**Desafío 3**



## Explicación línea por línea (desde la línea 1)

1. import java.util.concurrent.ThreadLocalRandom;  
   Importamos ThreadLocalRandom, generador de números aleatorios recomendado en Java moderno para simplicidad y rendimiento (sin necesidad de instanciar Random).
2. public class Desafio3 {  
   Declaramos la clase pública que contiene el main.
3. public static void main(String[] args) {  
   Punto de entrada de la aplicación Java.
4. final int MIN = 0;  
   Mínimo del rango (inclusive).
5. final int MAX = 10;  
   Máximo del rango (inclusive). Lo haremos inclusivo usando MAX + 1 con nextInt.
6. final int LIMITE = 25;  
   Límite superior para la suma de impares; cuando lo superemos, detenemos el procesamiento.
7. int sumaImpares = 0;  
   Acumulador de la suma de los valores impares generados.
8. while (true) {  
   Bucle indefinido porque no sabemos cuántos números necesitaremos hasta superar 25.
9. int n = ThreadLocalRandom.current().nextInt(MIN, MAX + 1);  
   Genera un entero aleatorio en [MIN, MAX]. nextInt(a, b) es [a, b), por eso usamos MAX + 1 para incluir el 10.
10. boolean esPar = (n % 2 == 0);  
    Determinamos la paridad usando el módulo 2.
11. System.out.printf("Generado: %d -> %s%n", n, esPar ? "par" : "impar");  
    Mostramos el número y un mensaje: “par” o “impar”.
12. if (!esPar) {  
    Solo si es impar, lo sumamos al acumulador.
13. sumaImpares += n;  
    Sumamos el valor impar a la suma acumulada.
14. if (sumaImpares > LIMITE) {  
    Verificamos si la suma **supera** 25 (estrictamente mayor que 25).
15. break;  
    Si la supera, salimos del bucle inmediatamente. El último impar ya quedó sumado.

16–17. }  
Cierre de los bloques if.

1. }  
   Cierre del bucle while.
2. System.out.println("Suma de impares = " + sumaImpares);  
   Mostramos la suma final de los impares (ya > 25 en el momento de salir).

20–21. }  
Cierre del main y de la clase.

Enlace a g