

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class PrincipalSO extends JApplet implements ActionListener {
    // Declaración global
    JLabel    lblClave, lblEncriptar,
              lblEncriptado, lblTitulo, lblEncriptado2, lblEncriptado3;
    JTextField txtClave;
    JButton    btnProcesar, btnCancela, btnDesencripta, btnEnvia, btnLimpia;
    JTextArea  txtEncriptar, txtEncriptado, txtEncriptado2, txtEncriptado3;
    JScrollPane scpScroll1, scpScroll2, scpScroll22, scpScroll23;
    int x=0;
    int    FILAS=0, COLUMNAS=0;
    private char  n[][];
    String  o[][];
    int canTex = 0;
    String original="";
    String mensaenrip="";
    String claveencriptada="";

    // Crea la interfaz gráfica de usuario: GUI
    public void init() {
        setLayout(null);
        setSize(640,480);
        setBackground(Color.lightGray);

        lblTitulo = new JLabel("Algoritmo de Transposición por
Sustitución",JLabel.CENTER);
        lblTitulo.setBackground(new Color(226,223,234));
        lblTitulo.setForeground(Color.black);
        lblTitulo.setOpaque(true);
        lblTitulo.setFont(new Font("dialog",3,23));
        lblTitulo.setBounds(0,10,640,30);
        add(lblTitulo);

        lblEncriptar = new JLabel("Texto a Encriptar:");
        lblEncriptar.setBounds(10,100,100,30);
        add(lblEncriptar);

        txtEncriptar = new JTextArea();
        txtEncriptar.setFont(new Font("Arial",Font.BOLD,13));
        scpScroll1 = new JScrollPane(txtEncriptar);
        scpScroll1.setBounds(120,100,180,80);
        add(scpScroll1);
```

```
lblClave = new JLabel("Clave :");
lblClave.setBounds(10,50,100,20);
add(lblClave);

txtClave = new JTextField();
txtClave.setFont(new Font("Arial",Font.BOLD,13));
txtClave.setBounds(120,50,180,30);
add(txtClave);

btnProcesar = new JButton("Encripta");
btnProcesar.setBounds(120,190,90,30);
btnProcesar.addActionListener(this);
add(btnProcesar);

btnLimpia = new JButton("Limpiar");
btnLimpia.setBounds(380,410,90,30);
btnLimpia.addActionListener(this);
add(btnLimpia);

btnCancela = new JButton("Cancelar");
btnCancela.setBounds(210,190,90,30);
btnCancela.addActionListener(this);
add(btnCancela);

lblEncriptado = new JLabel("Texto Encriptado:");
lblEncriptado.setBounds(400,40,100,30);
add(lblEncriptado);

txtEncriptado = new JTextArea();
txtEncriptado.setFont(new Font("Arial",Font.BOLD,13));
txtEncriptado.setEditable(false);
txtEncriptado.setBackground(Color.lightGray);
txtEncriptado.setForeground(Color.blue);
scpScroll2 = new JScrollPane(txtEncriptado);
scpScroll2.setBounds(380,80,200,100);
add(scpScroll2);

btnEnvia = new JButton("Enviar");
btnEnvia.setBounds(400,190,90,30);
btnEnvia.setVisible(false);
btnEnvia.addActionListener(this);
add(btnEnvia);

lblEncriptado2 = new JLabel("Texto Recibido");
lblEncriptado2.setBounds(130,270,100,30);
add(lblEncriptado2);

txtEncriptado2 = new JTextArea();
txtEncriptado2.setFont(new Font("Arial",Font.BOLD,13));
```

```
txtEncriptado2.setEditable(false);
txtEncriptado2.setBackground(Color.lightGray);
txtEncriptado2.setForeground(Color.blue);
scpScroll22 = new JScrollPane(txtEncriptado2);
scpScroll22.setBounds(100,300,200,100);
add(scpScroll22);

btnDesencrypta = new JButton("Desencrypta");
btnDesencrypta.setBounds(100,410,120,30);
btnDesencrypta.addActionListener(this);
btnDesencrypta.setVisible(false);
add(btnDesencrypta);

lblEncriptado3 = new JLabel("Mensaje Decifrado");
lblEncriptado3.setBounds(390,270,120,30);
add(lblEncriptado3);

txtEncriptado3 = new JTextArea();
txtEncriptado3.setFont(new Font("Arial",Font.BOLD,13));
txtEncriptado3.setEditable(false);
txtEncriptado3.setBackground(Color.lightGray);
txtEncriptado3.setForeground(Color.blue);
scpScroll23 = new JScrollPane(txtEncriptado3);
scpScroll23.setBounds(380,300,200,100);
add(scpScroll23);

}

// Procesa eventos de tipo ActionEvent
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == btnProcesar)
        procesar();

    if (e.getSource() == btnDesencrypta)
        desencrypta();

    if (e.getSource() == btnCancela)
        borrar();

    if (e.getSource() == btnEnvia)
        envia();

    if (e.getSource() == btnLimpia)
        limpiatodo();
}

void envia(){
    imprimir("Mensaje Enviado....");
    btnEnvia.setVisible(false);
}
```

```
        imprimir2(mensaenrip);
        btnDesencrypta.setVisible(true);
    }

    void procesar(){

        x=0;
        n = new char[0][0];

        String clave = txtClave.getText();
        original=clave;
        String texto = txtEncriptar.getText();

        if(!clave.equals("") && !texto.equals("")){
            canTex = texto.length();
            COLUMNAS = clave.length();
            FILAS = (canTex/( clave.length()+1));

            if(canTex % COLUMNAS != 0)
                FILAS = FILAS + 1;

            if(FILAS==1){
                FILAS=FILAS+1;
            }
            o = new String[FILAS][COLUMNAS];
            n = new char[FILAS][COLUMNAS];
            llenar(texto,FILAS,COLUMNAS,n);
            // listar(FILAS,COLUMNAS,n);

            ordenarEnAscenso();
            btnEnvia.setVisible(true);

        }
        else
            imprimir("ingrese mensaje");
    }

    void borrar(){
        txtClave.setText("");
        txtEncriptado.setText("");
        txtEncriptar.setText("");
    }

    void desencrypta(){
        /*txtClave.setText("");
        txtEncriptado.setText("");
        txtEncriptar.setText("");*/
        descriptar(original);
        btnDesencrypta.setVisible(false);
    }
}
```

```

    }
    void limpiatodo(){
        txtClave.setText("");
        txtEncriptado.setText("");
        txtEncriptar.setText("");
        txtEncriptado2.setText("");
        txtEncriptado3.setText("");
    }

    void llenar(String t,int f, int c, char[][] n){

        for (int i=0; i<f;i++)
            for (int j=0; j<c; j++){
                if (i==0){ // en esta parte ingresal arreglo la clave para
que valla como cabezara
                    for (int r=0; r<c; r++)
                        n[i][r] = txtClave.getText().charAt(r);
                    i++;
                }
                if(x<canTex) //el if es para que cuando llegue al final de
la cadena llene con
registros del arreglo que faltan
                    n[i][j] = t.charAt(x); //espacios en blanco los
                    else
                        n[i][j] = ' '; //ese era el error
                    x = x+1;
                }
            }
    }

    public void listar(int f, int c,char[][] a) {
    for (int i=0; i<f; i++) {
        for (int j=0; j<c; j++)
            txtEncriptado.append("n[" + i + "][" + j + "] : " +a[i][j] + "\t");
            txtEncriptado.append("\n");
        }
        imprimir("");
    }

    private void imprimir(String s) {
        txtEncriptado.append(s + "\n");
    }

    private void imprimir2(String s) {
        txtEncriptado2.append(s + "\n");
    }
}

```

```

        private void imprimir3(String s) {
            txtEncriptado3.append(s + "\n");
        }

    void ordenarEnAscenso() {
        String aux[][];
        String q[][];

        q = new String[FILAS][COLUMNAS]; //temporal para comparar cada
casilla datos en string nos facilitu usar numeros y letras
        aux = new String[FILAS][COLUMNAS]; // para guarda datos antes de
mover
        for (int a=0; a<COLUMNAS; a++) //lleno el arreglo q que es string con
datos n para comparar
            for(int b=0; b<FILAS; b++)
                q[b][a]=n[b][a]+"";
            for (int i=0;i<COLUMNAS-1; i++) {
                for (int j=i+1; j<COLUMNAS; j++){
                    if (q[0][i].compareTo(q[0][j]) > 0) {
                        for(int m=0;m<FILAS;m++){
                            aux[m][i] = q[m][i];
                            q[m][i] = q[m][j];
                            q[m][j] = aux[m][i];
                        }
                    }
                }
            }

        String mensaje="";
        String cla="";
        String letra="";
        for (int a=0; a<COLUMNAS; a++) //lleno el arreglo q que es string con
datos n para comparar
            for(int b=1; b<FILAS; b++){
                letra=q[b][a];
                // if(!letra.equals(' '))
                mensaje=mensaje+letra;
            }
        mensaenrip=mensaje;
// lista(FILAS,COLUMNAS,q);
        imprimir(mensaje);
        o=q;

        //para sacar la clave
        for (int a=0; a<COLUMNAS; a++)
            letra=q[0][a];
            cla=cla+letra;
            claveencriptada=cla;
    }

```

```

    }

    public void lista(int f, int c,String[][] a) {
    for (int i=0; i<f; i++) {
        for (int j=0; j<c; j++)
            txtEncriptado.append("n[" + i + "]"[" + j + "] : " +a[i][j] + "\t");
        txtEncriptado.append("\n");
    }
    imprimir("");
}

void desencriptar(String orig){

    char u=' ';
    String u2="";
    String p[][];
    p= new String[FILAS][COLUMNAS];

        for (int a=0; a<COLUMNAS; a++) {
            u=orig.charAt(a);
            u2=u+"";
            for(int v=0;v<COLUMNAS; v++){
                if(o[0][v].equals(u2)){
                    for(int n=0;n<FILAS; n++){
                        p[n][a]=o[n][v];
                    }
                }
            }
        }

    //lista(FILAS,COLUMNAS,p);

    String mensaje="";
    String letra="";
    for(int b=1; b<FILAS; b++)
        for (int a=0; a<COLUMNAS; a++){
            letra=p[b][a];
            //if(!letra.equals(' '))
            mensaje=mensaje+letra;
        }

    imprimir3 ("Clave :"+orig);
    imprimir3 ("texto :"+mensaje);
}

}

```