

Instalación y uso de entornos de desarrollo

2.2 Eclipse y Java



Autor: Lisa ERIKSEN

Fecha: 2025 / 2026

1. Objetivos de Aprendizaje

Practicar los pasos esenciales aprendidos en esta parte del curso:

- Crear un proyecto Java en Eclipse.
- Escribir, ejecutar y depurar un programa Java.
- Familiarizarse con el entorno de Eclipse (perspectiva Java y depuración).

2. Usa Eclipse para programar en Java

Imagina que trabajas en un estudio que desarrolla simuladores. Tu tarea será crear un pequeño programa Java que **simule una “Batalla de Robots”**, ejecutarlo en Eclipse y usar la depuración para analizar el comportamiento del programa.

2.1 Crear el proyecto

1. Instalar Eclipse si no está instalado en tu equipo: [download](#)
2. Abre Eclipse.
3. Ve a **File > New > Java Project**.
4. Nombra el proyecto **BatallaRobots**
5. Haz clic en **Finish**.

2.2 Crear la Clase principal

1. En la carpeta **src**, haz clic derecho → **New > Class**.
2. Escribe el nombre **BatallaRobots** para la clase y el nombre **batallaRobots** al paquete
3. Marca la opción: `public static void main(String[] args)`.
4. Haz clic en **Finish**.

2.3 Escribir el programa

👉 Copia y pega este código

```
public class BatallaRobots {
    public static void main(String[] args) {
        int energiaRobotA = 100;
        int energiaRobotB = 100;

        System.out.println("🤖 Bienvenido a la Batalla de Robots 🤖");
        System.out.println("Robot A vs Robot B\n");

        while (energiaRobotA > 0 && energiaRobotB > 0) {
            int golpeA = (int)(Math.random() * 20) + 1;
            int golpeB = (int)(Math.random() * 20) + 1;

            energiaRobotA -= golpeB;
            energiaRobotB -= golpeA;

            System.out.println("Robot A golpea con " + golpeA + " puntos.");
            System.out.println("Robot B golpea con " + golpeB + " puntos.");
            System.out.println("Energía A: " + energiaRobotA + " | Energía B: " + energiaRobotB
+ "\n");
        }

        if (energiaRobotA > energiaRobotB) {
            System.out.println("🏆 ¡Robot A gana la batalla!");
        } else if (energiaRobotB > energiaRobotA) {
            System.out.println("🏆 ¡Robot B gana la batalla!");
        } else {
            System.out.println("👉 ¡Empate!");
        }
    }
}
```

2.4 Ejecutar el programa

1. Haz clic derecho sobre BatallaRobots.java.
2. Selecciona **Run As > Java Application**.
3. Observa los resultados en la **Consola**

💬 Si todo está correcto, deberías ver cómo los robots pierden energía hasta que uno gana.

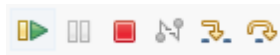
2.5 Depurar el programa

1. Haz clic en el margen izquierdo junto a la línea:

```
int golpeA = (int)(Math.random() * 20) + 1;
```

● Aparecerá un punto azul (punto de ruptura).

2. Haz clic derecho sobre BatallaRobots.java → **Debug As > Java Application**.
3. Eclipse cambiará a la vista de **Depuración**.
4. Usa los botones de la barra arriba:
 - **Resume** → continúa la ejecución.
 - **Step Over** → ejecuta una línea a la vez.
 - Observa el valor de las variables en la vista **Variables**



5. Cuando el programa se detenga en el punto de ruptura:
 - ¿Qué valor tiene la variable golpeA?
 - Intenta cambiarlo manualmente desde la vista **Variables**. ¿Se puede modificar valores durante la ejecución ?

6. Usa **Step Over** varias veces:
 - ¿Qué cambia en cada paso?
 - Explica qué línea está afectando a los puntos de vida del robot.
7. Coloca un nuevo punto rojo en una línea más adelante (por ejemplo, donde se reduce la vida del segundo robot).
 - ¿Por qué el programa se detiene en ese nuevo punto?
 - ¿En qué momento exacto la vida llega a 0?
8. Finaliza con **Terminate** cuando desees parar.

2.6 ¿Entendiste todo?

1. ¿Dónde se guarda el código fuente en un proyecto de Eclipse?
 - a) bin
 - b) src
 - c) JRE System Library
2. ¿Qué comando inicia la ejecución en Eclipse?
 - a) Debug As
 - b) Run As
 - c) Build Project
3. ¿Qué vista te muestra los valores de las variables al depurar?
 - a) Console
 - b) Variables
 - c) Breakpoints

3. ¡Practicamos! 🎮

Objetivo: ¡Mejora el programa de batalla de los robots!

Reto 1 — Nombres de robots desde el usuario 🤖

Pide al usuario que escriba los nombres de Robot A y Robot B.

Pista: Usa Scanner

Reto 2 — Mostrar la ronda 🕒

Antes de cada golpe, muestra:

```
===== Ronda 1 =====
===== Ronda 2 =====
```

Reto 3 — Limitar la fuerza máxima 🥊

Permite al usuario elegir el daño máximo posible (por ejemplo, entre 10 y 50).

Esto afecta a Math.random()

Reto 4 — Barra de energía 📊

En lugar de solo números, muestra:

```
Energía A: ##### (75)
Energía B: ###   (35)
```

Pista: usa un bucle for

Reto 5 — Modo Turbo ⚡

Si un robot obtiene un golpe de 20, gana otro golpe inmediato.

Pista: Usa un if condicional

Objetivo: Eres un constructor en **Minecraft** y necesitas saber cuántos bloques ocupa tu parcela rectangular y cuánta valla necesitarás para cerrarla. Vas a crear el proyecto en **Eclipse**, programar la clase Rectangulo, ejecutar y depurar el programa.



Pasos:

1. Crea un nuevo proyecto llamado Minecraft.
2. En la carpeta src, crea una clase llamada Rectangulo.
3. Escribe un programa que calcule el área y el perímetro de un rectángulo dado su ancho y alto.
4. Usa puntos de ruptura para depurar el programa y observa cómo cambian las variables durante la ejecución.