

La aplicación muestra un botón que, al hacer clic, genera una serie de ventanas en posiciones aleatorias en la pantalla. Una de estas ventanas contiene un botón llamado "Llave", que, al hacer clic, cierra todas las ventanas. Además, las ventanas parpadean intermitentemente para agregar un efecto adicional.

Creamos una ventana principal con un botón llamado "¿Quieres jugar a un juego? Encuentra la llave".

Al hacer clic en este botón, se llama al método crearVentanasAleatorias() y se desactiva la ventana principal.

```
frame = new JFrame();
frame.setSize(500, 200);
frame.setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE); //Deshabilito el cierre del programa
frame.setLocationRelativeTo(null);
frame.getContentPane().setLayout(null);
frame.setResizable(false);
frame.getContentPane().setBackground(Color.BLACK);
frame.setVisible(true);

// Botón principal
botonPrincipal = new JButton("¿Quieres jugar a un juego? \n"
    + "Encuentra la llave");
botonPrincipal.setBounds(50, 50, 350, 80);
botonPrincipal.setBackground(Color.white);
botonPrincipal.setForeground(Color.black);
botonPrincipal.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        //Creamos ventanas en posiciones aleatorias
        crearVentanasAleatorias();
        frame.setEnabled(false);
        frame.setVisible(true);
    }
});
frame.getContentPane().add(botonPrincipal);
}
```

Creamos entre 1 y 120 ventanas aleatorias con tamaños diferentes y posiciones diferentes, además de contar con un temporizador para que dichas ventanas vayan cambiando.

```
Random random = new Random();
int numVentanas = random.nextInt(120) + 1; //Crea entre 1 y 120 ventanas
List<JFrame> ventanas = new ArrayList<>();

//Para obtener el tamaño total de la pantalla hacemos lo siguiente
GraphicsEnvironment tamanio = GraphicsEnvironment.getLocalGraphicsEnvironment();
GraphicsDevice[] pantalla = tamanio.getScreens();

for (int i = 0; i < numVentanas; i++) {
    int screenIndex = random.nextInt(pantalla.length);
    GraphicsDevice screen = pantalla[screenIndex];
    Rectangle bounds = screen.getDefaultConfiguration().getBounds();

    JFrame ventana = new JFrame();
    ventana.setBounds(bounds.x + random.nextInt(bounds.width - 50), bounds.y + random.nextInt(bounds.height - 50));
    ventana.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    ventana.setVisible(true);

    Timer timer = new Timer();
    timer.scheduleAtFixedRate(new TimerTask() {
        public void run() {
            int newWidth = random.nextInt(150) + 100;
            int newHeight = random.nextInt(100) + 50;
            ventana.setSize(newWidth, newHeight);
        }
    }, 1000, 1000);
}
```

Creamos un botón “llave” que al pulsar en él elimine todas las ventanas, pero claro, al ser un bucle infinito, van a seguir apareciendo dichas ventanas

```
//Creamos el boton llave con características diferentes
if (i == 0) {
    int llaveX = bounds.x + random.nextInt(bounds.width - 200);
    int llaveY = bounds.y + random.nextInt(bounds.height - 50);

    JButton botonLlave = new JButton("Llave");
    botonLlave.setBounds(llaveX, llaveY, 5, 5);
    botonLlave.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            //Cerramos todas las ventanas
            for (JFrame v : ventanas) {
                v.dispose();
            }
        }
    });
    ventana.getContentPane().add(botonLlave);
}

//Hacemos que las ventanas parpadeen
Timer blinkTimer = new Timer();
blinkTimer.scheduleAtFixedRate(new TimerTask() {
    private boolean isVisible = true;
    public void run() {
        ventana.setVisible(isVisible);
        isVisible = !isVisible;
    }
}, 0, 500);

ventanas.add(ventana);
```

He tenido problemas a la hora de generar las repetidas ventanas y al seleccionar una para que sea la ventana “llave”.

Además he tenido problemas con el tiempo de parpadeo el cual he tenido que buscar en Internet, además de obtener las dimensiones de la pantalla donde se abre el programa.

Enlace del programa en mi repositorio:

[https://github.com/Yeraaay/DAM\\_2\\_YerayMoraledaCabrera/tree/00387c31e3a6f6b31d32b1ebfb11003fd597aa47/App\\_Troll](https://github.com/Yeraaay/DAM_2_YerayMoraledaCabrera/tree/00387c31e3a6f6b31d32b1ebfb11003fd597aa47/App_Troll)

La rama donde he realizado dicho proyecto ha sido en la rama main.