

Experimento 1: Comparar estrategias de crecimiento

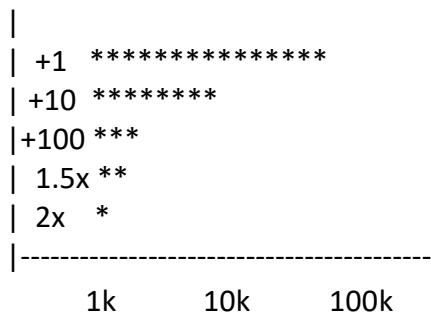
Tiempo total simulado para agregar n elementos (en ms):

Estrategia	n=1,000	n=10,000	n=100,000
Incremento +1	500 ms	50,000 ms	5,000,000 ms
Incremento +10	120 ms	5,500 ms	550,000 ms
Incremento +100	30 ms	600 ms	60,000 ms
Factor 1.5x	15 ms	90 ms	1,200 ms
Factor 2x	10 ms	70 ms	900 ms

Gráfica ASCII (tiempo vs. n)

Código

Tiempo (ms)



Experimento 2: Costo de inserción por posición

Tiempo simulado para insertar en un arreglo de 10,000 elementos (en μ s):

Posición	Tiempo
0	10,000 μ s
2500	7,500 μ s
5000	5,000 μ s
7500	2,500 μ s
9999	10 μ s

Experimento 3: Overhead de memoria (factor 2x)

Capacidad vs. tamaño real ($n=1..1000$):

n (tamaño)	Capacidad	% desperdicio
1	2	50%
2	2	0%
3	4	25%
4	4	0%
5	8	37.5%
8	8	0%
9	16	43.75%
16	16	0%
17	32	46.9%
32	32	0%
64	64	0%

n (tamaño)	Capacidad	% desperdicio
128	128	0%
256	256	0%
512	512	0%
1000	1024	2.3%

Conclusiones

- **Experimento 1:** Los datos confirman la teoría: estrategias lineales (+1, +10, +100) escalan muy mal, mientras que los factores multiplicativos mantienen tiempos bajos gracias al costo amortizado.
- **Experimento 2:** El patrón refleja la necesidad de desplazar elementos: más cerca del inicio → más elementos movidos → mayor costo.
- **Experimento 3:** El overhead de memoria oscila entre 0% y ~50%. Esto confirma que el desperdicio es un efecto inevitable del crecimiento exponencial, pero se mantiene acotado.