

## APLICACIÓN TROLL

Link del repositorio:

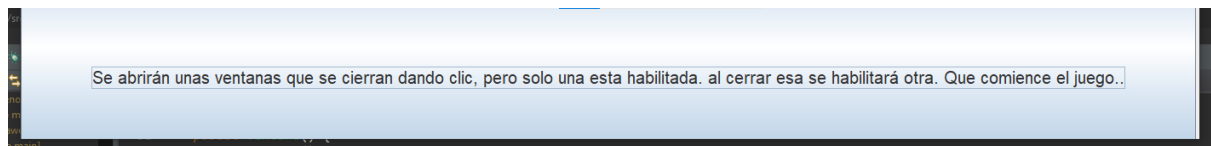
<https://github.com/JesGuiVil/Principal/tree/34364c4c38abe5dcbc80c108c768b1c19be1673a/trollwindow>

Branch:

main

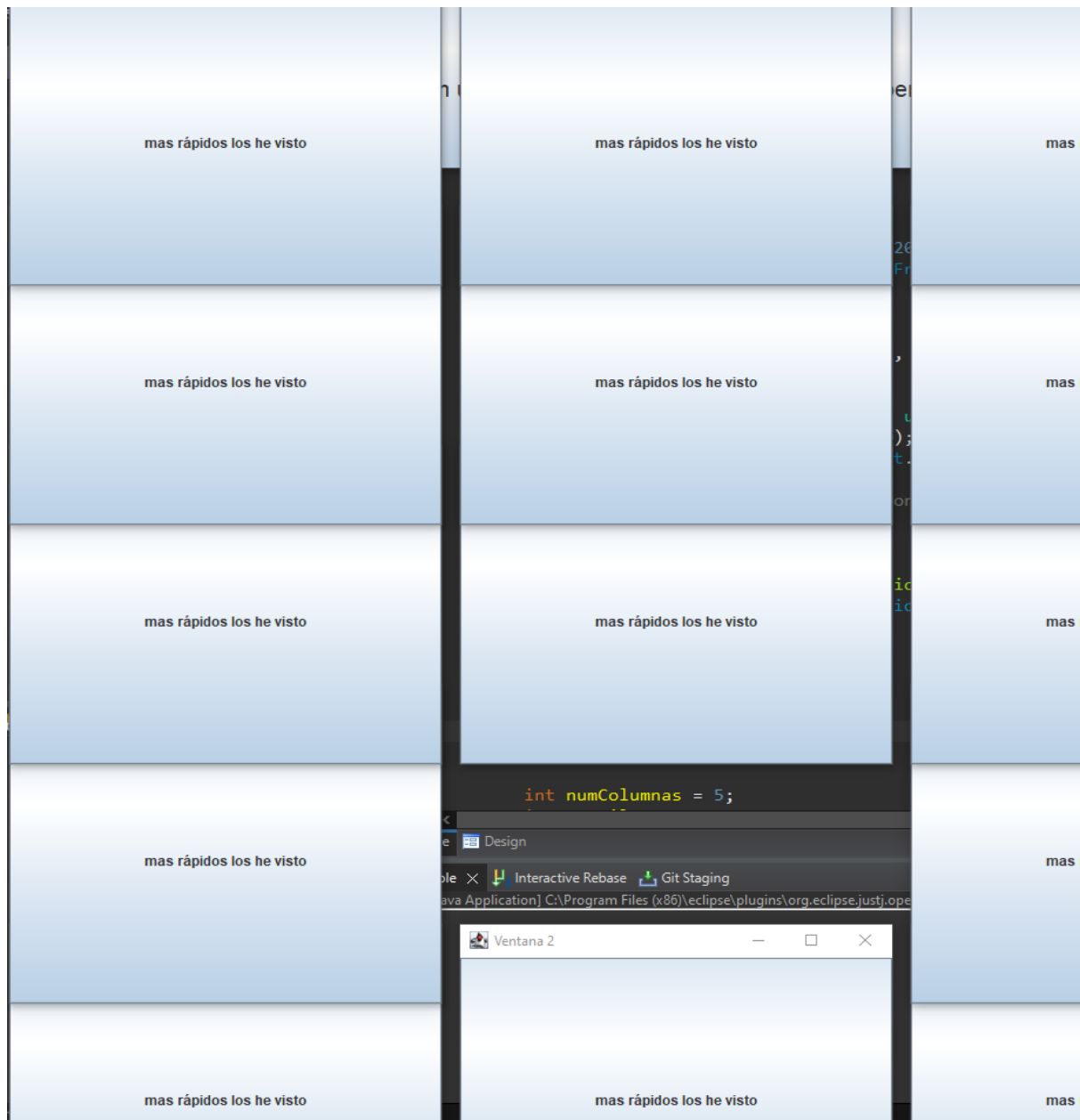
### Funcionamiento

La aplicación consiste en una ventana principal grande con un botón que da una explicación breve del “juego” que se inicia al hacer click en él.



```
private void initialize() {  
    frameUno = new JFrame();  
    frameUno.setBounds(200, -40, 1400, 200);  
    frameUno.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    JPanel panel = new JPanel();  
    frameUno.getContentPane().add(panel, BorderLayout.CENTER);  
    panel.setLayout(null);  
    botoncito = new JButton("Se abrirán unas ventanas que se cierran dando clic, pero solo una esta habilitada. al cerrar esa se  
    habilitará otra. Que comience el juego..");  
    botoncito.setBounds(0, 0, 1380, 180);  
    Font fuente = new Font("Arial", Font.PLAIN, 20); // Puedes ajustar el tamaño (18 en este caso)  
    // Establecer la fuente en el JButton  
    botoncito.setFont(fuente);  
    panel.add(botoncito);  
    botoncito.addActionListener(new ActionListener() {  
        public void actionPerformed(ActionEvent e) {  
            abrirVentanas();  
            frameUno.setEnabled(false);  
        }  
    });  
}
```

Luego se abren múltiples ventanas ocupando toda la pantalla, pero no las puedes cerrar porque todas están deshabilitadas, menos una elegida al azar. Cuando consigues cerrar la que está habilitada, otra al azar se habilita para poder ser cerrada. Durante todo este tedioso proceso va cambiando el mensajito de cada ventana para hacerlo algo mas divertido.



```
private void abrirVentanas() {
    int numColumnas = 5;
    int numFilas = 5;
    int screenWidth = (int) java.awt.Toolkit.getDefaultToolkit().getScreenSize().getWidth();
    int screenHeight = (int) java.awt.Toolkit.getDefaultToolkit().getScreenSize().getHeight();
    int windowHeight = screenHeight / numFilas;
    int windowWidth = screenWidth / numColumnas;
    ventanas.clear();
    for (int fila = numFilas - 1; fila >= 0; fila--) { // Comienza desde la fila inferior
        for (int columna = 0; columna < numColumnas; columna++) {
            JFrame ventana = new JFrame("Ventana " + (ventanaIndex + 1));
            ventana.setBounds(columna * windowWidth, fila * windowHeight - 30, windowWidth, windowHeight + 80);
            ventana.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
            ventanas.add(ventana);
            ventana.setVisible(true);
            JButton boton = new JButton("clic para cerrar");
```

```

boton.setRollerEnabled(false);
ventana.add(boton);
boton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        ventana.dispose();
    }
});
ventana.addWindowListener(new WindowAdapter() {
    @Override
    public void windowClosed(WindowEvent e) {
        frameUno.setVisible(true);
        if(ventanas.size()==1) {
            frameUno.setEnabled(true);
            restringirRaton();
        } else if(ventanas.size()<5 && ventanas.size()>2) {
            cambiarTextoBotonPrincipal("COMO SIGAS CERRANDO TE CASTIGO!!!");
            cambiarColorBotonPrincipal(Color.RED);
        }
        if(ventanas.size()<7 && ventanas.size()>5) {
            cambiarTextoBotones("¿¿estas intentando huir??");
        } else if(ventanas.size()<13 && ventanas.size()>7) {
            cambiarTextoBotones("¿sabe una cosssa??");
        } else if(ventanas.size()<15 && ventanas.size()>13) {
            cambiarTextoBotones("¿tu tiene el fakin bisho?");
        } else if(ventanas.size()<17 && ventanas.size()>15) {
            cambiarTextoBotones("va vení Cristo y va pegá a la Tierra!");
        } else if(ventanas.size()<19 && ventanas.size()>17) {
            cambiarTextoBotones("¿y QUIÉN! es el mesejero del mesejero??");
        } else if(ventanas.size()<22 && ventanas.size()>19) {
            cambiarTextoBotones("¿unas frases de skippy para amenizar?");
        } else if(ventanas.size()<25 && ventanas.size()>22) {
            cambiarTextoBotones("mas rápidos los he visto");
        }
        ventanas.remove(ventana);
        for (int i = 0; i < ventanas.size(); i++) {
            ventanas.get(i).setVisible(true);
        }
        habilitarVentanaAleatoria();
    }
});
ventanalIndex++;
}
}
habilitarVentanaAleatoria();
}
private void habilitarVentanaAleatoria() {
    if (!ventanas.isEmpty()) {
        int randomIndex = new Random().nextInt(ventanas.size());
        for (int i = 0; i < ventanas.size(); i++) {
            ventanas.get(i).setEnabled(i == randomIndex);
        }
    }
}

```

Cuando ya quedan pocas para terminar, la ventana principal va cambiando de color y mandando mensajes de aviso de que te castigará si las cierras todas. Si lo haces, tu ratón quedará bloqueado en una parte de la pantalla durante 30 segundos como castigo.

Métodos de color y cambio de mensaje:



```
private static void cambiarTextoBotonPrincipal(String nuevoTexto) {
    if (botoncito != null) {
        botoncito.setText(nuevoTexto);
    }
}

private static void cambiarColorBotonPrincipal(Color color) {
    if (botoncito != null) {
        botoncito.setBackground(color);
    }
}
```

Métodos de bloqueo del ratón:

```
private static class RatonClicker extends SwingWorker<Void, Void> {
    @Override
    protected Void doInBackground() throws Exception {
        try {
            robot = new Robot();
            timer = new Timer();
            int x = java.awt.MouseInfo.getPointerInfo().getLocation().x;
            int y = java.awt.MouseInfo.getPointerInfo().getLocation().y;
            // Restringe el movimiento del ratón dentro de una región específica
            int xMin = 100; // Límite izquierdo
            int xMax = 300; // Límite derecho
            int yMin = 100; // Límite superior
            int yMax = 200; // Límite inferior
            // Programa una tarea que restablecerá el movimiento del ratón después de 30 segundos
            timer.schedule(new TimerTask() {
                @Override
                public void run() {
                    liberarRaton();
                }
            }, 30000); // 30,000 milisegundos (30 segundos)
            while (true) {
                int currentX = java.awt.MouseInfo.getPointerInfo().getLocation().x;
                int currentY = java.awt.MouseInfo.getPointerInfo().getLocation().y;
                if (currentX < xMin) {
                    robot.mouseMove(xMin, currentY);
                } else if (currentX > xMax) {
                    robot.mouseMove(xMax, currentY);
                }
            }
        }
    }
}
```

```

if (currentY < yMin) {
    robot.mouseMove(currentX, yMin);
} else if (currentY > yMax) {
    robot.mouseMove(currentX, yMax);
}

Thread.sleep(10); // Espera para evitar que el ciclo sea demasiado rápido
} catch (AWTException | InterruptedException e) {
    e.printStackTrace();
}
return null;
}

}

public static void restringirRaton() {
    SwingUtilities.invokeLater(() -> {
        cambiarTextoBotonPrincipal("MIRA QUE TE HE AVISAO!!");
        cambiarColorBotonPrincipal(Color.CYAN);
    });
    RatonClicker ratonClicker = new RatonClicker();
    ratonClicker.execute();
}

public static void liberarRaton() {
    if (robot != null) {
        robot.mouseMove(java.awt.MouseInfo.getPointerInfo().getLocation().x, java.awt.MouseInfo.getPointerInfo().getLocation().y);
        robot = null;
    }
    if (timer != null) {
        timer.cancel();
        timer.purge();
        timer = null;
    }
    cambiarTextoBotonPrincipal("CLIC PARA SALIR DE ESTA MIERDA");
    cambiarColorBotonPrincipal(Color.GREEN);
}

```

Tras pasar los 30 segundos, la ventana principal cambia de color de nuevo y pone “CLICK PARA SALIR DE ESTA MIERDA” pero realmente si clicas se vuelven a desplegar todas las ventanas. debes cerrar con la X, alt+f4 o administrador de tareas.

## Problemas encontrados

El primer problema fue la distribución de las ventanas para que se crearan solas y se repartieran equitativamente por la pantalla con un tamaño uniforme. Hice un bucle con filas y columnas le pregunté a chat GPT si había algo como el grid de css pero en java y lo implementé.

Mi siguiente problema fué que si pasaba el ratón por encima de la ventana habilitada se resaltaba, pero lo solucioné con `boton.setRolloverEnabled(false)`

Necesitaba lo mismo para que no se resaltara tampoco el marco de la ventana, los botones de maximizar y minimizar y eso, pero no era posible. Así que decidí modificar la posición Y de las ventanas durante su creación para que el marco quedara escondido bajo la ventana colindante, solución cutre pero efectiva. Con la ventana principal hice lo mismo.

El último problema que tuve es que quería bloquear el ratón en una posición de la ventana pero no tenía ni idea de como hacerlo ni como ponerle un temporizador, pero mi amigo chat GPT me ayudó tras un buen rato discutiendo con él.