

# Simulador de Batalla de Robots

## Explicación de la Solución

### Introducción

El objetivo de este laboratorio fue desarrollar un programa en Java que simula una batalla entre robots. Para cumplirlo, utilicé el concepto de **Programación Orientada a Objetos (POO)** creando dos clases principales: Robot y JuegoBatalla.

### Estructura del Programa

- Clase Robot
  - Representa cada robot participante.
  - Atributos:
    - nombre: almacena el nombre del robot.
    - puntosVida: vida del robot (valor entre 50 y 100).
    - ataque: cantidad de daño que puede causar (valor entre 10 y 20).
  - Métodos:
    - atacar(Robot otroRobot): permite que un robot ataque a otro y reduzca su vida.
    - estaVivo(): devuelve true si el robot sigue con vida.
- Clase JuegoBatalla
  - Administra el proceso de la batalla.
  - Atributos:
    - robots: array que almacena los objetos de tipo Robot.
  - Métodos:
    - iniciarBatalla(): cada robot ataca a otro aleatorio en cada turno; elimina robots que pierden toda su vida.
    - mostrarGanador(Robot ganador): muestra al robot que ganó la batalla.
    - main(String[] args): método principal que solicita al usuario la cantidad de robots, sus nombres, y que luego inicia la batalla.

## Lógica de la Simulación

1. El programa solicita al usuario cuántos robots desea crear (máximo 10).
2. Por cada robot:
  - Se pide un nombre ingresado por el usuario.
  - Se generan puntos de vida aleatorios entre 50 y 100.
  - Se generan valores de ataque aleatorios entre 10 y 20.
3. Una vez creados los robots:
  - Inicia la batalla, donde cada robot ataca a otro robot aleatorio en cada turno.
  - Si un robot recibe daño suficiente para reducir su vida a 0 o menos, se elimina del combate.
4. El proceso continúa hasta que quede solo un robot vivo, quien será declarado el ganador.

## Detalles Adicionales

- Se usa Scanner para capturar las entradas del usuario.
- Se usa Random para asignar números aleatorios (vida, ataque y selección de objetivos).
- Los robots muertos son eliminados colocando su posición en el array como null.
- Se muestra en consola cada ataque, destrucción de robots y el anuncio final del ganador.

## Conclusión

La solución revisa conceptos fundamentales de la programación orientada a objetos en Java como:

- Creación de clases y objetos.
- Uso de arrays para almacenar objetos.
- Lógica de control de flujo mediante ciclos y condicionales.