



***UNIVERSIDAD POLITÉCNICA DE TLAXCALA***  
***“REGIÓN PONIENTE”***

***REPORTE CONTADOR DE PALABRAS***

***SISTEMAS INTELIGENTES***

***PROFA: ING. VANESA TENOPALA ZAVALA***

***INTREGRANTES DE EQUIPO:***

***INGRID MAGALI CRUZ MALDONADO***  
***ANTONIO OLVERA ESPINOZA***  
***DIANA LAURA PEÑA ROJAS***  
***RUBI JUAREZ GALLARDO***  
***LEO XELHUATZI LIRA***

***INGENIERIA: SISTEMAS COMPUTACIONALES***

***GRADO: 8to***

***GRUPO: “A”***

***ENERO – ABRIL 2024***



## **RESUMEN**

El contador de palabras es una aplicación de interfaz gráfica desarrollada en Java con el uso de la biblioteca Swing. El propósito principal de la aplicación es contar y resaltar tanto el número de palabras en un texto como la ocurrencia de ciertos conectores específicos en dicho texto. Además, proporciona funcionalidades para limpiar la pantalla de entrada y reiniciar los conteos.

## **INTRODUCCIÓN**

El programa presenta una interfaz gráfica de usuario construida con componentes de la biblioteca Swing. Ofrece un área de texto para ingresar un documento, dos botones para realizar acciones específicas (contar y limpiar), y etiquetas para mostrar los resultados de la cuenta de palabras y conectores. La aplicación se centra en la identificación y resaltado de conectores predefinidos en el texto, además de contar las palabras totales. Los conectores incluidos abarcan una variedad de términos y frases utilizados comúnmente en la construcción de conexiones lógicas y argumentativas.

## **DESARROLLO**

La funcionalidad principal del programa se encuentra bajo un método, este se utiliza para para contar las palabras en el texto ingresado. Además, identifica y resalta ciertos conectores en el texto, utilizando estilos para cambiar el color de dichos conectores a rojo en el componente JTextPane.

El conjunto de conectores a buscar está predefinido en el código y abarca una variedad de palabras clave y frases comúnmente utilizadas en la conexión de ideas dentro de un texto. La búsqueda de conectores se realiza sin distinción entre mayúsculas y minúsculas.



## CODIGO

```
import java.util.HashSet;
import java.util.Set;
import java.util.StringTokenizer;

private void contarPalabrasYConectores() {
    String texto = textoInput.getText();

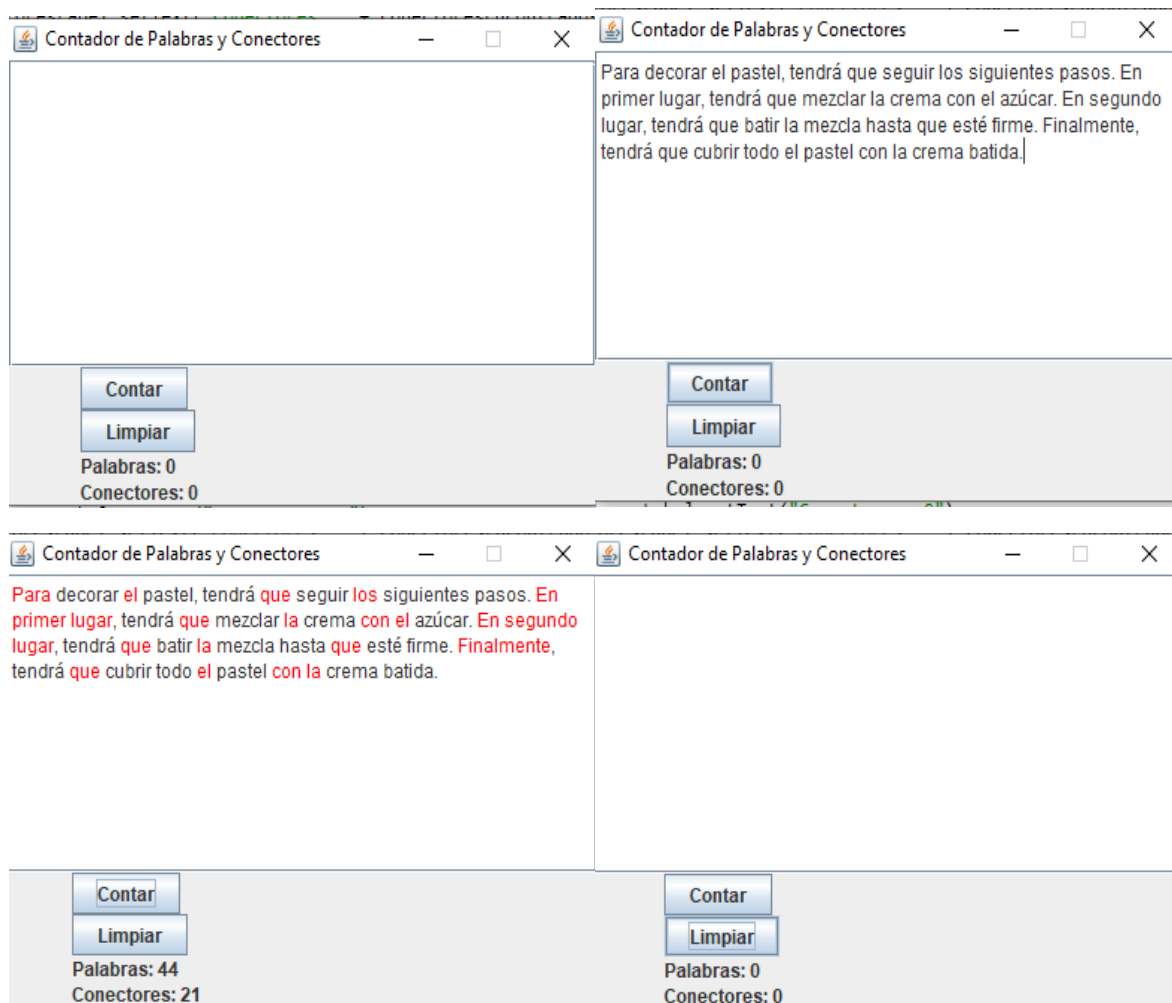
    // Contar palabras
    StringTokenizer tokenizer = new
StringTokenizer(texto, " \t\n\r\f.,;:!?'\\"()[]{}<>*&^%$#@+-
=|`~\\\/");
    int palabras = tokenizer.countTokens();
    resultadoPalabrasLabel.setText("Palabras: " +
palabras);

    Set<String> conectores = new HashSet<>(Arrays.asList(
        "y", "de", "es", "una", "es
decir", "lo", "la", "al", "del", "se", "o", "pero", "sin embargo",
"además", "por lo tanto", "en conclusión",
        "por último", "finalmente", "en resumen", "en
definitiva", "en fin",
        "para que", "a fin de que", "con el propósito
de", "con el fin de", "con el objetivo de",
        "asimismo", "igualmente", "de igual manera",
"adicionalmente", "también",
        "por otro lado", "por otra parte", "por otro
parte", "así", "entonces",
        "por consiguiente", "así que", "por lo
tanto", "por ende", "por eso",
        "porque", "ya que", "pues", "debido a que",
"a causa de",
        "por ejemplo", "por instancia", "como", "tal
como", "según",
        "comparado con", "a diferencia de", "al
contrario de",
        "primero", "en primer lugar", "segundo", "en
segundo lugar", "finalmente",
        "para", "al mismo tiempo", "posteriormente",
```



```
"anteriormente", "luego",  
    "por consiguiente", "en cambio", "de otro  
modo", "aunque", "si bien", "el", "en", "un", "los", "las",  
    "les", "que", "con"  
    ));  
// Contar total de conectores encontrados  
    int conectoresEncontrados = 0;  
    for (String conector : conectores) {  
        int indice = -1;  
        while ((indice =  
texto.toLowerCase().indexOf(conector.toLowerCase(), indice +  
1)) != -1) {  
            if (isPalabraCompleta(texto, indice,  
conector.length())) {  
                conectoresEncontrados++;  
            }  
        }  
    }  
  
    resultadoConectoresLabel.setText("Conectores: " +  
conectoresEncontrados);  
}  
  
    private boolean isPalabraCompleta(String texto, int  
indice, int longitudConector) {  
        // Verificar que el conector no esté dentro de otra  
palabra  
        if (indice > 0 &&  
Character.isLetterOrDigit(texto.charAt(indice - 1))) {  
            return false;  
        }  
  
        if (indice + longitudConector < texto.length() &&  
Character.isLetterOrDigit(texto.charAt(indice +  
longitudConector))) {  
            return false;  
        }  
  
        return true;  
    }  
}
```

## EJECUCIÓN



## CONCLUSIÓN

El programa proporciona una solución interactiva para el análisis de textos, ofreciendo una herramienta útil para contar palabras y destacar conectores importantes. Su interfaz gráfica facilita la interacción del usuario, permitiendo una rápida visualización de los resultados. Además, la capacidad de limpiar la pantalla proporciona una funcionalidad conveniente para el usuario. En general, el "ContadorPalabras" es una aplicación eficiente para realizar análisis básicos de texto.