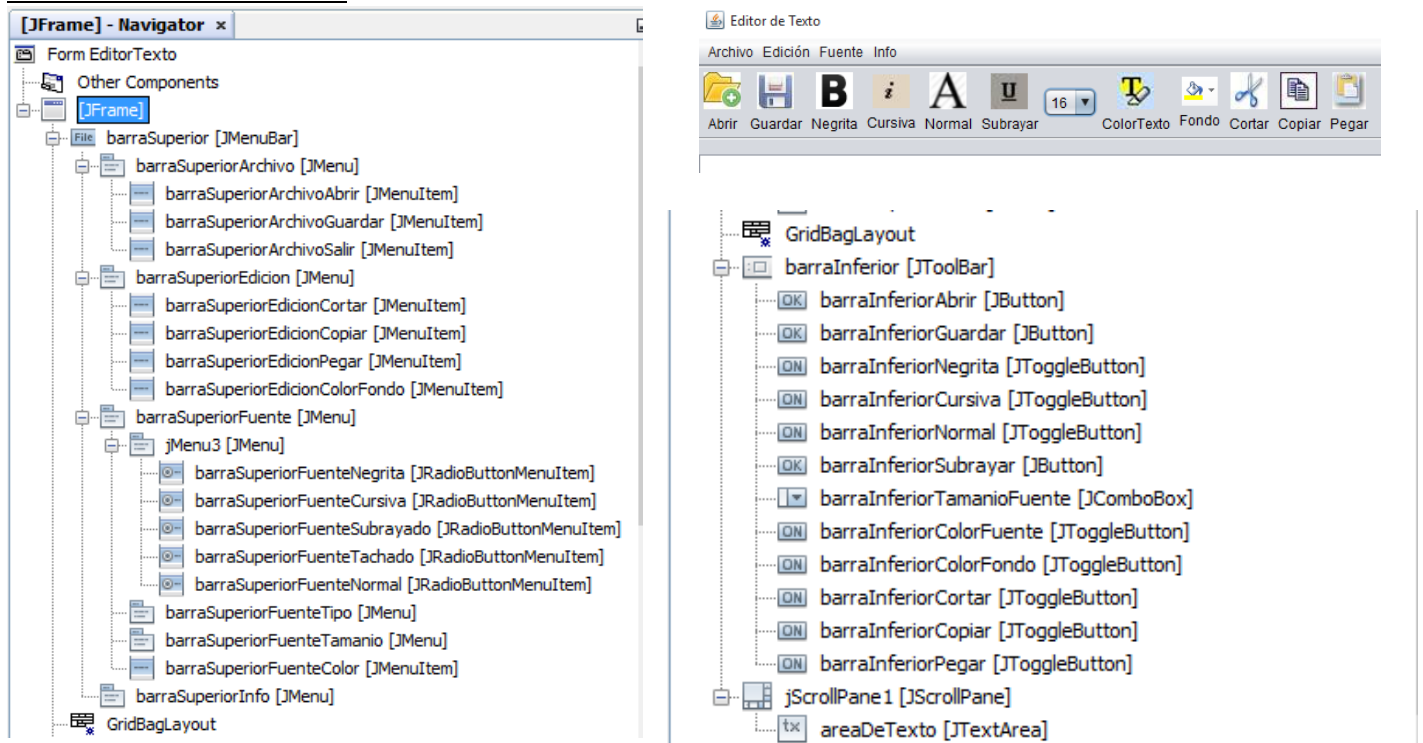


Jose Luis García de Diego
PASOS PARA DESARROLLAR EL PROGRAMA EDITOR

1.- DISEÑO DE LOS OBJETOS



2.- Cambiar las propiedades de los objetos

En Properties cambio el nombre del Objeto en el diseño (Text)

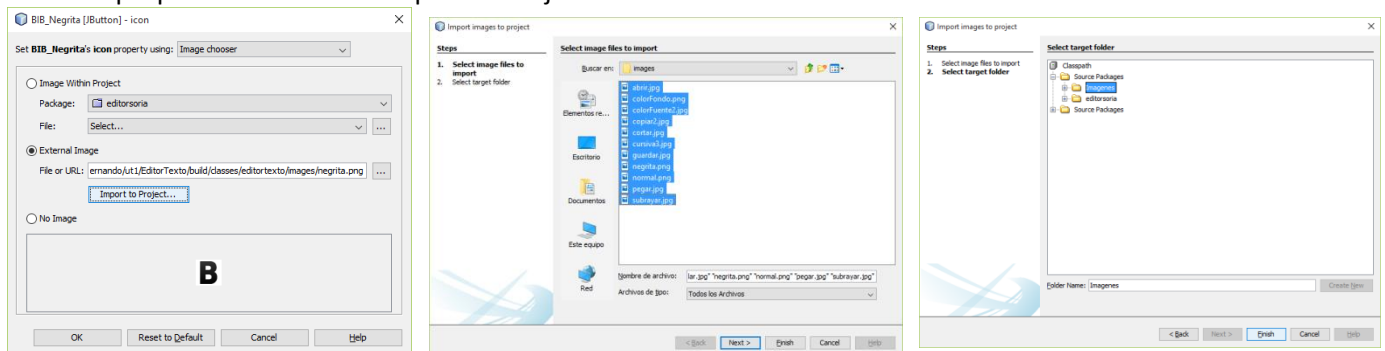
Poner una imagen en el objeto (Icon)

En Code cambio el nombre del Objeto en la jerarquía (Variable Name)

En Events Seleccionar en el campo "ActionPerformed" La opción que sale por defecto en el tirador.

3.- Importar las imágenes del proyecto.

Desde la propiedad Icon ... => "Import to Project" =>



Poner las imagen en los objetos que queramos desde el paquete al que hemos importado.

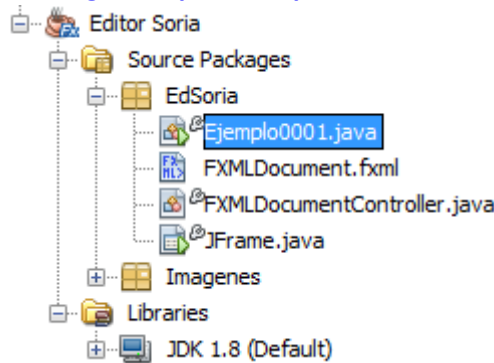
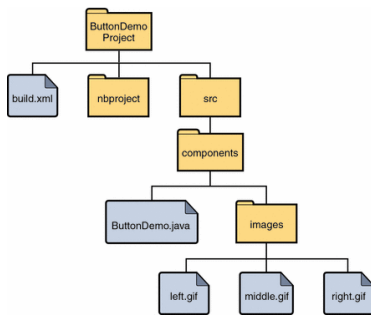
4.- Importar los paquetes que vamos a utilizar (Desde código de programa)

package (nuestro JFrame);

```
import java.awt.Color;
import java.awt.Font;
import java.awt.GraphicsEnvironment;
import java.awt.datatransfer.Clipboard;
import java.awt.datatransfer.DataFlavor;
import java.awt.datatransfer.StringSelection;
import java.awt.datatransfer.Transferable;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.font.TextAttribute;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Map;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JColorChooser;
import javax.swing.JFileChooser;
import javax.swing.JLabel;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.filechooser.FileView;
```

(*) Enlaces a métodos propios de Java/Swing

<http://docs.oracle.com/javase/tutorial/uiswing/examples/components/index.html>



5.- PROGRAMA MAIN

```
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    <!-- editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) -->
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
    * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        //Chequea problemas en ejecución (Ej. Excepciones = División por 0)
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
        //Cada excepción de ejecución identificada en un catch ejecuta el método incluido en él.
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(EditorTexto.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(EditorTexto.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(EditorTexto.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(EditorTexto.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    <!-- editor-fold -->

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new EditorTexto().setVisible(true);
        }
    });
} //Fin MAIN
```

6.- Ir resolviendo los diferentes métodos invocados en cada objeto.

Utilizar el siguiente método para comprobar que funcionan las llamadas a los métodos.

```
private void handleButtonAction(ActionEvent event) {
    TAAreaDeTexto.setText("Has pulsado ...");
}
```

Archivo Abrir

Clases que necesita el método abrir.

Import de todos los que de error (9, pero A mi no me ha dado ningún error).

```

class JavaFileView extends FileView {
    public String getName(File file) {
        String filename = file.getName();
        if (filename.endsWith(".java")) {
            String name = filename + " : " + file.length();
            return name;
        }
        return null;
    }
}

class FiltroTxt extends javax.swing.filechooser.FileFilter {
    public boolean accept(java.io.File f) {
        if (f.isDirectory()) {
            return false;
        }
        String nombre = f.getName().toLowerCase();
        if (nombre != null) {
            if (nombre.endsWith(".txt")) {
                return true;
            }
        }
        return false;
    }
    public String getDescription() {
        return "*.txt ";
    }
}

private void barraSuperiorArchivoAbrirActionPerformed(java.awt.event.ActionEvent evt) {
    JFileChooser file = new JFileChooser();
    JLabel nombreFichero = new JLabel();
    FileView view = new JavaFileView();
    file.setFileView(view);
    file.addChoosableFileFilter(new FiltroTxt());
    int opcion = file.showOpenDialog(this);
    if (opcion == JFileChooser.APPROVE_OPTION) {
        try {
            nombreFichero.setText(file.getSelectedFile().getName());
            FileReader reader = null;
            reader = new FileReader(file.getSelectedFile().getAbsolutePath());
            try {
                areaDeTexto.read(reader, file.getSelectedFile().getName());
            } catch (IOException ex) {
                Logger.getLogger(EditorTexto.class.getName()).log(Level.SEVERE, null, ex);
            } catch (FileNotFoundException ex) {
                Logger.getLogger(EditorTexto.class.getName()).log(Level.SEVERE, null, ex);
            }
        } else {
            nombreFichero.setText("");
        }
    }
}

```

Archivo Guardar.

```
private void barraSuperiorArchivoGuardarActionPerformed(java.awt.event.ActionEvent evt) {
    FileWriter writer = null;
    JFileChooser filechooser = new JFileChooser();
    filechooser.showSaveDialog(this);

    if (filechooser.getSelectedFile() != null){ //si no se pulsa Cancelar
        String file = filechooser.getSelectedFile().getAbsolutePath()+".txt";
        try {
            writer = new FileWriter(file);
            areaDeTexto.write(writer);
        } catch (IOException exception) {
            System.err.println("Error al guardar"); exception.printStackTrace();
        } finally {
            if (writer != null) {
                try { writer.close();
                } catch (IOException exception) {
                    System.err.println("Error closing writer"); exception.printStackTrace();
                }
            }
        }
    }
}
```

Archivo Salir.

```
private void barraSuperiorArchivoSalirActionPerformed(java.awt.event.ActionEvent evt) {
    if (JOptionPane.showConfirmDialog(this, "¿Desea Salir?", "Salir", JOptionPane.YES_NO_OPTION) == 0) {
        this.dispose();
        System.exit(0);
    }
}
```

Crear Objeto Portapapeles

Declarar la variable justo debajo de : “public class **EditorTexto** extends javax.swing.JFrame {“

```
Clipboard portapapeles = getToolkit().getSystemClipboard();
```

Edición - Cortar

```
private void barraSuperiorEdicionCortarActionPerformed(java.awt.event.ActionEvent evt) {  
    StringSelection ts = new StringSelection(areaDeTexto.getSelectedText());  
    portapapeles.setContents(ts, ts);  
    areaDeTexto.replaceRange("", areaDeTexto.getSelectionStart(), areaDeTexto.getSelectionEnd());  
}
```

Edición - Copiar

```
private void barraSuperiorEdicionCopiarActionPerformed(java.awt.event.ActionEvent evt) {  
    StringSelection ts = new StringSelection(areaDeTexto.getSelectedText());  
    portapapeles.setContents(ts, ts);  
}
```

Edición - Pegar

```
private void barraSuperiorEdicionPegarActionPerformed(java.awt.event.ActionEvent evt) {  
    Transferable tf = portapapeles.getContents(this);  
    try {  
        String cadena = "";  
        cadena = (String) tf.getTransferData(DataFlavor.stringFlavor);  
        areaDeTexto.insert(cadena, areaDeTexto.getSelectionStart());  
    } catch (Exception ee) {  
    }  
}
```

Crear Objeto Portapapeles

Declarar la variable justo debajo de : “public class **EditorTexto** extends javax.swing.JFrame {“

```
private Color color = Color.BLACK;  
private JColorChooser colorChooser = new JColorChooser();
```

Edición – Color de fondo

```
areaDeTexto.setBackground(colorChooser.showDialog(this, "Elija color de fondo", color));
```

Crear las objetos para trabajar los Estilos de Fuentes.

Declarar la variable justo debajo de : “public class **EditorTexto** extends javax.swing.JFrame {“

```
//Fuente por defecto
private String tipo = "Times";
private int estilo = Font.PLAIN;
```

Fuente – Estilo - Negrita

```
String tipoFuente = areaDeTexto.getFont().getFontName();
int tamanoFuente = areaDeTexto.getFont().getSize();
Font fuente = new Font (tipoFuente, Font.BOLD, tamanoFuente);
areaDeTexto.setFont(fuente);
```

Fuente – Estilo - Cursiva

```
String tipoFuente = areaDeTexto.getFont().getFontName();
int tamanoFuente = areaDeTexto.getFont().getSize();
Font fuente = new Font (tipoFuente, Font.ITALIC, tamanoFuente);
areaDeTexto.setFont(fuente);
```

Fuente – Estilo - Subrayado

```
Map atributosFuente = areaDeTexto.getFont().getAttributes();
atributosFuente.put(TextAttribute.UNDERLINE,TextAttribute.UNDERLINE_ON);
Font fuente = new Font(atributosFuente);
areaDeTexto.setFont(fuente);
```

Fuente – Estilo - Tachado

```
Map atributosFuente = areaDeTexto.getFont().getAttributes();
atributosFuente.put(TextAttribute.STRIKETHROUGH, TextAttribute.STRIKETHROUGH_ON);
Font fuente = new Font(atributosFuente);
areaDeTexto.setFont(fuente);
```

Fuente – Estilo - Normal

```
// String tipoFuente = areaDeTexto.getFont().getFontName();
int tamanoFuente = areaDeTexto.getFont().getSize();
Font fuente = new Font (tipo, Font.PLAIN, tamanoFuente);
areaDeTexto.setFont(fuente);
```

Si ponemos esto no lo hace

// Coge el tipo fuente definido al comienzo

Crear el objeto tipos de Fuentes.

Declarar la variable justo debajo de : “public **JFrame**() { initComponents(); “
private String tipo = "Times";

//tipos de fuente: botón

```
String [] listaDeFuentes = GraphicsEnvironment.getLocalGraphicsEnvironment().getAvailableFontFamilyNames();
for (int i=0; i < listaDeFuentes.length; i++){
    final JMenuItem TIPO_FUENTE = new JMenuItem();
    TIPO_FUENTE.setText(listaDeFuentes[i]);
    TIPO_FUENTE.addActionListener(new ActionListener(){
        public void actionPerformed (ActionEvent ae) {
            int estiloFuente = areaDeTexto.getFont().getStyle();
            int tamanoFuente = areaDeTexto.getFont().getSize();
            Font f = new Font(TIPO_FUENTE.getText(), estiloFuente, tamanoFuente);
            areaDeTexto.setFont(f);
        }
    });
    barraSuperiorFuenteTipo.add(TIPO_FUENTE);
}
```