

## Práctica #12 – Swing: Notepad pt3

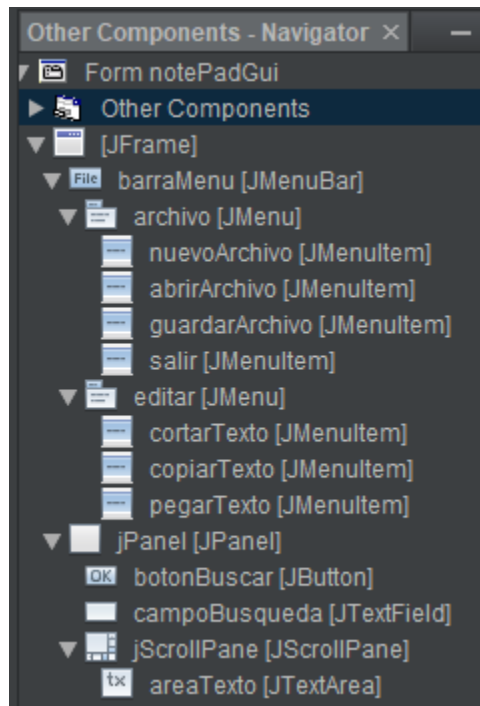


### RESULTADOS DE APRENDIZAJE:

---

Al finalizar esta sesión de práctica el estudiante deberá:

- ✓ Crear código que permita la captura de datos.
- ✓ Realizar operaciones de asignación de datos.
- ✓ Comprender el uso de métodos en Java.
- ✓ Realizar operaciones matemáticas en Java.



## NOTEPADGUI.JAVA

Agregar el siguiente código que dará funcionalidad al botón de buscar en nuestra app

```
package notepad;  
  
import java.awt.Color;  
import java.awt.FileDialog;  
import java.awt.datatransfer.Clipboard;  
import java.awt.datatransfer.DataFlavor;  
import java.awt.datatransfer.StringSelection;  
import java.awt.datatransfer.Transferable;  
import java.io.BufferedReader;  
import java.io.FileReader;  
import java.io.FileWriter;  
import java.io.IOException;  
import javax.swing.text.DefaultHighlighter;  
import javax.swing.text.Document;  
import javax.swing.text.Highlighter;  
import javax.swing.text.JTextComponent;  
  
public class notePadGui extends javax.swing.JFrame {  
  
    String nombreArchivo;  
    Clipboard clipboard = getToolkit().getSystemClipboard();  
    public notePadGui() {  
        initComponents();  
    }  
  
    @SuppressWarnings("unchecked")  
  
    private void botonBuscarActionPerformed(java.awt.event.ActionEvent evt) {  
  
        if(!"".equals(campoBusqueda.getText())) {  
            buscarAreaTexto(areaTexto, campoBusqueda.getText());  
        }else{  
            removerHighLight(areaTexto);  
        }  
    }  
}
```

```
}  
  
class resaltar extends DefaultHighlighter.DefaultHighlightPainter{  
    public resaltar(Color color){  
        super(color);  
    }  
}  
  
DefaultHighlighter.HighlightPainter highlighter = new resaltar(Color.yellow);  
  
public void removerHighLight(JTextComponent textoComp){  
    Highlighter removeHighlighter = textoComp.getHighlighter();  
    Highlighter.Highlight[] remover = removeHighlighter.getHighlights();  
  
    for(int i = 0; i<remover.length; i++){  
        if(remover[i].getPainter() instanceof resaltar){  
            removeHighlighter.removeHighlight(remover[i]);  
        }  
    }  
}  
  
public void buscarAreaTexto(JTextComponent textoComp, String textoString){  
    removerHighLight(textoComp);  
    try{  
        Highlighter hilight = textoComp.getHighlighter();  
        Document doc = textoComp.getDocument();  
        String texto = doc.getText(0, doc.getLength());  
  
        int pos = 0;  
  
        while((pos = texto.toUpperCase().indexOf(textoString.toUpperCase(),pos))>= 0){  
            hilight.addHighlight(pos, pos+textoString.length(), highlighter);  
            pos += textoString.length();  
        }  
    } catch (Exception e){  
    }  
}
```