**Búsqueda 1**: Buscar el elemento comparándolo secuencialmente (de ahí su nombre) con cada elemento del array hasta encontrarlo, o hasta que se llegue al final.

**Búsqueda 2**: Con un array ordenado, se compara el elemento a buscar con un elemento cualquiera del array (empezando por el elemento central): si el valor de éste es mayor que el del elemento buscado se repite el procedimiento en la parte del array que va desde el inicio de éste hasta el elemento tomado, en caso contrario se toma la parte del array que va desde el elemento tomado hasta el final. De esta manera obtenemos intervalos cada vez más pequeños, hasta que se obtenga un intervalo indivisible (de 1 sólo elemento). Si el elemento no se encuentra dentro de este último entonces se deduce que el elemento buscado no se encuentra en todo el array.

**Ordenación 1**: revisa cada elemento del array que va a ser ordenado comparándolo con el siguiente elemento, intercambiándolos de posición si están en el orden equivocado.

Será necesario revisar varias veces toda la lista hasta que no se necesiten más intercambios, lo cual significa que la lista está ordenada.

**Ordenación 2**: Elegir un elemento del array de elementos a ordenar, al que llamaremos **pivote**.

Resituar los demás elementos del array a cada lado del pivote, de manera que a un lado queden todos los menores que él, y al otro los mayores. Los elementos iguales al pivote pueden ser colocados tanto a su derecha como a su izquierda, dependiendo de la implementación deseada. En este momento, el pivote ocupa exactamente el lugar que le corresponderá en la lista ordenada.

El array queda separado en dos subarrays, uno formado por los elementos a la izquierda del pivote, y otro por los elementos a su derecha.

Repetir este proceso de forma **recursiva** para cada subarray mientras éstos contengan más de un elemento. Una vez terminado este proceso todos los elementos estarán ordenados.