

## Solución Implementada – Laboratorio 2

### Parte 1: Uso de Git y GitHub

Para la primera parte del laboratorio, configuré mi entorno de trabajo utilizando Git y GitHub como herramientas principales.

### Parte 2: Desarrollo del Simulador de Batalla de Robots

La segunda parte consistió en programar un simulador de batalla entre robots, utilizando Java y aplicando los conceptos de Programación Orientada a Objetos y arrays.

#### Diseño del Programa

El programa consta de dos clases principales:

- Robot.java: define la estructura de un robot, con atributos como nombre, vida, ataque y defensa. Incluye métodos para atacar a otros robots (**atacarA**) y para verificar si el robot sigue vivo (**estaVivo**).
- SimuladorBatalla.java: contiene la lógica principal del juego, como la creación de robots, la ejecución de turnos de combate y la determinación del ganador.

#### Flujo del programa

1. Presentación: Al iniciar, el programa muestra un título de bienvenida en consola.
2. Ingreso de datos: El usuario debe ingresar:
  - La cantidad de robots participantes (entre 2 y 10).
  - El nombre, puntos de vida (50–100), ataque (10–20) y defensa (0–10) para cada robot.
3. Simulación de la batalla:
  - Mientras haya más de un robot vivo, cada robot realiza un ataque a otro robot seleccionado aleatoriamente.

- Se calcula el daño teniendo en cuenta la defensa del objetivo.
- Si el robot atacado reduce su vida a cero, es considerado eliminado.

#### 4. Finalización:

- Cuando solo queda un robot vivo, se muestra en pantalla como el ganador, indicando su nombre y puntos de vida restantes.

#### Particularidades de la implementación

- Validaciones: Se realizaron validaciones para asegurar que los valores ingresados por el usuario estén en los rangos correctos.
- Control de flujo: Después de cada ataque, se solicita al usuario presionar Enter para continuar, permitiendo observar el desarrollo de la batalla turno por turno.
- Array de Robots: Se utiliza un arreglo de objetos `Robot[] robots` para almacenar y manipular a los participantes de la batalla.

#### Resultado Final

La simulación ejecuta de manera exitosa los turnos de ataque entre robots, gestionando las eliminaciones de manera automática y mostrando el estado actualizado de la batalla en cada paso. La implementación cumple con todos los requisitos del laboratorio, incluyendo las validaciones de ingreso, el control de pausas, y el manejo adecuado de arrays y objetos en Java.

Amy Chen Wu. C32203