

# Aplicación Troll

Enlace al proyecto en GitHub:

[https://github.com/jesusordogomez/Workspace\\_DAM\\_II/tree/dc910b388e3065561a004f7ad746330335ddb179/Troll](https://github.com/jesusordogomez/Workspace_DAM_II/tree/dc910b388e3065561a004f7ad746330335ddb179/Troll)

El propósito de esta aplicación es trollear al usuario creando una ventana donde no pueda cerrar directamente haciendo clic en el botón "X" de la esquina superior derecha. En su lugar, el usuario debe hacer clic en un botón que se mueve de manera aleatoria por la ventana antes de poder cerrarla.

Esta interfaz no era lo primero que tenía en mente, había pensado en abrir una ventana y que al cerrarla salieran varias ventanas por la pantalla, también había pensado crear una ventana y hacer que se moviera aleatoriamente por la pantalla, al final, me decidí por mi proyecto.

Creo que el proyecto es sencillo, pero es sencillo, divertido y eficaz para poder trollear al usuario.

## CÓDIGO PROYECTO

```
1 package Ejercicio;
2
3 import javax.swing.*;
4
5
6
7
8
9
10
11 public class troleo {
12
13     private JFrame frame;
14     JButton boton;
15
16     /**
17      * Launch the application.
18      */
19     public static void main(String[] args) {
20         EventQueue.invokeLater(new Runnable() {
21             public void run() {
22                 try {
23                     troleo window = new troleo();
24                     window.frame.setVisible(true);
25                 } catch (Exception e) {
26                     e.printStackTrace();
27                 }
28             }
29         });
30     }
31
32     /**
33      * Create the application.
34      */
35     public troleo() {
36         initialize();
37     }
38 }
```

1. Creo la clase a través de Application Window, para que se inicialice directamente la interfaz gráfica.
2. Dentro de la clase troleo, he creado inicializado un frame y un botón.

```
private void initialize() {
    //creo el jframe
    frame = new JFrame();
    frame.setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE); //aquí hago que la ventana no se pueda cerrar dándole a la x
    //le doy un color y unas dimensiones específicas
    frame.getContentPane().setBackground(new Color(87, 87, 87));
    frame.setPreferredSize(new Dimension(600, 600));
    frame.setResizable(false); //con el método resizable desactivo la capacidad de cambiar la dimensión de la ventana
    frame.pack(); //con el pack me aseguro de que no se pueda redimensionar la ventana
    frame.setVisible(true);
    frame.setTitle("Troleo");
    frame.getContentPane().setLayout(null);
    frame.setLocationRelativeTo(null);
}
```

3. En el método initialize, creo el frame y le doy color, dimensiones, lo hago visible y le pongo un título.
4. Con el método .setDefaultCloseOperation, le digo que no se cierre si el usuario quiere cerrarlo dando a la X.
5. Con el método .setLocationRelativeTo(null), hago que el frame siempre esté en medio de la pantalla.

```

boton = new JButton();
boton.setBounds(222, 194, 150, 50);
boton.setText("¿Jugamos?");
boton.setBackground(new Color(202, 202, 202));
frame.getContentPane().add(boton);

```

6. Creo el botón con su texto, color, dimensiones y lo agrego al panel.

```

JTextArea TextArea = new JTextArea();
TextArea.setEnabled(false);
TextArea.setFont(new Font("Monospaced", Font.BOLD, 14));
TextArea.setBounds(176, 40, 267, 50);
frame.getContentPane().add(TextArea);
TextArea.setBackground(new Color(87, 87, 87));
TextArea.setText("PARA CERRAR TIENES QUE PULSAR");

```

7. Creo un TextArea con un tipo de letra, una dimensión, un color y le escribo un texto con el que ya interactuo con el usuario.

```

//con el metodo mouseListener implemento un par de metodos para manejar el raton
boton.addMouseListener( new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        // TODO Auto-generated method stub
        frame.dispose();//con el dispose() lo que hace que cuando cliques el boton de la ventana se pueda cerrar e
    }

    @Override
    public void mouseEntered(MouseEvent e) {///con este metodo interactuo con el raton y hago un random para que el
        // TODO Auto-generated method stub
        Random r = new Random();//
        boton.setBounds(r.nextInt(0,350),r.nextInt(0,350), 75, 40);
        boton.setText("Salir");
    }
}

```

8. El método MouseListener lo creo para que pueda implementar lo que yo necesito. Cuando añado el método, se me crean varios métodos distintos con lo que le puedes ir agregando acciones para el ratón.
9. Utilizo el método mouseClicked para cuando el usuario acierte a darle al botón, la ventana pueda cerrarse.
10. El método mouseEntered, lo utilizo para que cuando el ratón entre en contacto con el botón, se genere un movimiento random dentro de la ventana y dicho botón se mueva para que el usuario tenga difícil poder cerrar la ventana.

En resumen, el código es sencillo pero práctico, a la vez que fácil de entender para llevarlo a cabo. Las dificultades que he tenido, primeramente fueron las implementaciones que quería hacer para otro tipo de troleo, no conseguía hacer lo que yo tenía en mente. Con el código que utilizo en la tarea, el problema principal era que lo que tenía en mente de generar el movimiento del botón, no sabía como hacerlo aunque al final buscando lo pude conseguir. Una vez conseguido el movimiento, ahora tenía que saber qué dimensión poner al frame, porque sinceramente era complicado pulsar el botón en una ventana tan grande, ya que también el botón se movía a puntos más lejanos.