

## Tablas

### Tabla arreglos

Estructura	Indexación	Inserción	Modificación	Borrado	Búsqueda
Arreglo homogéneo (como en Java)	$O(1)$	$O(N)$	$O(1)$	$O(N)$	$O(N)$
Arreglo heterogéneo (como en Python)		$O(N)$ <i>*mediante concatenación</i>	$O(N)$ <i>*Mediante slicing + concatenación</i>	$O(N)$ <i>*Mediante slicing</i>	

### Tabla arreglos dinámicos

Estructura	Indexación	Inserción	Modificación	Borrado	Búsqueda
Arreglo dinámico homogéneo	$O(1)$	$O(1)$ si es en un extremo del arreglo según análisis amortizado, $O(N)$ en caso contrario	$O(1)$	$O(1)$ si es en un extremo del arreglo, $O(N)$ en caso contrario	$O(\log_2 N)$ si está ordenado, $O(N)$ en caso contrario
Arreglo dinámico heterogéneo			$O(1)$ si es en un extremo del arreglo según análisis amortizado, $O(N)$ en caso contrario		

### Tabla ordenamiento y búsqueda

Estructura	Indexación	Modificación	Inserción y Borrado	Búsqueda	Ordenamiento
Arreglo homogéneo	$O(1)$	$O(1)$	$O(N)$	$O(N)$ si no está ordenado	$O(N \cdot \log_2 N)$
Arreglo heterogéneo		$O(N)$		$O(\log_2 N)$ si está ordenado	

### Tabla listas enlazadas

Estructura	Indexación	Inserción	Modificación	Borrado	Búsqueda
Lista enlazada	$O(1)$ para el primer y último elemento, $O(N)$ para el resto de elementos				$O(N)$

## Tabla montículos binarios

Estructura	Indexación	Inserción	Modificación	Borrado	Búsqueda	Heapify
Montículo binario	$O(1)$ para la cima	$O(\log_2 N)$	$O(N)^*$	$O(\log_2 N)$ para la cima, $O(N)$ en caso contrario	$O(N)$	$O(N)^{**}$

\* Una modificación implicaría dos pasos: borrado (que incluye la búsqueda) + inserción

\*\* Se refiere a tomar un arreglo  $A$  de tamaño  $N$  y convertirlo en un montículo binario. La demostración puede encontrarse en la sección 6.3 de *Introduction to algorithms, 3rd edition*