Clase práctica 12/9

Imagina que estás creando un juego de robots de batalla. Debes implementar una clase llamada RobotDeBatalla, que representará a un robot que participa en batallas. Cada robot tiene las siguientes características y comportamientos:

1. **Atributos**:
   - nombre (str): El nombre del robot.
   - energia (int): La energía actual del robot (debe estar entre 0 y 100).
   - ataque (int): La fuerza de ataque del robot (debe estar entre 0 y 50).
   - defensa (int): La capacidad de defensa del robot (debe estar entre 0 y 50).
2. **Métodos**:
   - __init__(self, nombre: str): Inicializa el robot con un nombre y los atributos de energía en 100, ataque en un valor aleatorio entre 20 y 50, y defensa en un valor aleatorio entre 10 y 50.
   - atacar(self, otro_robot): el ataque reduce la defensa del oponente en primer lugar, si la defensa quedó en 0 entonces le reduce la energía del oponente con el valor de ataque que le quede.
   - recargar_energia(self): Incrementa la energía del robot hasta el máximo (100).
   - esta_vivo(self): Retorna True si el robot tiene más de 0 de energía, y False en caso contrario.
3. **Comportamiento del Robot**:
   - Si un robot tiene su energía en 0, ya no puede atacar ni recargar energía.
   - Cuando ataca a otro robot le reduce la energía 5 puntos, y le disminuye vida al otro robot según el valor propio de ataque y considerando la defensa del otro robot.
   - La defensa del robot se ve reducida en el valor de cada ataque que recibe, si recibe un ataque con valor igual o mayor a su defensa, ésta queda anulada, sino se reduce de acuerdo al valor del ataque.
   - Cuando un robot recibe un ataque, puede recargar su energía si su vida es menor que su energía.
   - Para respetar el encapsulamiento debemos incorporar métodos en la clase que permitan modificar los atributos de instancia cuando se requiera, y otros métodos que permitan consultar los valores de los atributos de instancia.

Para probar la clase, implementar una clase tester que simule una pelea de robots. Cada vez que el robot ataca, debe mostrarse un mensaje indicando cuánta energía le queda al oponente. Puede usarse un comportamiento aleatorio para determinar cuando un robot ataca a otro, y cuando un robot puede recargar su energía.