

Checkpoint Metacognitivo – Fase REFLEXIONA

¿Qué descubrí sobre mi propio proceso de pensamiento que no sabía antes?

Descubrí que mi primera reacción al analizar un código no es pensar en la sintaxis ni en si funciona correctamente, sino en **cuántas veces se repite una operación**. Antes no era consciente de que mi atención se iba directamente al número de iteraciones y al crecimiento del trabajo conforme aumenta el tamaño de la entrada. También me di cuenta de que mi comprensión mejora mucho cuando traduzco la teoría a **números concretos**, como al comparar el crecimiento de $O(n)$ y $O(n^2)$ con valores grandes.

¿Cómo describiría mi “estilo” de análisis?

Mi estilo de análisis es principalmente **práctico e intuitivo**, con apoyo visual y numérico. No parto de fórmulas matemáticas abstractas, sino de observar patrones en el código y pensar qué pasa cuando los datos crecen. Me ayuda ver tablas, comparaciones y escenarios reales más que definiciones formales. A partir de ahí, conecto esos patrones con la teoría de complejidad.

¿Qué conexión inesperada hice entre este tema y algo que ya sabía?

Hice la conexión entre el análisis de complejidad y problemas reales de rendimiento que ya había visto, como aplicaciones que se vuelven lentas aunque “funcionen bien”. Entendí que muchos de esos problemas no se deben a errores de programación, sino a **decisiones algorítmicas mal escaladas**, algo similar a cómo en otras áreas una solución puede servir a pequeña escala pero fallar cuando el contexto cambia.

Si empezara la semana de nuevo, ¿qué haría diferente en mi proceso de aprendizaje?

Desde el inicio analizaría la complejidad **antes** de asumir que una solución es buena solo porque funciona. Me obligaría a pensar siempre en el peor caso y a probar mentalmente el comportamiento del algoritmo con datos grandes. También documentaría mi razonamiento desde el principio, en lugar de hacerlo solo cuando detecto un error, para identificar más rápido mis supuestos incorrectos.