Estudio de Códigos - Nelson Rodríguez

# Código 1: Ejecuciones de Fibonacci

Este código muestra tres métodos para calcular la serie de Fibonacci:

**- Recursiva:** fácil de comprender, pero ineficaz; su crecimiento es exponencial.

**-** **La memoización:** emplea caché para almacenar los resultados y optimizar el rendimiento.

**- Iterativa:** es la más veloz y la que menos memoria utiliza.

Se comprueba que las tres generan los mismos resultados con valores bajos (n<10).

# Código 2: Evaluación de la memoria y los tiempos

Cada implementación de Fibonacci es evaluada por este código en términos del tiempo que toma ejecutarse y la cantidad máxima de memoria utilizada. Lo que los hallazgos indican es que:

- La recursiva, a medida que n se incrementa, utiliza una gran cantidad de tiempo y memoria (por ejemplo: para n=37 puede tardar varios segundos).

- La versión con memoización es veloz y requiere escasa memoria.

- La versión iterativa presenta estabilidad y rapidez y con una memoria casi estable.

Esto evidencia que el desempeño se ve afectado por la selección del algoritmo.

# Código 3: Gráfica comparativa utilizando Plotly

Este código, mediante Plotly, muestra los resultados obtenidos en una gráfica. En el gráfico se puede ver que la versión recursiva consume recursos a un ritmo mucho más acelerado que las demás, que permanecen casi estables. La visualización es útil para comprender de manera más efectiva la distinción entre las complejidades O(n) y O(log n).

Resumen general

Los tres códigos muestran de manera práctica cómo distintas implementaciones de un mismo algoritmo pueden diferir en términos de memoria y tiempo. Esto es fundamental para seleccionar la solución más apropiada, de acuerdo con el problema y los recursos existentes.

# Codigo log: Gráfica comparativa

El segundo gráfico utiliza una escala logarítmica en los ejes. Esto hace que la diferencia entre los algoritmos sea más clara: la búsqueda lineal se expande en proporción al tamaño de la lista, pero la búsqueda binaria lo hace con una velocidad mucho más lenta, siguiendo una curva logarítmica. Esta escala permite ver más fácilmente que la búsqueda binaria conserva su eficacia, aun en listas grandes.