Realizar la consulta y definición de cada uno de los siguientes conceptos:

Ventana principal y contenedores

Tal como se ha mencionado en las primeras entradas sobre java Swing, los contenedores son componentes que permiten almacenar, alojar o contener otros elementos gráficos nuevamente mencionamos que es el Tapiz donde vamos a pintar.

Java Swing provee algunos contenedores útiles para diferentes casos, así cuando desarrollamos una Ventana podemos decidir de qué manera presentar nuestros elementos, como serán alojados y de qué forma serán presentados al usuario.

Distribución gráfica de los elementos

En java, cuando hacemos ventanas, la clase que decide cómo se reparten los botones (Y demás controles) dentro de la ventana se llama Layout. Esta clase es la que decide en qué posición van los botones y demás componentes, si van alineados, en forma de matriz, cuáles se hacen grandes al agrandar la ventana, etc. Otra cosa importante que decide el Layout es qué tamaño es el ideal para la ventana en función de los componentes que lleva dentro.

Divisiones, paneles, etiquetas, zonas de texto

Como se ha mencionado, los componentes gráficos son estos elementos que permiten brindar una interacción con el usuario del sistema..... Cada componente corresponde a una clase en Java, por esta razón cuando desarrollamos y queremos vincular uno de estos elementos simplemente instanciamos la clase que necesitamos, es decir, si queremos un Área de texto debemos crear un objeto de la clase JTextArea....

En la introducción sobre Swing vimos un pequeño árbol de herencia, sin embargo este no enmarca todos los componentes Gráficos de la librería sino solo algunos de los principales, a continuación vamos a ampliar el número de esos componentes agrupándolos en categorías dependiendo de su funcionalidad.

Delegación de eventos

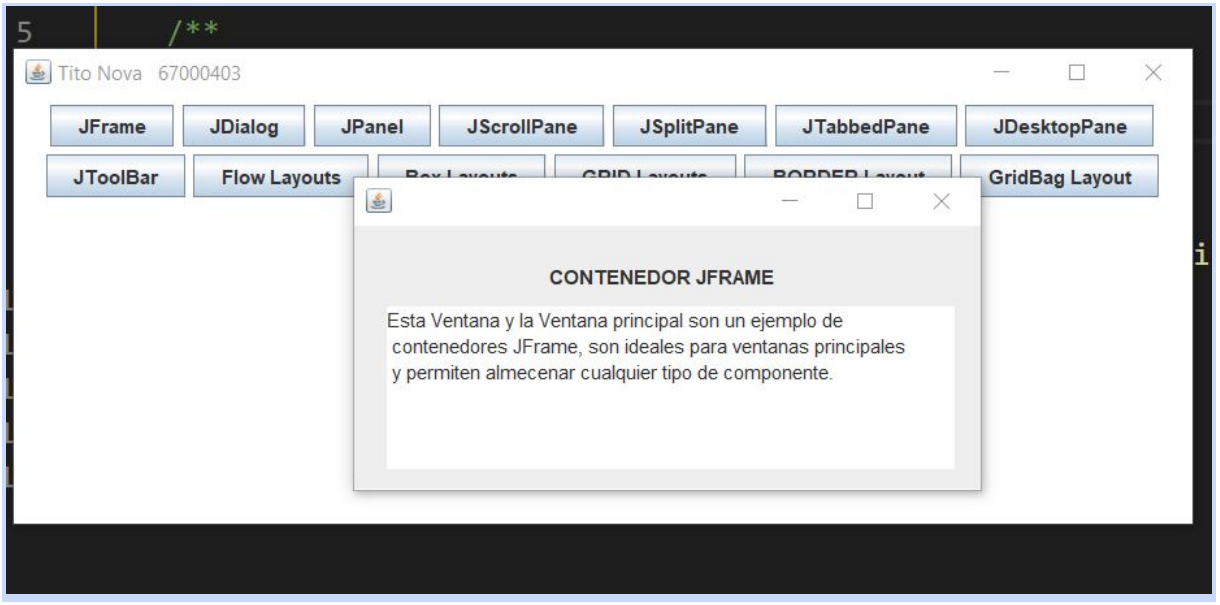
Los eventos son básicamente las acciones que suceden al ejecutar o realizar alguna invocación a un evento es una acción que podemos controlar cuando por ejemplo se ejecuta algún proceso, si ingresamos a cierto componente, presionamos un botón, movemos el mouse etc internamente el sistema identifica que es lo que se está haciendo, si hacemos clic, presionas una tecla, movemos el mouse, todas esas son acciones que pueden ser identificadas y controladas.En Java estos eventos son conocidos como Listeners o escuchadores, cuando trabajamos con esto a la vez estamos aplicando conceptos de POO como es el trabajo con interfaces, ya que para poder hacer uso de cualquier tipo de evento debemos implementar la interface adecuada que nos permite realizar el proceso en este ejemplo al trabajar con eventos del mouse implementaremos la interface MouseListener, que nos provee una serie de métodos importantes para controlar las acciones del ratón.

TITO ALEXANDER NOVA AREVALO **67000403**

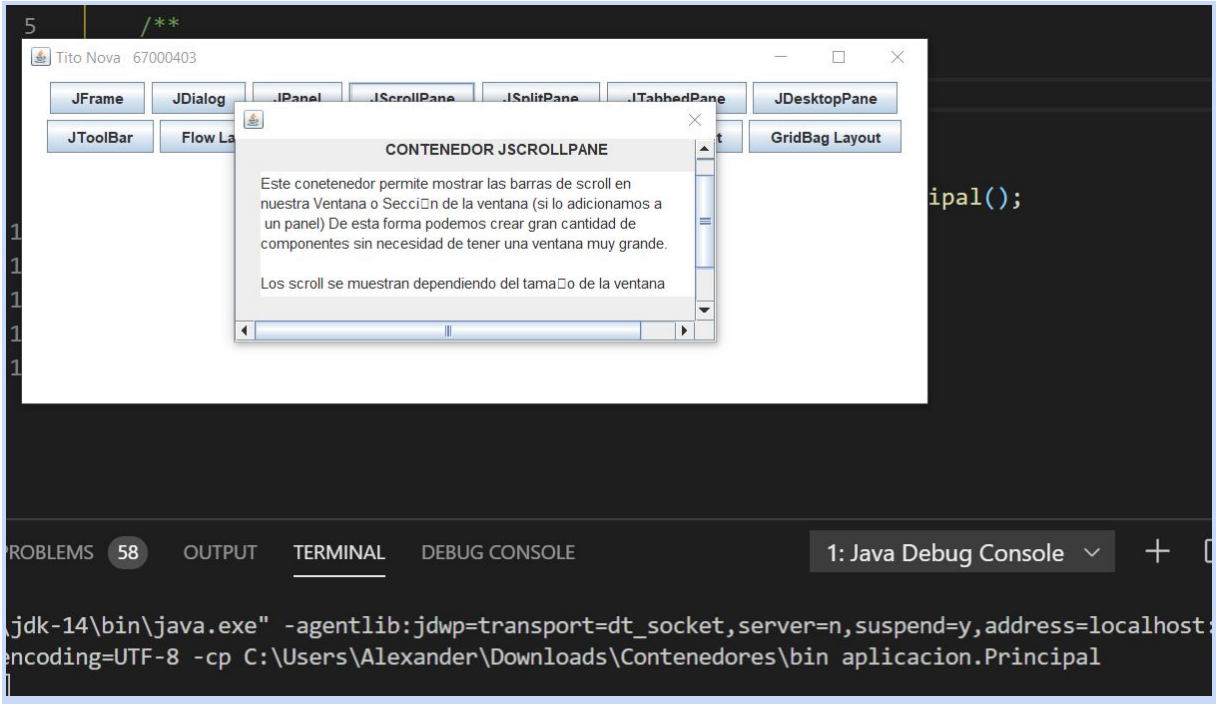
EVENTOS, LAYOUTS Y CONTENTS

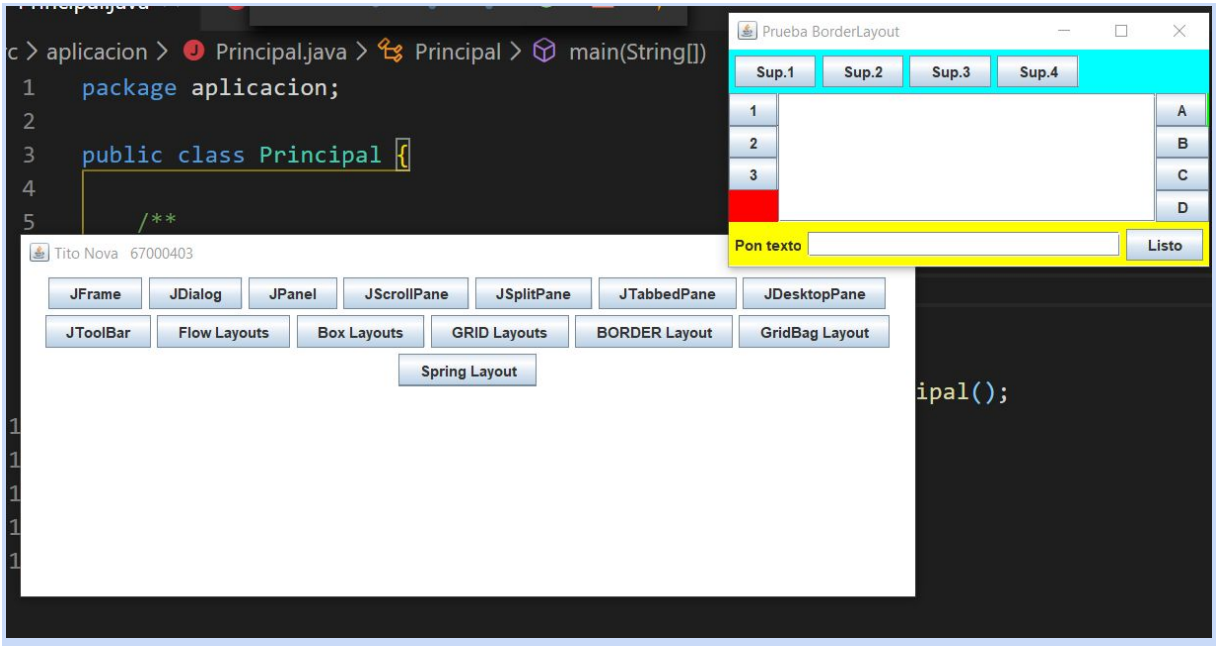
```java
@Override
public void actionPerformed(ActionEvent evento) {
    if (evento.getSource()==botonJFrame)
    {
        ClaseFrame miClaseJFrame= new ClaseFrame();
        miClaseJFrame.setVisible(true);
    }
    if (evento.getSource()==botonJDialog)
    {
        ClaseJDialog miClaseJDialog=new ClaseJDialog(miVentanaPrincipal,true);
        miClaseJDialog.setVisible(true);
    }
    if (evento.getSource()==botonJPanel)
    {
        ClaseJPanel miClaseJPanel= new ClaseJPanel(miVentanaPrincipal,true);
        miClaseJPanel.setVisible(true);
    }
    if (evento.getSource()==botonJScrollPane)
    {
        ClaseJScrollPane miClaseJScrollPane = new ClaseJScrollPane(miVentanaPrincipal,true);
        miClaseJScrollPane.setVisible(true);
    }
    if (evento.getSource()==botonJTabbedPane)
    {
```

Connect Ln 199, Col 7 Tab Size: 4 UTF-8 CRLF Java



TITO ALEXANDER NOVA AREVALO **67000403**





TITO ALEXANDER NOVA AREVALO **67000403**