



4. HashTable

📅 Date	Empty
⚙️ Status	Not started
📄 Type	Empty
Units	4. Estructuras de almacenamiento

HashTable

Esta estructura de datos utiliza una función hash para identificar los datos mediante una clave asociada. Este tipo de colecciones tienen un tamaño modificable ya que incrementa / decrementa cuando se añaden o borran elementos. Una de las grandes características de este tipo de colecciones es que es sincronizable, ya que puede ser accedido por dos hilos al mismo tiempo

Los constructores que se pueden utilizar para crear una HashTable son:

```
Hashtable listaHash = new Hashtable(); // se puede identificar el tipo de la clave asociada y el tipo del objeto
<Integer, Integer> listaHash = new Hashtable<Integer, Integer>();
```

Los métodos más utilizados son:

Los metodos mas utilizados son:

```
// Vaciar la tabla "t": t.clear(); // Comprobar si en la tabla "t" existe el valor "a": t.contains(a) → boolean // Comprobar si en la tabla "t" existe el índice de cadena "s": t.containsKey(s) → boolean // Obtiene en la tabla "t" el valor "n" con el índice "s": n = t.get(s); // Conocer si la tabla "t" está vacía: t.isEmpty() → boolean
Añadir el valor "a" en "t" con el índice "s" o reemplazar el índice "s" si ya existe: // Añadir t.put(s, a); // Reemplazar b por a b = t.put(s, a); // Eliminar el valor de "t" que ocupa el índice "s"; a = t.remove(s); // Obtener el número de elementos que tiene la tabla "t": t.size() → int
```

Recorrer elementos

Para poder recorrer los elementos de una lista HasstTable hay que obtener las claves asociadas a cada uno, y con ellas obtener el valor. Para ello se utilizan los métodos:

```
Enumeration<tipo_clave> elementos = lista.elements(); Enumeration<tipo_clave> clavesElementos = lista.keys();
```

Con el primero de los métodos se obtiene un enumeration con todos los elementos. En este caso no se puede aplicar un foreach para poder recorrerlo por lo que se utiliza un while:

```
Enumeration<Integer> elementos = hashtable.elements(); while (elementos.hasMoreElements()) { System.out.println(elementos.nextElement().toString()); }
```

Con el segundo método se obtienen todas las claves de los elementos de la lista. Con la misma estructura que el caso anterior se puede aplicar un while donde se saca objeto a objeto mediante sus claves

```
Enumeration<Integer> claves = hashtable.keys(); while (claves.hasMoreElements()) { Integer elemento = hashtable.get(claves.nextElement()); System.out.println(elemento.toString()); }
```

Realizar búsquedas

Si se quiere borrar un elemento que cumple una determinada condición

```
Enumeration<Integer> elementos = hashtable.elements(); while  
(elementos.hasMoreElements()) { if (elementos.nextElement() =  
= 1){ hashtable.remove() } }
```

HashMap

El concepto de uso y métodos son los mismos que la colección HashTable, con la diferencia que no se trata de una colección sincronizarle