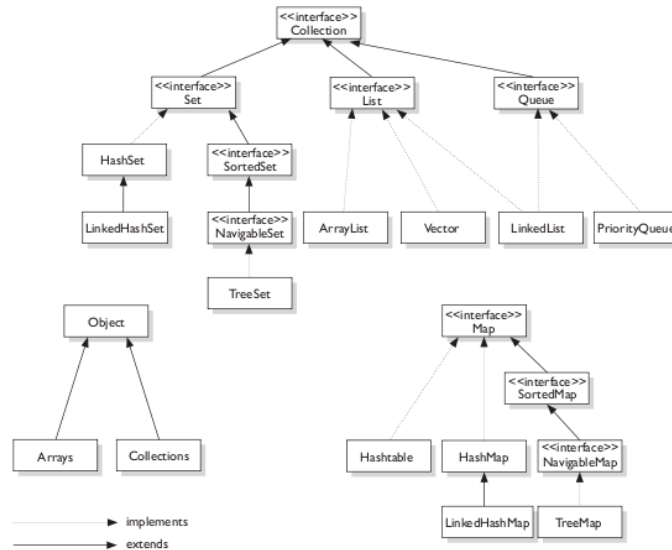


## Collections Java

Ese artículo describe el framework Collections de Java.



Jerarquía de Interfaces y clases [1]

### ArrayList

Es un tipo arreglo que incrementa y decreta sus elementos dinámicamente. Cuenta con una rápida iteración y acceso aleatorio. No se debe utilizar cuando se realizarán una amplia cantidad de operaciones de inserción y eliminación de elementos.

### Vector

Básicamente tiene las mismas características que un ArrayList, simplemente sus métodos son sincronizados para la utilización de multihilos seguros.

### LinkedList

Es similar a un ArrayList excepto que sus elementos están doblemente ligados lo cual permite agregar o remover elementos en el principio o fin de la lista, es una buena opción cuando se necesite implementar una estructura tipo cola o pila.

### HashSet

Utiliza el hashcode del objeto que esta siendo insertado, se puede implementar el método hashCode() de los objetos para un rendimiento más eficiente. Este tipo de estructuras no permite valores duplicados y no guarda un orden.

### LinkedHashSet

Es la versión ordenada de un HashSet que mantiene doblemente ligados los elementos que contiene. Esta clase se debe utilizar cuando sea de utilidad el orden en que fueron insertados los elementos. Este tipo de estructuras no permite valores duplicados.

### TreeSet

Garantiza el orden de los elementos de manera ascendente, opcionalmente se puede definir el orden de los elementos definiendo los métodos Comparable o Comparator. Este tipo de estructuras no permite valores duplicados.

### HashMap

Es un tipo de diccionario el cual se relaciona "clave-valor", este tipo de estructura no cuenta con un orde. HashMap permite un valor null para una clave y multiples valores null en la colección.

### Hashtable

Hashtable tiene las mismas características que un HashMap excepto que sus métodos son sincronizados, haciéndolos seguros para el manejo de multihilos y en consecuencia mas lentos. Su otra gran diferencia es que no permite valores null en la clave o en la colección

### LinkedHashMap

Es similar a HashMap excepto que mantiene el orden de inserción. Es mas lenta que un HashMap en las operación de agregar o eliminar elementos, pero su velocidad de iteración es mayor que la de un HashMap.

### TreeMap

Es un mapa que garantiza el orden de los elementos de manera ascendente, opcionalmente se puede definir el orden de los elementos definiendo los métodos Comparable o Comparator.

### PriorityQueue

Su propósito es la creación de colas con las definición de prioridad para entradas y salidas. Sus elementos son ordenados. También se puede implementar un Comparator el cual representara la prioridad de cada elemento.