|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Repetición de clave | No hay repetición de clave | Estar ordenado por clave | Estar ordenado por orden de inserción | Par clave valor | Solo clave |
| ArrayList | X | X |  |  |  | X |
| LinkedList | X | X |  |  |  | X |
| TreeSet |  | X | X |  |  | X |
| LinkedHashSet  (Extend HashSet) |  | X |  | X |  | X |
| HashSet |  | X |  | X |  | X |
| TreeMap |  | X | X |  | X | X |
| HashMap |  | X |  | X | x | X |
| LinkedHashMap |  | X |  | X | x | X |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | List | Set | Map | devuelve |
| Colección vacía | .isEmpty() | .isEmpty() | .isEmpty() | boolean |
| Tamaño | .size() | .size() | .size() | int |
| Borrar colección | .clear() | .clear() | .clear() | void |
| Añadir | .add(E e) | .add(E e) | .put(K key, V value) | boolean/V |
| Eliminar por clave | - | - | .remove(Object key) | V |
| Eliminar | .remove(Object o) | .remove(Object o) | .remove(Object key, Object value) | boolean |
| Buscar clave | .contains(Object o) | .contains(Object o) | .containsKey(Object key) | boolean |
| Buscar valor | - | - | .containsValue(Object value) | boolean |
| Modificar | .set(int index, E element)\* | - | .replace(K key, V oldValue, V newValue) | E/boolean |
| Obtener posición | indexOf(Object o)\* | - | - | int |
| Obtener valor | .get(int index)\* |  | .get(Object key) | E/V |
| Convertir en set |  |  | .entrySet() | Set<Map.Entry<K,V> |
| Convertir en set claves | - | - | .keySet() | Set< K> |
| Convertir en set valores | - | - | .values() | Collection<V> |
| Iterar | .iterator() | .iterator() |  | Iterator<E> |
| Iterar inverso | .descendingIterator() | .descendingIterator() |  | Iterator<E> |
| Ordenar | .sort(Comparator<? Super E c> | - | - | void |
| Asociar | - | - | .put(K key, V value) | V |
| Comparar |  | .comparator() | .comparator() | Comparator<? Super E/K> |