

**Tabla comparativa de operaciones:**

Tamaño (n)	Posición objetivo	Función A ( $O(n)$ )	Función B ( $O(n^2)$ )
100	50	51	10,000
10,000	5,000	5,001	100,000,000
1,000,000	999,999	1,000,000	1,000,000,000,000

**Explicación:**

Dos códigos pueden “funcionar igual” porque producen el mismo resultado lógico, pero su rendimiento depende de cuántas operaciones realizan internamente. En entradas pequeñas, las diferencias entre  $O(n)$  y  $O(n^2)$  son difíciles de percibir, por lo que el problema no se detecta durante desarrollo o pruebas manuales. Sin embargo, al crecer el tamaño de los datos, el número de operaciones aumenta de forma radical. Esto provoca que un código aparentemente correcto se vuelva lento, consuma demasiados recursos o incluso sea inutilizable en producción.

**Propuesta para detectar estos problemas en código real:**

- Análisis de complejidad obligatorio en revisiones de código
- Pruebas de rendimiento con datos grandes, no solo casos pequeños
- Uso de profiling para contar iteraciones y tiempo real
- Alertas automáticas para ciclos anidados dependientes de n
- Comparar siempre contra soluciones conocidas óptimas