Laboratorio de programación orientada a objetos
Práctica de laboratorio 1

Importante: Las prácticas de laboratorio deben ser realizadas de forma individual, se debe trabajar solamente en la práctica ya que el navegar, chatear, mensajear o realizar actividades que no tengan nada que ver con la práctica implica la NO aprobación de la misma.
Es necesario anotar la hora de inicio y finalización de cada una de las prácticas.
Al terminar la práctica dar aviso al instructor para que sea revisada.

Práctica 14. Interfaz de Gráfica de Usuario y manejo de eventos.

En la presente práctica se creará una GUI que permita por medio de acciones, abrir un archivo y mostrar su contenido en un Area de texto dentro de la interfaz gráfica; para ello el trabajo de esta práctica consiste en:

Paso 1: Al nivel de los paquetes personajes y pruebas, crear el paquete **ventanas**.

Paso 2: Dentro del paquete ventanas, crear la clase **VentanaPrincipal** la cual contendrá los elementos de los pasos siguientes.

Paso 3. En la sección de los atributos definir los componentes siguientes:

```
JFrame f;
JLabel lblFile;
JTextField txtFil
JButton btnOpen;
JTextArea txtCont
JLabel lblN;
JLabel lblLeidos;
JButton btnExit;
```

Paso 4. Dentro del **constructor** definir las propiedades de los componentes con los siguientes datos:

f : Frame con el título "Practica 14"
lblFile : la etiqueta debe mostrar el texto "Nombre de archivo”
txtFile : El ancho de la caja de texto deberá ser de 20 caracteres
btnOpen : El botón deberá decir en su texto ”Abrir archivo”
txtContenido : El área de texto deberá de ser de 30 filas por 40 columnas
lblLeidos : La etiqueta deberá mostrar el texto ”Caracteres leídos”
lblN : Etiqueta que deberá mostrar sólo en numero CERO 0
btnExit : Botón con el texto “Salir”

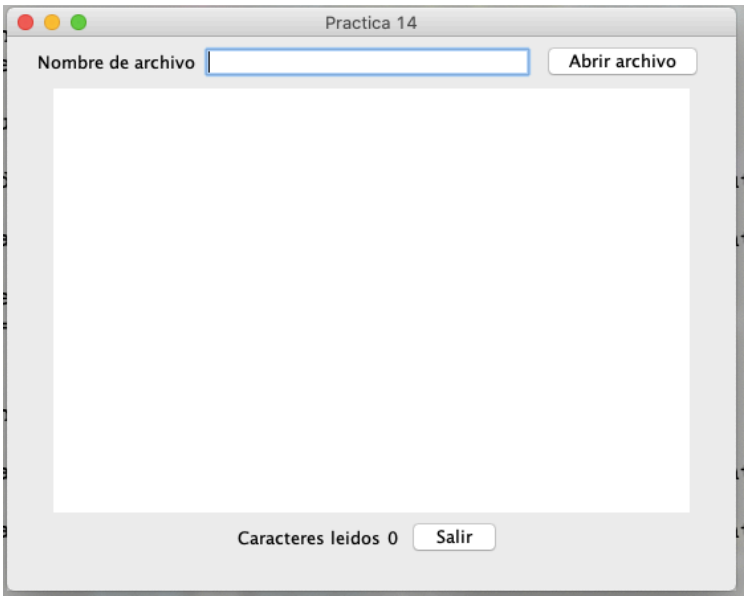
Paso 5. Crear el método **initComponents** en el cual deberá:

A. Definir para el Frame el tipo de Layout, el cual deberá ser FlowLayout
B. Agregar al Frame (f) los siguientes componentes en el orden que se especifica: etiqueta **lblFile**, campo de texto **txtFile**, botón **btnOpen**, área de texto **txtContenido**, etiqueta **lblLeidos**, etiqueta **lblN** y por último el botón **btnExit**.

C. Establecer el tamaño de la ventana (frame) en 550 x 440 pixeles.
D. Agregar el cerrado de la aplicación por defecto al cerrar la ventana
E. Establecer el frame en visible.

Paso 6. Crear el método **main** el cual deberá crear una instancia **v** de **VentanaPrincipal** y por medio de esa instancia mandar llamar al método **initComponents()**.

NOTA. Hasta este punto se puede compilar y ejecutar la VentanaPrincipal y la ventana resultado deberá verse como en la imagen muestra.



Paso 7. En el método **initComponents** deberá realizar las siguientes incrustaciones de código:

A. Después de agregar el botón **btnOpen** mandar llamar a su método **addActionListener** el cual por medio de una clase **anónima** dentro de su método obligatorio mandará llamar al método **openFile()**.
B. Después de agregar el botón **btnExit** mandar llamar a su método **addActionListener** el cual por medio de una clase **anónima** dentro de su método obligatorio mandará llamar al método **dispose()** el cual pertenece al **Frame**.

Paso 8. Agregar el método **openFile** después del método principal dentro del cual deberá realizar lo siguiente:

A. En la variable **file** de tipo cadena almacenar el texto de la caja de texto **txtFile**.
B. En la variable **path** de tipo cadena, almacenar el directorio del usuario agregando la variable **file** capturada en el inciso anterior.
C. **Opcional**. Imprimir en pantalla la variable **path** para identificar que se trata de un archivo en el directorio del usuario.
D. Limpiar el contenido del área de texto **txtContenido** usando el método setText
E. Establecer la variable **archivo** de tipo File para determinar si **existe**; en caso de no existir mostrar un mensaje de que no existe por medio de un **showMessageDialog**.
F. **En caso de existir**, este puede ser un archivo o un directorio; sí **es un archivo** mandar llamar al método **getContenido()**; si **es un directorio** deberá llamar al método **getList()**; ambos reciben un File.

NOTA: Todos los componentes tienen métodos para acceder o modificar sus atributos

Paso 9. El método **getContenido()** que recibe un Flie deberá abrir el archivo para lectura y el contenido de este mostrarlo en el área de texto **txtContenido**; precaución ya que al establecer el texto borrará lo anterior, por lo que deberá agregar más texto.

Al momento de leer el contenido, deberá contar los caracteres leídos para sumarlos a una variable **count** que almacena los caracteres leídos del archivo.

Al concluir de leer el archivo deberá establecer la variable **count** en la etiqueta **lblN**.

Paso 10. El método **getList()** que recibe un **File** y este a su vez es un directorio, obtener el listado de archivos y directorios que contiene y mostrarlos en el área de texto **txtContenido**; considerar la advertencia anterior y no sobre-escribir cada nombre de archivo.

Para cada archivo o directorio contar los caracteres para agregarlos a una variable **count** y al final modificar el texto de la variable **lblN** con el contador obtenido.

Paso 11. Compilar y ejecutar VentanaPrincipal.java

La salidas muestra son las siguientes:

Si es un archivo	Si es un directorio