

```
import java.util.Random;

public class Dado {
    private int caras; // Número de caras del dado

    public Dado(int caras) {
        this.caras = caras;
    }

    public int tirar() {
        Random random = new Random();
        return random.nextInt(caras) + 1; // Devuelve un número aleatorio entre 1 y "caras"
    }
}
```

```

public class Jugador {
    private String nombre;
    private int salud;
    private int poderDeAtaque;

    public Jugador(String nombre) {
        this.nombre = nombre;
        this.salud = 100; // Salud inicial del jugador
        this.poderDeAtaque = 10; // Poder de ataque inicial
    }

    public String getNombre() {
        return nombre;
    }

    public int getSalud() {
        return salud;
    }

    public void recibirDanio(int danio) {
        salud -= danio;
        if (salud < 0) salud = 0; // La salud no puede ser negativa
    }

    public void sanar(int puntos) {
        salud += puntos;
        if (salud > 100) salud = 100; // La salud no puede exceder 100
    }

    public int realizarTiradaAtaque(Dado dado) {
        return dado.tirar() + poderDeAtaque; // Suma el valor de la tirada con el poder de
ataque
    }

    @Override
    public String toString() {
        return nombre + " (Salud: " + salud + ")";
    }
}

```

```

import java.util.Scanner;

public class JuegoRol {
    private Jugador jugador1;
    private Jugador jugador2;
    private Dado dadoAtaque;

    public JuegoRol() {
        dadoAtaque = new Dado(6); // Dado de 6 caras para los ataques
        jugador1 = new Jugador("Jugador 1");
        jugador2 = new Jugador("Jugador 2");
    }

    public void mostrarEstado() {
        System.out.println(jugador1);
        System.out.println(jugador2);
    }

    public void realizarAccion(Jugador atacante, Jugador defensor) {
        Scanner scanner = new Scanner(System.in);
        System.out.println(atacante.getNombre() + ", ¿Qué acción deseas realizar?");
        System.out.println("1. Atacar");
        System.out.println("2. Sanar");
        System.out.print("Elige una opción (1/2): ");
        int opcion = scanner.nextInt();

        if (opcion == 1) {
            // Realizar tirada de ataque
            int dano = atacante.realizarTiradaAtaque(dadoAtaque);
            System.out.println(atacante.getNombre() + " ha lanzado el dado y ha obtenido " +
dano + " de daño.");
            defensor.recibirDanio(dano);
            System.out.println(defensor.getNombre() + " ahora tiene " + defensor.getSalud() + "
de salud.");
        } else if (opcion == 2) {
            // Sanar
            int curacion = dadoAtaque.tirar();
            System.out.println(atacante.getNombre() + " ha lanzado el dado y ha recuperado " +
curacion + " puntos de salud.");
            atacante.sanar(curacion);
            System.out.println(atacante.getNombre() + " ahora tiene " + atacante.getSalud() + "
de salud.");
        } else {
            System.out.println("Opción inválida. Intenta de nuevo.");
        }
    }

    public void jugar() {

```

```

Scanner scanner = new Scanner(System.in);
while (jugador1.getSalud() > 0 && jugador2.getSalud() > 0) {
    mostrarEstado();
    realizarAccion(jugador1, jugador2);
    if (jugador2.getSalud() > 0) {
        mostrarEstado();
        realizarAccion(jugador2, jugador1);
    }
}

if (jugador1.getSalud() > 0) {
    System.out.println(jugador1.getNombre() + " ha ganado el combate.");
} else if (jugador2.getSalud() > 0) {
    System.out.println(jugador2.getNombre() + " ha ganado el combate.");
} else {
    System.out.println("El combate ha terminado en empate.");
}

public static void main(String[] args) {
    JuegoRol juego = new JuegoRol();
    juego.jugar();
}
}

```