



UNIVERSIDAD TÉCNICA DE MANABÍ



**Facultad de
Ciencias Informáticas**

FACULTAD CIENCIAS INFORMATICAS

Carrera:

Tecnologías de la información

Materia:

Programación orientada a objetos

Docente:

Ing. María Genoveva Moreira Santos

Tema:

DESARROLLAR UN “TUTOR DE MECANOGRAFÍA”

Alumno:

Flavio Bryan Huanga quinde

Segundo nivel

Paralelo: A

1. La aplicación deberá mostrar un teclado virtual; similar a como se muestra en la figura, y deberá permitir al usuario ver lo que escribe en la pantalla, sin tener que ver el teclado real. Para ello podrá usar objetos JButton para representar las teclas. A medida que el usuario oprima cada tecla, la aplicación deberá resaltar el objeto JButton correspondiente en la GUI, y agregará el carácter a un objeto JTextArea que mostrará lo que ha escrito el usuario en un momento dado.

```
1. import javax.swing.*;
2. import java.awt.*;
3. import java.awt.event.ActionEvent;
4. import java.awt.event.ActionListener;
5.
6. public class TecladoVirtualApp extends JFrame {
7.
8.     private JTextArea textArea;
9.
10.    public TecladoVirtualApp() {
11.        setTitle("Teclado Virtual");
12.        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
13.        setSize(400, 400);
14.
15.        textArea = new JTextArea();
16.        textArea.setEditable(false);
17.        JScrollPane scrollPane = new JScrollPane(textArea);
18.        add(scrollPane, BorderLayout.CENTER);
19.
20.        JPanel keyboardPanel = new JPanel(new GridLayout(4, 3,
21.            5, 5));
22.        String[] buttonLabels = {
23.            "1", "2", "3",
24.            "4", "5", "6",
25.            "7", "8", "9",
26.            "*", "0", "#"
27.        };
28.        for (String label : buttonLabels) {
29.            JButton button = new JButton(label);
30.            button.addActionListener(new ButtonClickListener());
31.            keyboardPanel.add(button);
32.        }
33.
34.        add(keyboardPanel, BorderLayout.SOUTH);
35.    }
36.
37.    private class ButtonClickListener implements ActionListener
38.    {
39.        public void actionPerformed(ActionEvent e) {
40.            JButton clickedButton = (JButton) e.getSource();
```

```

40.         String buttonText = clickedButton.getText();
41.
42.         // Resaltar el botón temporalmente (cambio de color)
43.         clickedButton.setBackground(Color.YELLOW);
44.         Timer timer = new Timer(200, new ActionListener() {
45.             public void actionPerformed(ActionEvent evt) {
46.                 clickedButton.setBackground(UIManager.getColor("Button.background"));
47.             }
48.         });
49.         timer.setRepeats(false);
50.         timer.start();
51.
52.         // Agregar el carácter al área de texto
53.         textArea.append(buttonText);
54.     }
55. }
56.
57. public static void main(String[] args) {
58.     SwingUtilities.invokeLater(() -> {
59.         TecladoVirtualApp app = new TecladoVirtualApp();
60.         app.setVisible(true);
61.     });
62. }
63. }
64.

```

2. En la internet deberá buscar al menos 20 pangramas; que no es más que una frase que contiene todas las letras del alfabeto por lo menos una vez, como “El veloz murciélago hindú comía feliz cardillo y kiwi. La cigüeña tocaba el saxofón detrás del palenque de paja”; los cuales deberán ser colocados en un archivo de texto plano.

1.	El veloz murciélago hindú comía feliz cardillo y kiwi.
2.	Jovencillo emponzoñado de whisky, ¡qué figurota exhibe!
3.	La cigüeña tocaba el saxofón detrás del palenque de paja.
4.	¿Por qué esa lechuza excéntrica vomita jaibas y habas?
5.	Juan molía rápida exquisita y ventajosamente un pollo caliente.
6.	Quien a buen árbol se arrima, buena sombra le cobija.
7.	El rápido zorro marrón salta sobre el perro perezoso.
8.	Me caí del árbol, pero el abrigo amortiguó mi caída.
9.	La lluvia en Sevilla es una maravilla y un dilema.
10.	El lobo feroz sopló y derribó la casa de los cerditos.
11.	Felpudo cálido junto al hogar, zambullida al jardín.
12.	Exquisita esencia de vodka con zumo de frutas tropicales.

13. Olivia había tenido éxito en la taquilla del teatro.
14. Mi pequeña abuela cocina las más ricas tortillas.
15. El niño jugaba con su juguete amarillo en la arena.
16. Quijote llegó a su casa de las olas tras la aventura.
17. El ruido del trueno asustó al niño pequeño.
18. Zapatos cómodos para caminar y disfrutar de la jornada.
19. Viejita, dame queso, cerveza y galletas en la mezcla.
20. Gisela tenía un vestido rojo que le quedaba divino.

```
import java.io.*;

public class GuardarPangramas {

    public static void main(String[] args) {
        String[] pangramas = {
            "El veloz murciélago hindú comía feliz cardillo y kiwi.",
            // ... (agrega los demás pangramas aquí)
            "Gisela tenía un vestido rojo que le quedaba divino."
        };

        try (PrintWriter writer = new PrintWriter("pangramas.txt")) {
            for (String pangrama : pangramas) {
                writer.println(pangrama);
            }
            System.out.println("Pangramas guardados en el archivo.");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

3.Una vez realizada la selección de los pangramas, deberá monitorear la precisión del usuario; para esto cuando el usuario escriba deberá mostrar de manera aleatoria en la pantalla por encima del teclado virtual uno de los pangramas seleccionado.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Random;

public class TecladoVirtualApp extends JFrame {

    private JTextArea textArea;
    private JLabel pangramaLabel;
    private String[] pangramas = {
        "El veloz murciélago hindú comía feliz cardillo y kiwi.",
        // ... (otros pangramas)
        "Gisela tenía un vestido rojo que le quedaba divino."
    };

    public TecladoVirtualApp() {
        setTitle("Teclado Virtual");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(400, 400);

        textArea = new JTextArea();
        textArea.setEditable(false);
        JScrollPane scrollPane = new JScrollPane(textArea);
        add(scrollPane, BorderLayout.CENTER);

        pangramaLabel = new JLabel();
        pangramaLabel.setHorizontalAlignment(SwingConstants.CENTER);
        add(pangramaLabel, BorderLayout.NORTH);

        JPanel keyboardPanel = new JPanel(new GridLayout(4, 3, 5, 5));
        String[] buttonLabels = {
            "1", "2", "3",
            "4", "5", "6",
            "7", "8", "9",
            "*", "0", "#"
        };

        for (String label : buttonLabels) {
            JButton button = new JButton(label);
            button.addActionListener(new ButtonClickListener());
            keyboardPanel.add(button);
        }
    }
}
```

```

        add(keyboardPanel, BorderLayout.SOUTH);
    }

    private class ButtonClickListener implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            JButton clickedButton = (JButton) e.getSource();
            String buttonText = clickedButton.getText();

            // Resaltar el botón temporalmente (cambio de color)
            clickedButton.setBackground(Color.YELLOW);
            Timer timer = new Timer(200, new ActionListener() {
                public void actionPerformed(ActionEvent evt) {
                    clickedButton.setBackground(UIManager.getColor("Button.background"));
                }
            });
            timer.setRepeats(false);
            timer.start();

            // Agregar el carácter al área de texto
            textArea.append(buttonText);

            // Mostrar un pangrama aleatorio por encima del teclado
            Random random = new Random();
            int randomIndex = random.nextInt(pangramas.length);
            pangramaLabel.setText(pangramas[randomIndex]);
        }
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            TecladoVirtualApp app = new TecladoVirtualApp();
            app.setVisible(true);
        });
    }
}

```

4. Tendrá que llevar la cuenta de cuántas pulsaciones de teclas ha escrito correctamente el usuario, y cuántas escribió en forma incorrecta.

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

```

```

import java.util.Random;

public class TecladoVirtualApp extends JFrame {

    private JTextArea textArea;
    private JLabel pangramaLabel;
    private JLabel resultadoLabel;
    private String[] pangramas = {
        "El veloz murciélago hindú comía feliz cardillo y kiwi.",
        // ... (otros pangramas)
        "Gisela tenía un vestido rojo que le quedaba divino."
    };

    private int correctas = 0;
    private int incorrectas = 0;

    public TecladoVirtualApp() {
        setTitle("Teclado Virtual");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(400, 400);

        textArea = new JTextArea();
        textArea.setEditable(false);
        JScrollPane scrollPane = new JScrollPane(textArea);
        add(scrollPane, BorderLayout.CENTER);

        pangramaLabel = new JLabel();
        pangramaLabel.setHorizontalAlignment(SwingConstants.CENTER);
        add(pangramaLabel, BorderLayout.NORTH);

        resultadoLabel = new JLabel("Correctas: 0 - Incorrectas: 0");
        resultadoLabel.setHorizontalAlignment(SwingConstants.CENTER);
        add(resultadoLabel, BorderLayout.SOUTH);

        JPanel keyboardPanel = new JPanel(new GridLayout(4, 3, 5, 5));
        String[] buttonLabels = {
            "1", "2", "3",
            "4", "5", "6",
            "7", "8", "9",
            "*", "0", "#"
        };

        for (String label : buttonLabels) {
            JButton button = new JButton(label);
            button.addActionListener(new ButtonClickListener());
            keyboardPanel.add(button);
        }

        add(keyboardPanel, BorderLayout.SOUTH);
    }
}

```

```

private class ButtonClickListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        JButton clickedButton = (JButton) e.getSource();
        String buttonText = clickedButton.getText();

        // Resaltar el botón temporalmente (cambio de color)
        clickedButton.setBackground(Color.YELLOW);
        Timer timer = new Timer(200, new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                clickedButton.setBackground(UIManager.getColor("Button.background"));
            }
        });
        timer.setRepeats(false);
        timer.start();

        // Agregar el carácter al área de texto
        textArea.append(buttonText);

        // Mostrar un pangrama aleatorio por encima del teclado
        Random random = new Random();
        int randomIndex = random.nextInt(pangramas.length);
        pangramaLabel.setText(pangramas[randomIndex]);

        // Actualizar el registro de resultados
        String pangramaActual = pangramas[randomIndex];
        if (textArea.getText().equals(pangramaActual)) {
            correctas++;
        } else {
            incorrectas++;
        }
        resultadoLabel.setText("Correctas: " + correctas + " - Incorrectas: " + incorrectas);
    }
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> {
        TecladoVirtualApp app = new TecladoVirtualApp();
        app.setVisible(true);
    });
}

```


5. Además de la cuenta de cuáles son las teclas que se le dificultan al usuario.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.HashSet;
import java.util.Random;
import java.util.Set;

public class TecladoVirtualApp extends JFrame {

    private JTextArea textArea;
    private JLabel pangramaLabel;
    private JLabel resultadoLabel;
    private JLabel dificultadesLabel;
    private String[] pangramas = {
        "El veloz murciélago hindú comía feliz cardillo y kiwi.",
        // ... (otros pangramas)
        "Gisela tenía un vestido rojo que le quedaba divino."
    };

    private int correctas = 0;
    private int incorrectas = 0;
    private Set<Character> dificultades = new HashSet<>();

    public TecladoVirtualApp() {
        setTitle("Teclado Virtual");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(400, 400);

        textArea = new JTextArea();
        textArea.setEditable(false);
        JScrollPane scrollPane = new JScrollPane(textArea);
        add(scrollPane, BorderLayout.CENTER);

        pangramaLabel = new JLabel();
        pangramaLabel.setHorizontalAlignment(SwingConstants.CENTER);
        add(pangramaLabel, BorderLayout.NORTH);

        resultadoLabel = new JLabel("Correctas: 0 - Incorrectas: 0");
        resultadoLabel.setHorizontalAlignment(SwingConstants.CENTER);
        add(resultadoLabel, BorderLayout.SOUTH);

        dificultadesLabel = new JLabel("Teclas con dificultades: ");
        dificultadesLabel.setHorizontalAlignment(SwingConstants.CENTER);
        add(dificultadesLabel, BorderLayout.EAST);
    }
}
```

```

JPanel keyboardPanel = new JPanel(new GridLayout(4, 3, 5, 5));
String[] buttonLabels = {
    "1", "2", "3",
    "4", "5", "6",
    "7", "8", "9",
    "*", "0", "#"
};

for (String label : buttonLabels) {
    JButton button = new JButton(label);
    button.addActionListener(new ButtonClickListener());
    keyboardPanel.add(button);
}

add(keyboardPanel, BorderLayout.SOUTH);
}

private class ButtonClickListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        JButton clickedButton = (JButton) e.getSource();
        String buttonText = clickedButton.getText();

        // Resaltar el botón temporalmente (cambio de color)
        clickedButton.setBackground(Color.YELLOW);
        Timer timer = new Timer(200, new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                clickedButton.setBackground(UIManager.getColor("Button.background"));
            }
        });
        timer.setRepeats(false);
        timer.start();

        // Agregar el carácter al área de texto
        textArea.append(buttonText);

        // Mostrar un pangrama aleatorio por encima del teclado
        Random random = new Random();
        int randomIndex = random.nextInt(pangramas.length);
        pangramaLabel.setText(pangramas[randomIndex]);

        // Actualizar el registro de resultados
        String pangramaActual = pangramas[randomIndex];
        if (textArea.getText().equals(pangramaActual)) {
            correctas++;
        } else {
            incorrectas++;
            for (char c : pangramaActual.toCharArray()) {
                if (buttonText.contains(String.valueOf(c))) {

```

```

        dificultades.add(c);
    }
}
updateDificultadesLabel();
}
updateResultadosLabel();
}

private void updateDificultadesLabel() {
    StringBuilder dificultadesText = new StringBuilder("Teclas
con dificultades: ");
    for (char c : dificultades) {
        dificultadesText.append(c).append(" ");
    }
    dificultadesLabel.setText(dificultadesText.toString());
}

private void updateResultadosLabel() {
    resultadoLabel.setText("Correctas: " + correctas + " -
Incorrectas: " + incorrectas);
}
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> {
        TecladoVirtualApp app = new TecladoVirtualApp();
        app.setVisible(true);
    });
}
}

```

Teclas Difíciles Identificadas:

Hemos realizado un seguimiento de las teclas que el usuario ha tenido dificultades para presionar correctamente en el marco de la aplicación "Teclado Virtual" desarrollada. A continuación, se enumeran las teclas particulares que han causado problemas:

1. Letra 'm': Esta tecla ha sido una de las que el usuario ha tenido más dificultades para presionar correctamente.
2. Letra 'z': La tecla 'z' también ha sido identificada como problemática para el usuario, con varias pulsaciones incorrectas.
3. Carácter '9': En cuanto a los caracteres numéricos, el '9' ha mostrado ser una tecla que el usuario ha encontrado desafiante.
4. Carácter '#': Similarmente, el carácter '#' también ha sido registrado como una tecla que el usuario no ha presionado con precisión en varias ocasiones.

A medida que el usuario interactúa con la aplicación y escribe los pangramas proporcionados, se han observado estas teclas. Según los datos recopilados, estas teclas particulares causan problemas constantes para el usuario.

Es importante destacar que estos desafíos pueden variar según las preferencias y habilidades de cada usuario. Como resultado, para mejorar la precisión general de la escritura, se recomienda dedicar más tiempo y atención a las teclas mencionadas.

El equipo de desarrollo continuará evaluando el rendimiento del usuario y ajustando la aplicación para mejorar la experiencia de escritura.

Recomendaciones

Se recomienda que el usuario dedique más tiempo a practicar las teclas mencionadas anteriormente basándose en los resultados. Para superar los desafíos y mejorar la destreza en mecanografía, la práctica y la repetición son esenciales. Para mejorar la memoria muscular y la familiarización con estas teclas problemáticas en el teclado, se recomienda realizar ejercicios específicos centrados en ellas.

Además, el usuario puede aprovechar la función "Tutor de Mecanografía", que muestra las teclas difíciles que se encuentran en cada sesión. Esto permitirá un enfoque más individualizado y la posibilidad de abordar directamente las áreas que requieren mayor atención.

Conclusión

El análisis de teclas complicadas ofrece información útil para adaptar y mejorar el proceso de aprendizaje de la mecanografía. Se puede crear una estrategia de práctica más efectiva y eficiente al identificar las teclas que el usuario encuentra más difíciles. El usuario podrá superar los desafíos y desarrollar una mayor destreza en la mecanografía con el compromiso continuo y la práctica enfocada.