sin botones.
* **JFrame** es una ventana con barra de titulo y con los botones que permiten su manipulación.
* **JDialog** permite visualizar una cuadro de diálogo.
* **JApplet** permite crear un **applet swing**.

Para dar funcionalidad a las ventanas, la swing proporciona un conjunto de componentes que derivan de la clase JComponent, los más utilizados son:

* **JComponent**: Clase base para los componentes swing.
  + **AbstractButton**: Define el comportamiento común de los botones y los menús.
    - **JButton**. Botón.
    - **JMenuItem**. Elemento de un menú.
      * **JCheckBoxMenuItem**: Elemento del menú que se puede seleccionar o deseleccionar.
      * **JMenu**: Menú de una barra de menús.
      * **JRadioButtonMenuItem**: Elemento que forma parte de un grupo de elementos de menú.
    - **JToggleButton**: Botón de estados.
      * **JCheckBox**. Casilla de verificación.
      * **JRadioButton**: Botón de opción.
  + **JColorChooser**: Diálogo para seleccionar colores.
  + **JComboBox**: Combinación de caja de texto y lista desplegable.
  + **JLabel**: Etiqueta.
  + **JList**: Lista desplegable.
  + **JMenuBar**: Barra de menús.
  + **JOptionPane**: Componente que facilita la visualización de un cuadro de diálogo.
  + **JPanel**: Contenedor genérico.
  + **JPopupMenu**: Menú que aparece cuando se selecciona un elemento de una barra de menús.
  + **JProgressBar**: Barra de progreso.
  + **JScrollBar**: Barra de desplazamiento.
  + **JScrollPane**: Área de trabajo con barras de desplazamiento.
  + **JSeparator**: Separador para colocar entre dos elementos del menú.
  + **JSlider**: Permite seleccionar un valor dentro de un intervalo que define.
  + **JTableHeader**: Se utiliza para manejar la cabecera de una tabla.
  + **JTextComponent**: Clase base para los componentes de texto.
    - **JEditorPane**: Edita diferentes tipos de contenido.
      * **JTextPane**: Edita texto con formato, incluyendo imágenes.
    - **JTextArea**: Caja de texto multilínea.
    - **JTextField**: Caja de texto de una línea.
      * **JPasswordField**: Se usa para introducir contraseñas.
  + **JToolbar**: Barra de herramientas.

PROPIEDADES:

Las propiedades más importantes que presentan la mayoría de los componentes swing son las siguientes:

* **background**: Determina el color de fondo del componente.
* **font**: Establece el tipo de fuente que va a mostrar el componente.
* **foreground**: Establece el color de la fuente que muestra el componente.
* **horizontalAlignment**: Establece la alineación del texto dentro del componente en el eje X.
* **icon**: Indica si el componente muestra o no un icono, y cual sería.
* **labelFor**: Indicaría si el componente es etiquetable.
* **text**: Muestra el texto que indica la propiedad en componentes como caja de texto o etiquetas.
* **toolTipText**: Con esta propiedad definimos el texto que aparece como tool tip.
* **verticalAlignment**: Establece la alineación del texto dentro del componente en el eje Y.
* **aligmentX**: Alineamiento horizontal preestablecido para el componente.
* **aligmentY**: Alineamiento vertical preestablecido para el componente.
* **autoscrolls**: Determina si el componente puede realizar scroll de forma automática cuando se arrastra el ratón.
* **border**: Establece el tipo de borde que va presentar el componente.
* **componentPopupMenu**: Estable el menú contextual que se muestra en este componente.
* **cursor**: Establece el tipo de cursor que se va a mostrar cuando el curso entre en el componente.
* **disableIcon**: Establece el icono a mostrar si el componente no está activo.
* **enabled**: Nos indica si el componente está o no activo.
* **focusable**: Establece si el componente puede o no, recibir el foco.
* **horizontalTextPosition**: Establece la posición horizontal del texto del componente, en relación con su imagen.
* **iconTextGap**: Si el componente tiene activo el texto y el icono, con esta propiedad se establece la distancia entre ellos.
* **inheritsPopupMenu**: Indica si el menú contextual se hereda o no.
* **inputVerifier**: Establece el componente de verificación para este componente.
* **maximunsize**: Establece el tamaño máximo del componente.
* **minimunsize**: Establece el tamaño mínimo del componente.
* **name**: Establece el nombre del componente.
* **opaque**: Modifica la opacidad del componente.
* **preferredSize**: Tamaño predefinido para este componente.
* **verifyInputWhenFocusTarget**: Si el componente es verificado antes de recibir el foco.
* **verticalTextPosition**: Establece la posición vertical del texto del componente, en relación con su imagen.

MÉTODOS:

Cada componente que forma parte de una aplicación GUI utilizando Java, dispone de un conjunto completo de métodos, que nos permite comunicarnos con el componente y modificar su aspecto o comportamiento. A continuación se va a mostrar una lista de los componentes más importantes de la swing Java, junto con sus métodos más destacados:

* **JFrame**. Los métodos más importantes son: getTitle(), setBounds(x,yx,w,h), setLocation, setMaximumSize(w,h), setMinimumSize(w,h), setPreferredSize(w,h), setResizable(bool), setSize(w,h), setTitle(str) y setVisible(bool).
* **JPanel**. El panel de contenido dispone del método add(), para añadir componentes al panel. paintComponent(Graphics g) sirve para incluir componentes gráficos al panel.
* **JLabel**: Las etiquetas tienen como métodos más importantes: setText() que permite modificar el texto, setVerticalAlignment() para modificar la alineación vertical, etc.
* **JButton**: Los botones presentan entre otros métodos: setEnabled(bool) que permite activar o desactivar el botón, isEnabled() que permite comprobar si está o no activo, setMnemonic(char) que permite asociar una tecla al botón, etc.
* **JProgressBar**, **JScrollBar**. Estos componentes disponen de los siguientes métodos: setExtent(), setMaximum(), setMinimum(), setValue(), getValueIsAdjusting() y setOrientation().
* **JSlider**: Este componente tiene como métodos más destacados: setPaintTicks(bool), setPaintLabels(bool). setMajorTickSpacing(int) y setMinorTickSpacing(int) para determinar los valores del intervalo, setSnapToTicks(true) y setInverted(bool).
* **JRadioButton**. Este componente tiene como método principal isSelected() que devuelve el estado del mismo.
* **JList**. En las listas desplegables destacan los métodos siguientes para editar los elementos de la lista: addElement(objName), elementAt(index), removeElement(objName), removeAllElements() y removeElementAt(idx). Para comprobar el estado de la lista se usan los métodos contains(objName), indexOf(name) y size(). Para convertir la lista en array se usa copyInto(array).
* **JComboBox**. Dispone de diferentes métodos, destacando setEditable(true) que permite editar los items del combo box. Para recuperar los ítems seleccionados se usa getSelectedItem() y getSelectedIndex(). Otros métodos útiles son getItemAt(idx), getItemCount(), setSelectedItem(obj) y setMaximumRowCount(x).
* **JTextPane** y **JTextArea**. Como métodos más útiles de estos componentes destacan getText(), setText(), append(str), insert(str,pos), replace(str,beg,end), setEditable(bool), setLineWrap(bool) y setWrapStyleWord(bool).
* **JMenuItem**. Para añadir separadores se usa el método addSeparator() o insertSeparator(posn). Los Menú ítems se pueden activar o desactivar con setEnabled(bool) y comprobar si están activos con isEnabled(). Para añadir teclas de acceso rápido se usa el método setMnemonic(char).