

# Estudio Experimental: Complejidad de Peor Caso vs. Amortizada en Vectores Dinámicos

Martin Valdenegro, Felipe Colli, Benjamin Fernandez  
*Instituto Nacional Jose Miguel Carrera*

1° de Abril, 2025

## 1 Observación

En el análisis teórico de algoritmos, la complejidad del peor caso ( $O$ ,  $\Theta$ ) proporciona una cota superior estricta sobre el tiempo de ejecución de una operación individual. Sin embargo, para ciertas estructuras de datos, como los vectores dinámicos (arrays que crecen bajo demanda), las operaciones costosas (ej. redimensionamiento al añadir un elemento) ocurren con poca frecuencia. El análisis amortizado sugiere que el costo promedio por operación, considerado sobre una secuencia larga, puede ser significativamente menor que el costo del peor caso de una operación aislada. Se observa que, en la práctica, estructuras como `std::vector` en C++ o `ArrayList` en Java son muy eficientes a pesar de tener operaciones con un peor caso lineal ( $O(n)$ ).

## 2 Pregunta de Investigación

¿Hasta qué punto el tiempo de ejecución real medido para secuencias de operaciones (específicamente inserciones) en un vector dinámico se alinea mejor con las predicciones del análisis amortizado ( $O(1)$  por operación en promedio) en comparación con las predicciones basadas únicamente en el análisis del peor caso ( $O(n)$  para la operación más costosa)? ¿Podemos cuantificar experimentalmente la frecuencia y el impacto de las operaciones de peor caso en el rendimiento general?

## 3 Hipótesis

Se hipotetiza que el tiempo *total* para realizar  $N$  inserciones sucesivas en un vector dinámico crecerá de forma aproximadamente lineal con  $N$ , lo que implica que el tiempo *promedio medido* por inserción será cercano a una constante ( $O(1)$ ), validando así la predicción del análisis amortizado para el rendimiento agregado. Aunque se observarán picos de tiempo correspondientes a las operaciones de redimensionamiento (validando el análisis de peor caso para *esas* operaciones específicas), su impacto en el tiempo promedio disminuirá a medida que  $N$  aumente, demostrando que el análisis amortizado es un predictor más útil del rendimiento práctico general de la estructura de datos en este escenario.

## 4 Método de Experimentación

Para probar la hipótesis, se realizará el siguiente experimento:

1. **Implementación/Selección:** Se utilizará una implementación estándar de vector dinámico (ej. `std::vector` de C++ o `ArrayList` de Java) o se implementará una versión simple siguiendo la estrategia de duplicar la capacidad cuando esté lleno.
2. **Operación Bajo Prueba:** La operación principal a medir será la inserción al final ("pushback" o `add`).

### 3. Procedimiento:

- Para diferentes valores de  $N$  (número total de inserciones, ej.  $10^3, 10^4, 10^5, 10^6, 10^7$ ):
    - (a) Iniciar con un vector dinámico vacío.
    - (b) Realizar  $N$  inserciones consecutivas al final del vector.
    - (c) Medir y registrar el tiempo de ejecución de **cada** inserción individual. Utilizar un cronómetro de alta resolución (ej. `std::chrono` en C++, `System.nanoTime()` en Java).
    - (d) Registrar el tiempo **total** para las  $N$  inserciones.
  - Repetir cada experimento (para cada  $N$ ) varias veces (ej. 5-10 veces) para mitigar fluctuaciones y calcular promedios y desviaciones estándar.
4. **Entorno Controlado:** El experimento se ejecutará en la misma máquina, con carga mínima del sistema, utilizando el mismo lenguaje/compilador y opciones de optimización (si aplica) para todas las pruebas.

## 5 Análisis de Resultados

Los datos recolectados se analizarán de la siguiente manera:

1. **Visualización de Tiempos Individuales:** Se generará un gráfico del tiempo de ejecución por cada inserción individual (eje Y) vs. el número de inserción (índice  $i$  de 1 a  $N$ , eje X) para un valor grande de  $N$ . Esto permitirá visualizar los picos de tiempo correspondientes a las redimensiones.
2. **Análisis de Picos:** Se identificará el tiempo máximo registrado para una sola inserción y se comparará con el tamaño del vector ( $k$ ) en ese momento para verificar si se aproxima a una relación lineal (validación cualitativa del peor caso  $O(k)$ ).
3. **Cálculo del Tiempo Promedio Amortizado Experimental:** Para cada  $N$ , se calculará el tiempo promedio por inserción dividiendo el tiempo total medido por  $N$ .
4. **Visualización del Tiempo Promedio:** Se generará un gráfico del tiempo promedio por inserción (eje Y) vs.  $N$  (eje X). Se espera que esta curva se estabilice en un valor constante para  $N$  suficientemente grande.
5. **Visualización del Tiempo Total:** Se generará un gráfico del tiempo total (eje Y) vs.  $N$  (eje X). Se verificará si la relación es aproximadamente lineal (pendiente constante).
6. **Comparación Cuantitativa:** Se comparará el tiempo promedio experimental estabilizado con el tiempo de una inserción sin redimensionamiento para evaluar la sobrecarga introducida por las redimensiones en promedio.

## 6 Conclusión Preliminar (Esperada)

Se espera concluir que los resultados experimentales respaldan firmemente la hipótesis. El análisis de los tiempos individuales mostrará la existencia de operaciones costosas ( $O(k)$ ) predichas por el análisis del peor caso, pero el análisis del tiempo total y promedio por operación demostrará que el rendimiento agregado de  $N$  inserciones escala linealmente con  $N$ , haciendo que el costo promedio por operación sea efectivamente constante ( $O(1)$ ), tal como predice el análisis amortizado. Esto subrayará la importancia del análisis amortizado para evaluar el rendimiento práctico de estructuras de datos que exhiben costos variables por operación, y justificará la eficiencia observada de los vectores dinámicos en la mayoría de las aplicaciones. Se reconocerá que el análisis del peor caso sigue siendo relevante para sistemas donde una única operación lenta no es tolerable (ej. sistemas de tiempo real estricto).