

métodos	Con huecos	Sin huecos	
		No ordenado	Ordenado
<b>C    añadir (create)</b>	Si ( <b>ocupacion</b> < <b>MAX_DISPONIBLE</b> ) recorro toda la <b>lista</b> si encuentro un hueco (null) → <b>inserto</b> el elemento y <b>ocupacion++</b>  eficiente3   facillmentar2	Si ( <b>ocupacion</b> < <b>MAX_DISPONIBLE</b> ) <b>inserto</b> el elemento en <b>lista[ocupacion++]</b>  eficiente1   facillmentar1	Si ( <b>ocupacion</b> < <b>MAX_DISPONIBLE</b> ) <b>insertoOrdenado</b> el elemento en <b>lista</b> y <b>ocupacion++</b>  eficiente2   facillmentar3
<b>R    buscar (read)</b>	Si (ocupacion>0) Recorro toda la <b>lista</b> si encuentro el elemento (equals) → lo <b>devuelvo</b> devuelvo una excepción de NoEncontrado  eficiente3   facillmentar1	Si (ocupacion>0) Recorro la <b>lista</b> hasta <b>ocupacion -1</b> si encuentro el elemento (equals) → lo <b>devuelvo</b> devuelvo una excepcion de NoEncontrado  eficiente2   facillmentar1	Si (ocupacion>0) Usar un metodo de <b>buscarOrdenado</b> y <b>devolver</b> el elemento o la Excepcion si no existe  eficiente1   facillmentar3
<b>U    modificar (update)</b>	Si (ocupacion>0) Recorro toda la <b>lista</b> si encuentro el elemento (equals) → lo <b>modifico</b> devuelvo una excepción de NoEncontrado  eficiente3   facillmentar1	Si (ocupacion>0) Recorro la <b>lista</b> hasta <b>ocupacion -1</b> si encuentro el elemento (equals) → lo <b>modifico</b> devuelvo una excepcion de NoEncontrado  eficiente2   facillmentar1	Si (ocupacion>0) Usar un metodo de <b>buscarOrdenado</b> y <b>modificarlo</b> el elemento o la Excepcion si no existe  eficiente1   facillmentar3
<b>D    borrar (delete)</b>	Si (ocupacion>0) Recorro toda la <b>lista</b> si encuentro el elemento (equals) → lo <b>borro</b> y <b>ocupacion--</b> devuelvo una excepción de NoEncontrado  eficiente3   facillmentar1	Si (ocupacion>0) Recorro la <b>lista</b> hasta <b>ocupacion -1</b> si encuentro el elemento (equals) lo <b>borro</b> <b>ocupacion--</b> <b>//desplazo</b> el resto de la lista hacia la izquierda <b>recorro (i)</b> des la <b>posicion actual</b> hasta <b>ocupacion -1</b> <b>lista[i] = lista[i+1]</b> devuelvo una excepcion de NoEncontrado  eficiente2   facillmentar2	Si (ocupacion>0) Usar un metodo de <b>buscarOrdenado</b> <b>si encuentro el elemento</b> lo <b>borro</b> <b>ocupacion--</b> <b>desplazo</b> el resto de la lista hacia la izquierda devuelvo una excepcion de NoEncontrado  eficiente1   facillmentar3
<b>MoverMensaje (Carpeta origen, Carpeta destino, int codigo)</b>	Si ( <b>ocupacion(destino)</b> < <b>MAX_DISPONIBLE</b> ) <b>mensaje = buscar</b> (Origen, codigo) <b>Añadir</b> (Destino, mensaje) <b>Borrar</b> (Origen, mensaje)		

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.