

## **UT6 – Reflexión final de unidad**

El universo avanza, y con él las unidades temáticas. “Los buckets se llenan, las colisiones se resuelven, pero el acceso en  $O(1)$  permanece eterno” (San Knuth, versículo no canonizado).

En esta unidad temática me sentí tocado por la mano invisible de la eficiencia. El maravilloso mundo del hashing me recibió con brazos abiertos y estructuras prepotentes. Si algo me quedó grabado en la médula espinal del programador, es que un buen `HashMap` no se discute, se venera.

La función `hash...` ¡Qué entidad mística! Un simple número, pero con el poder de decidir el destino de claves y valores en el cosmos binario (“eguro, ahí va”). Me resultó particularmente curioso (y estupefacto) cómo su implementación varía según el tipo de objeto: no es lo mismo hacer `hash` de un `String` que de un `int`, o de un objeto custom que, por dignidad, merece su propio `hashCode()`. Es como si Java supiera que cada tipo tiene su propio camino hacia el paraíso de las posiciones de memoria.

Disfruté como un zángano (o un niño, si se quiere) con un juguete nuevo explorando métodos para resolver colisiones. ¡Lineal! ¡Cuadrático! ¡Encadenado! ¡Árbol binario si te pintan los extremos! Cada estrategia me ofrecía nuevas formas de domar el caos y alcanzar la eficiencia deseada.

Pude meterle mano a estructuras como `HashMap` y `TreeMap` (aunque este último con menos solvencia) y otros campeones del almacenamiento inteligente, y comprobé en carne propia cómo estas joyas permiten búsquedas y escrituras en tiempos que harían llorar de alegría al mismísimo Jesucristo.

Comparar su rendimiento, analizar su consumo de memoria, entender cuándo conviene cada uno... fue como encontrar el equilibrio entre poder y responsabilidad. No todo es velocidad; también es saber cuándo y cómo usarla. Y ahí, amigos, está el arte.

Me voy de esta unidad no solo con teoría y código, sino con la certeza de que hay belleza en la eficiencia y poesía en las estructuras de datos. “Bienaventurados los que entienden `hashCode()` y `equals()`, porque de ellos será el Reino del Heap” (Evangelio según Java, capítulo JVM).

