

Tenemos un array: **lista**
y una variable con la ocupación: **ocupacion** (empieza en cero y se va incrementando)
y otra variable **MAX_DISPONIBLE**: con el total máximo admitido.

Métodos	Con huecos 	Sin huecos 	
		No ordenado 	Ordenado 
C añadir (create) 	Si (ocupacion < MAX_DISPONIBLE) recorro toda la lista si encuentro un hueco (null) → inserto el elemento y ocupacion++ eficiente  facillmentar 	Si (ocupacion < MAX_DISPONIBLE) inserto el elemento en lista[ocupacion++] eficiente  facillmentar 	Si (ocupacion < MAX_DISPONIBLE) insertoOrdenado el elemento en lista y ocupacion++ eficiente  facillmentar 
R buscar (read) 	Si (ocupacion >0) Recorro toda la lista si encuentro el elemento (equals) → lo devuelvo devuelvo una excepción de NoEncontrado eficiente  facillmentar 	Si (ocupacion >0) Recorro la lista hasta ocupacion -1 si encuentro el elemento (equals) → lo devuelvo devuelvo una excepción de NoEncontrado eficiente  facillmentar 	Si (ocupacion >0) Usar un metodo de buscarOrdenado y devolver el elemento o la Excepcion si no existe eficiente  facillmentar 
U modificar (update) 	Si (ocupacion >0) Recorro toda la lista si encuentro el elemento (equals) → lo modifico devuelvo una excepción de NoEncontrado eficiente  facillmentar 	Si (ocupacion >0) Recorro la lista hasta ocupacion -1 si encuentro el elemento (equals) → lo modifico devuelvo una excepción de NoEncontrado eficiente  facillmentar 	Si (ocupacion >0) Usar un metodo de buscarOrdenado y modificarlo el elemento o la Excepcion si no existe eficiente  facillmentar 
D borrar (delete) 	Si (ocupacion >0) Recorro toda la lista si encuentro el elemento (equals) → lo borro y ocupacion-- devuelvo una excepción de NoEncontrado eficiente  facillmentar 	Si (ocupacion >0) Recorro la lista hasta ocupacion -1 si encuentro el elemento (equals) lo borro ocupacion-- //desplazo el resto de la lista hacia la izquierda recorro (i) des la posición actual hasta ocupacion -1 lista[i] = lista[i+1] devuelvo una excepción de NoEncontrado eficiente  facillmentar 	Si (ocupacion >0) Usar un metodo de buscarOrdenado si encuentro el elemento lo borro ocupacion-- desplazo el resto de la lista hacia la izquierda devuelvo una excepción de NoEncontrado eficiente  facillmentar 