

unam
donde se construye el
futuro



FES Aragón
somos todos

Estructura de Datos

Ing. Miguel Ángel Sánchez Hernández

La API de Colecciones

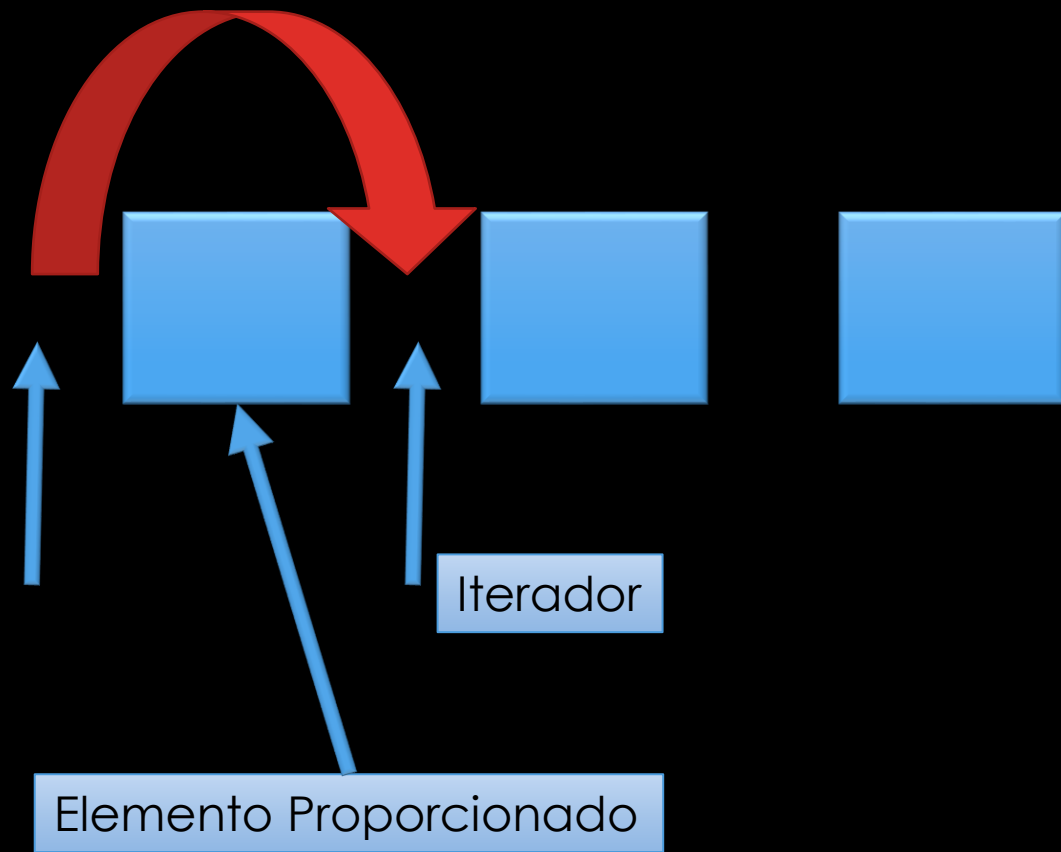
Como parte de Java, se proporciona una librería, conocida como API de Colecciones (Collection), su mayor parte reside en el paquete de java.util.

Las interfaces fundamentales para las clases de colección en Java son, la interfaz Collection e Iterator.

```
public interface Collection<E>{  
    boolean add(E elemento);  
    Iterator<E> iterator()  
}
```

```
public interface Iterator<E>{  
    E next();  
    boolean hasNext();  
    void remove();  
}
```

Iterator



```
Iterator<String> it=c.iterator();  
it.next(); //saltamo al primer elemento  
it.remove(); eliminamos
```

No se debe hacer
`it.remove(); eliminamos`
`it.remove(); eliminamos`

Si se debe hacer
`it.remove(); eliminamos`
`it.next();`
`it.remove(); eliminamos`

Estructura de datos que ofrece Java

LinkedList: Lista doblemente enlazada, su posición de los elementos tiene importancia, es decir se añade el elemento al final de la lista. Esto ocasiona que no se permita acceso directo, es decir si se quiere acceder al n-ésimo elemento se tiene que recorrer n-1 elementos, por eso si se necesita acceder a un elemento proporcionando un índice entero no se debe ocupar LinkedList.


ArrayList: Sucesión indizada que crece y decrece dinámicamente, es decir encapsula una matriz de objetos que se reasigna dinámicamente, este no está sincronizado, esta implementa también la interfaz de List. Si se necesita sincronización es más seguro ocupar la clase Vector.

Si tenemos pocos datos y no importa el coste de poder acceder a los datos por get,set y solo ocupar la eficiencia de insertar y eliminar datos LinkedList es la opción, pero si nos interesa el coste de get y set y son muchos datos ArrayList es la opción.

HashSet: Es una colección sin orden que rechaza duplicados, es decir una estructura de datos que es la **tabla dispersa**, están calculan un entero, que se llama código disperso para cada objeto, este se deriva de algún campo del objeto, esto es por medio de la clase hashCode.

Una tabla dispersa es una matriz de lista enlazada, en donde cada lista recibe el nombre de compartimiento, para averiguar donde se encuentra el objeto, se calcula su código de dispersión.


Si tenemos un código de dispersión 76268 y se tienen 128 compartimientos, su lugar que ocupa es $76268 \% 128$ es 108, es decir $76268 = 595 * 128 + 108$, tomamos el residuo. Si está vacío el compartimiento se inserta, si no es el caso hay una colisión. Java implementa esta estructura de tipo conjunto, es decir no admite duplicados, por lo tanto, si hay una colisión no se inserta el objeto.



TreeSet: Es similar a un conjunto disperso, pero tiene una mejora adicional que es un conjunto de árbol siendo esta una colección ordenada, los elementos se insertan en la colección en cualquier orden, y cuando se itera, los valores se presentan automáticamente en orden.

Para saber cómo debe ordenarlos las clases deben implementar la interfaz Comparable, y los valores que importan son:

- 0 si a y b son iguales
- >0 si a va después de b
- <0 si a va antes de b



Colas de Prioridad: Recuperan sus elementos por orden de clasificación después de insertar esos elementos por orden arbitrario, cuando llamamos a remove tenemos siempre el elemento más pequeño, estas no ordenan todos sus elementos, si los recorremos no estarán del todo ordenados. Estas ocupan una estructura de datos denominada cumulo, no es otra cosa que un árbol binario autoorganizado en donde la adición y eliminación hacen que el elemento más pequeño baje hasta la raíz. También podemos ocupar Comparable como se hizo en TreeSet.