

Estructura de datos	Inserción	Borrado	Búsqueda	Cuando usarlo
Vector	Al final $O(1)$ En medio (n)	Al final $O(1)$ En medio $O(n)$	índice $O(1)$ sino $O(n)$	Para pequeños volúmenes de datos y borrado/inserción por el final.
Lista Enlazada	$O(1)$ Esperado	Principio $O(1)$ Final $O(n)$ En medio $O(n)$	$O(n)$	
Lista Doblemente Enlaza	$O(1)$	$O(1)$	$O(n)$	estructurade datos flexible y eficiente para aplicaciones que requieren inserciones y borrados arbitrarios
Pila	Solo Final $O(1)$	Solo Final $O(1)$	No hay	
Cola	Solo Final $O(1)$	Solo principio $O(1)$	No hay	
Arboles binarios	$O(\text{Log}n)$	$O(\text{Log}n)$	Si está equilibrado $O(n \log n)$ sino puede llegar a ser lineal	
AVL	$O(\text{Log}n)$	$O(\text{Log}n)$	$O(\log n)$	
Heap	$O(1)$ mejor caso, $O(\log n)$ peor caso debido a la altura			Es la estructura de datos más eficiente al ser un árbol completo en el que el nodo más alto tiene más prioridad.
Dispersión abierta	$O(1)$	$O(1)$	$O(1)$	Cuando se conozcan bien la naturaleza de las claves. Tener buen F.dispersion, dimensionar bien para no redispersar.
Dispersión cerrada	$O(1)$	$O(1)$	$O(1)$	Solventa las colisiones con cubetas, mas sencilla y utiliza mejor el espacio
Malla Regular	$O(1)$	$O(1)$		Eficientes si los datos están repartidos de forma uniforme, asi cualquier operación sera prácticamente $O(1)$
KD-Tree			$O(n^{1/2} + K)$ para K datos en el rectángulo(rango)	
QuadTrees				
Range trees				

Claves repetidas	Clave == dato	Clave j= dato
No	Set	Map
Si	MultiSet	Multimap

This document was created with Win2PDF available at <http://www.win2pdf.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.
This page will not be added after purchasing Win2PDF.