

# PROYECTO FINAL: DESARMA LA BOMBA

Alexis Moran Bonilla

Estructura de datos

## ESTRUCTURA CLAVE

```
// Lógica del juego
private Stack<String> secuencia;      // pila con códigos (tope = próximo código)
private Queue<String> errores;        // cola con entradas incorrectas
private final int MAX_ERRORES = 3;
```

```
private void reiniciarJuego() {
    // Generar 4 códigos aleatorios 00..99 y poner en pila
    secuencia = new Stack<>();
    Random rnd = new Random();
    for (int i = 0; i < 4; i++) {
        int num = rnd.nextInt(bound: 100); // 0..99
        secuencia.push(String.format(format: "%02d", num));
    }
}
```

- Se basa en dos estructuras: **Pilas** para los códigos de desactivación, y **Colas** para gestionar los errores.
- La pila simula el orden estricto de los códigos que deben ser introducidos

# DEMOSTRACIÓN

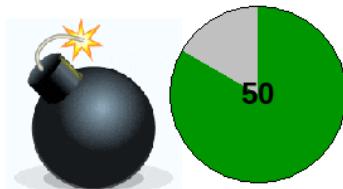
El Applet funciona en Java y tiene una interfaz visual (AWT/Applet).

El jugador tiene **60 segundos** y **3 errores** para desactivar los 4 códigos.

Usamos un hilo para el temporizador y cambiamos la imagen a bomb.gif si el juego termina mal.

## Juego Anti-Bomba (Desarme)

Código (00-99):



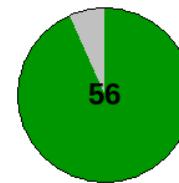
Códigos restantes (pila): 4  
Errores (cola): 3 / 3

## BOMBA EXPLOTÓ (errores)

Demasiados errores. Bomba explotó.

## Juego Anti-Bomba (Desarme)

Código (00-99):



Códigos restantes (pila): 4  
Errores (cola): 0 / 3

CÓDIGO REQUERIDO: 12 | Códigos restantes: 4

# DISEÑO TÉCNICO

El uso de `Stack<String>` (*Pila*) asegura que el código correcto sea siempre `secuencia.peek()`.

El uso de `Queue<String>` (*Cola*) nos permite registrar los errores en el orden en que ocurrieron, cumpliendo con la estructura.

```
// Lógica del juego
private Stack<String> secuencia;          // pila con códigos (tope = próximo código)
private Queue<String> errores;            // cola con entradas incorrectas
private final int MAX_ERRORES = 3;
```

```
// Comparar con el tope de la pila
String esperado = secuencia.peek();
if (code.equals(esperado)) {
    secuencia.pop();
```

```
// Descomenta la línea de abajo para mostrar el código requerido con el mensaje de error.
String nextCode = secuencia.isEmpty() ? "" : " | CÓDIGO REQUERIDO: " + secuencia.peek();
infoLabel.setText("INCORRECTO. Errores: " + errores.size() + " / " + MAX_ERRORES + nextCode);
```

# CONCLUSIONES Y FUTURO

Demuestra cómo las Pilas y Colas se aplican a problemas reales de control de flujo.

Aprendizaje fue la importancia de elegir la estructura de datos adecuada para modelar el problema.

Como mejora futura, se podría agregar autenticación y un sistema de puntajes



GRACIAS POR SU ATENCIÓN

