

TRABAJO FINAL UNIDAD 6

PROGRAMACION

EDITOR DE TEXTO

A screenshot of a Java Swing window titled "Student Registration". The window has a standard title bar with minimize, maximize, and close buttons. It contains four text input fields labeled "First Name", "Last Name", "Email ID", and "Phone No.". At the bottom, there are two buttons: "Regist..." and "clear".

Javier Ruiz Molero

[javierrm11/EDITOR TEXTO JRM \(github.com\)](https://github.com/javierrm11/EDITOR_TEXTO_JRM)

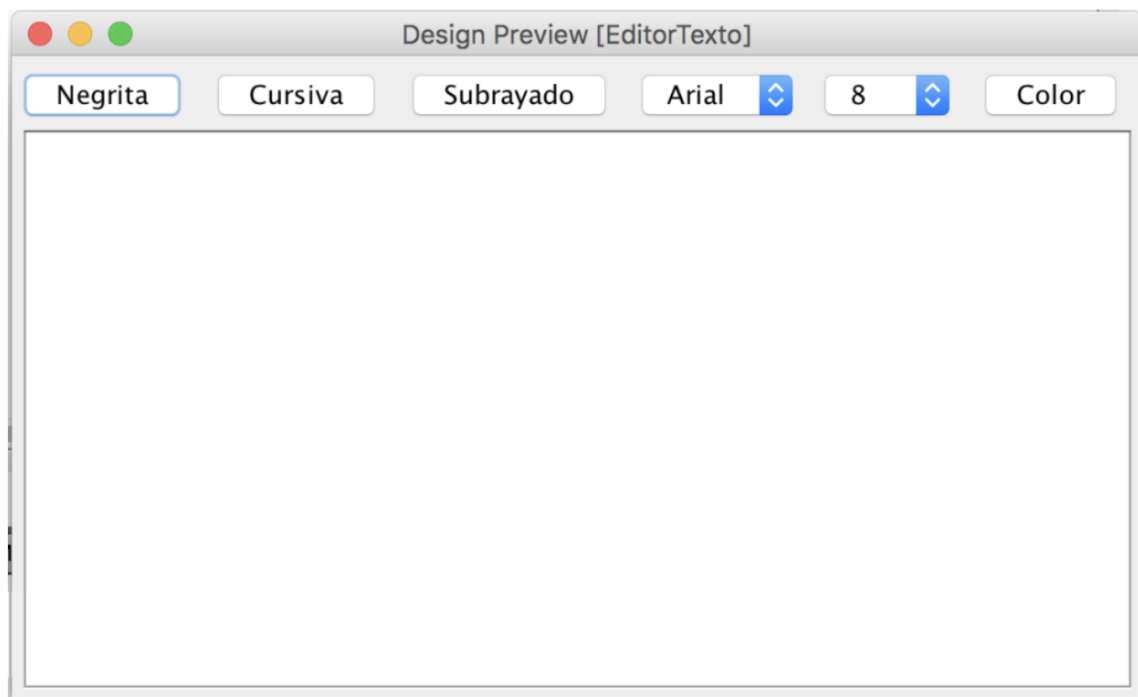
INDICE

1. EnunciadoDelTrabajo	3
2. RealizacionDelTrabajo	4
a. OrganizacionDelProyecto	4
b. CreacionDelFrame	5
c. CreacionDeLosBotones	5
d. CreacionPanelTexto	6
e. ActionsListener	7
f. MenuBar	8

Enunciado del trabajo

Se debe realizar un editor de texto con Java Swing, todo el texto debe estar explicado en un pdf, se debe adjuntar en él un enlace a un repositorio GIT.

El nombre del proyecto será EDITOR_TEXTO_JGR (siendo JGR las iniciales de los nombres y apellidos). El pdf también seña EDITOR_TEXTO_JGR.pdf



Realización del trabajo

1º Como vamos a organizarnos

Vamos a ver como vamos a realizarlo y sobretodo en que orden vamos a proceder.

Lo primero, es decir que lo vamos a hacer con la librería de Java que es **Swing**, que sirve para hacer interfaces gráficas a nuestro programa.

- El **primer paso** que he seguido para el proyecto es la creación del Frame del programa y su tamaño mas o menos, en los siguientes puntos veremos que hace el código.
- El **segundo paso** es la creación de los botones, de los combos y del selector de colores.
- El **tercer paso** es la creación del JTextPane para escribir.
- El **cuarto paso** es la funcionalidad de los botones al escribir y el scroll en el JTextPane.
- El **quinto paso** es la creación de un menú arriba para poder guardar el archivo y poder terminar el programa.
- Y el **ultimo paso** es darle la funcionalidad al menú y dejar bonito el programa.

2º Creación del Frame

La primera línea lo que hace es inicializar el Editor (Programa). Luego con setTitle, le asignamos que nombre va a tener.

```
EDITOR_TEXTO_JRM editor = new EDITOR_TEXTO_JRM();
editor.setTitle("Design Preview [ Editor de Texto ]");
editor.setLocation(400, 100);
editor.setSize(750, 600);
editor.setVisible(true);
editor.setDefaultCloseOperation(EXIT_ON_CLOSE);
```

Con setLocation le tenemos que pasar dos parámetros, el eje X y el eje Y, lo que dice es donde de la pantalla del usuario se va a abrir. Luego le tenemos que meter el size que es lo que ocupa de ancho y de alto y luego tenemos 2 expresiones diferentes. setVisible y la ultima línea indica que cuando se le de a la X para cerrar el programa, el programa termine.

3º Creación del panel de los botones

En la siguiente imagen vemos como declarar las variables para cada cosa:

- JPanel → contenedor para el contenido.
- JButton → Botones.
- JComboBox → desplazador para elegir algún valor.

```
- private JPanel panel;
private JButton negrita, cursiva, subraya,
color;
private JComboBox<String> combo, tamaño;
```

Ahora voy a explicar como se hace con un botón y explicare más adelante cosas que sean complicadas, sin tener que poner todo el código.

1. Inicializar el botón

```
negrita = new JButton("Negrita");
```

2. Añadirle Padding

```
negrita.setBounds(30, 20, 80, 30);
```

3. Añadir fondo al botón

```
negrita.setBackground(Color.darkGray);
```

4. Cambiar color de fuente

```
negrita.setForeground(Color.white);
```

5. Poner un borde diferente, lo he investigado a través de la página oficial de Oracle.

```
negrita.setBorder(BorderFactory.createRaisedBevelBorder());
```

6. Añadir al panel

```
panel.add(negrita);
```

4º Creación del panel para escribir

Como hemos visto en el apartado uno de la creación del botón debemos de crear una variable indicando el tipo, en este caso es ***JTextPane***.

```
escribir = new JTextPane();  
JScrollPane scrollPane = new  
JScrollPane(escribir);
```

Iniciamos además el JScrollPane ya que debemos de tener un scroll horizontal y vertical entonces se crea así y debemos pasarle como parámetro el JTextPane que hemos creado anteriormente.

```
escribir.setEditable(true);  
panel.add(scrollPane);
```

- setEditable: Lo que hace es que se pueda escribir en el panel de texto.
- Add: Añadir al panel principal el scrollPanel

5º Actions Listener

Primer os voy a enseñar un ejemplo de como crear que el botón tenga una acción cuando se le de al botón, combo...

```
negrita.addActionListener(new Bold());
```

Lo que hace es al botón negrita añadir la acción cuando se le de y se llamara Bold la función/método que la crearemos ahora después. Y hemos añadido las siguientes:

- Bold: Texto negrita
 - Cursiva: Texto en Cursiva
 - Subraya: Texto subrayado
 - CambiarFuente: Para cambiar la fuente del texto
 - Tamaño: Para cambiar el tamaño del texto.
 - SeColor: Para entrar al selector de colores para la fuente.
- a. Bold, Cursiva, Subraya

Vamos a explicar uno y así se hacen todos. Bold:

```
MutableAttributeSet attrs =  
escribir.getInputAttributes();  
boolean bold = (StyleConstants.isBold(attrs))  
? false : true;
```

```
StyleConstants.setBold(attrs, bold);  
escribir.setCharacterAttributes(attrs, false);
```

Lo que hace la primera línea es obtener los atributos del JTextPanel. Los atributos de entrada son los atributos que se aplicarán al texto que se escriba después. Luego, verifica si el texto está en negrita o no. Si devuelve true, pone el texto en negrita y si no es así lo desactiva. Y luego añade los atributos.

6º Añadir Menu Bar

Esto no era parte del trabajo pero quería ir más allá y añadir un menú arriba para poder guardar el archivo y poder terminar el programa. Para ello vamos a seguir los mismos pasos mas o menos que los pasos anteriores.

1. Añadimos las variables

```
private JMenuBar menubar;  
private JMenu menu;  
private JMenuItem mi1,mi2,mi3;
```

2. Inicializar el menú bar.

```
menubar=new JMenuBar();  
setJMenuBar(menubar);
```

Lo que hace el setMenu es indicar que el menú del programa JFrame es el menú que hemos creado.

3. Inicializar los Items del bar menú.

```
menu=new JMenu("Archivo"); // Inicializa el  
menu  
  
mi1=new JMenuItem("Guardar"); // Inicializa un  
elemento del menu
```

4. Action Listener del menú

Lo que vamos a hacer primero indicar el nombre de la función del evento/acción que se hace cuando se pinche en dichos apartados, en este caso, esta claro que una va a ser para guardar el archivo y otro para salir del programa.

```
mi1.addActionListener(new Guardar());  
mi2.addActionListener(new Salir());
```

Ahora aquí os dejo el código que hemos utilizado para guardar un archivo:

```
private class Guardar implements ActionListener  
{  
    public void actionPerformed(ActionEvent e) {  
        JFileChooser file = new JFileChooser();  
        int seleccion =  
file.showSaveDialog(null);  
        if (seleccion ==  
JFileChooser.APPROVE_OPTION) {  
            File texto =  
file.getSelectedFile();  
            try {  
                FileWriter escribe = new  
FileWriter(texto);  
  
escribe.write(escribir.getText());  
                escribe.close();  
  
JOptionPane.showMessageDialog(null, "Guardado  
exitosamente");  
            } catch (IOException ex) {  
                ex.printStackTrace();  
  
JOptionPane.showMessageDialog(null, "Error al  
guardar el archivo");  
            }  
        }  
    }  
}
```

Lo que hace es hacer un FileChooser como hemos visto en la unidad. Luego, llamas al método **showSaveDialog(null)**, que muestra el diálogo de guardado y devuelve un entero que indica si el usuario ha seleccionado "Guardar" o ha cancelado el diálogo.

Ahora hay un if y lo que hace si ha seleccionado Guardar:

- Obtienes el texto y lo guardas en una variable.
- Haces un try para intentarlo y si da error pues salta una excepción.
- Lo que hace el try es coger el archivo que ha seleccionado el usuario. Y lo que hace es escribir en ese archivo el texto que hay en nuestro programa y lanza un mensaje con JOptionPane.

Y en el Action Listener de Salir el código es:

```
private class Salir implements ActionListener {  
    public void actionPerformed(ActionEvent e) {  
        dispose();  
    }  
}
```

Lo que hace con dispose(); es terminar el programa.