

Guía de Colecciones en Java

Arrays comunes

Declarar y crear

```
int[] numeros = new int[5]; // Array de tamaño 5 con valores por defecto (0)
String[] nombres = {"Ana", "Luis", "Carlos"}; // Inicialización directa
```

Asignar o modificar elementos

```
numeros[0] = 10;
nombres[2] = "Lucía";
```

Eliminar elementos

No se pueden eliminar elementos de un array fijo (su tamaño es constante). Solo puedes **reemplazar** un valor por uno neutral:

```
numeros[2] = 0; // Por convención, "eliminar"
nombres[1] = null; // Reemplazar por null
```

Guía de Colecciones en Java

Arrays comunes

Recorrer

For clásico (por índice)

```
for (int i = 0; i < numeros.length; i++) {  
    System.out.println(numeros[i]);  
}
```

For-each (enhanced for)

```
for (int numero : numeros) {  
    System.out.println(numero);  
}
```

```
for (String nombre : nombres) {  
    System.out.println(nombre);  
}
```



Guía de Colecciones en Java

List (ArrayList, LinkedList, etc.)

Declarar y agregar

```
List<String> lista = new ArrayList<>();  
lista.add("Manzana");  
lista.add("Banana");  
lista.add(1, "Naranja"); // Inserta en la posición 1
```

Modificar elementos

```
lista.set(0, "Pera"); // Reemplaza el elemento en el índice 0
```

Eliminar elementos

```
lista.remove(1);           // Elimina por índice  
lista.remove("Pera");      // Elimina por objeto  
lista.clear();             // Elimina todos los elementos
```



Guía de Colecciones en Java

List (ArrayList, LinkedList, etc.)

Recorrer

For-each

```
for (String fruta : lista) {  
    System.out.println(fruta);  
}
```

Con índice

```
for (int i = 0; i < lista.size(); i++) {  
    System.out.println(lista.get(i));  
}
```

Con iterador

```
Iterator<String> it = lista.iterator();  
while (it.hasNext()) {  
    System.out.println(it.next());  
}
```



Guía de Colecciones en Java

Set (HashSet, TreeSet, LinkedHashSet)

Declarar y agregar

```
Set<String> conjunto = new HashSet<>();  
conjunto.add("Rojo");  
conjunto.add("Azul");
```

Modificar

No se puede modificar directamente un elemento, hay que eliminar y volver a agregar:

```
if (conjunto.contains("Rojo")) {  
    conjunto.remove("Rojo");  
    conjunto.add("Verde");  
}
```

Eliminar

```
conjunto.remove("Azul"); // Elimina un elemento  
conjunto.clear(); // Elimina todos los elementos
```



Guía de Colecciones en Java

Set (HashSet, TreeSet, LinkedHashSet)

Recorrer

For-each

```
for (String color : conjunto) {  
    System.out.println(color);  
}
```

Con iterador

```
Iterator<String> it = conjunto.iterator();  
while (it.hasNext()) {  
    System.out.println(it.next());  
}
```



Guía de Colecciones en Java

Map (HashMap, TreeMap, LinkedHashMap)

Declarar y agregar

```
Map<Integer, String> mapa = new HashMap<>();  
mapa.put(1, "Uno");  
mapa.put(2, "Dos");
```

Modificar

```
mapa.put(1, "NuevoUno"); // Reemplaza el valor de la clave 1
```

Eliminar

```
mapa.remove(2); // Elimina por clave  
mapa.clear(); // Borra todo
```



Guía de Colecciones en Java

Map (HashMap, TreeMap, LinkedHashMap)

Recorrer

Claves

```
for (Integer clave : mapa.keySet()) {  
    System.out.println(clave);  
}
```

Valores

```
for (String valor : mapa.values()) {  
    System.out.println(valor);  
}
```

Pares clave-valor

```
for (Map.Entry<Integer, String> entry : mapa.entrySet()) {  
    System.out.println(entry.getKey() + " = " + entry.getValue());  
}
```


Guía de Colecciones en Java

TreeMap

Declarar y agregar

```
TreeMap<String, Integer> edades = new TreeMap<>();  
edades.put("Ana", 30);  
edades.put("Carlos", 25);  
edades.put("Luis", 40);
```

Modificar

```
edades.put("Carlos", 28); // Actualiza el valor asociado a "Carlos"
```

Eliminar

```
edades.remove("Ana"); // Elimina por clave  
edades.clear(); // Elimina todos los elementos del TreeMap
```



Guía de Colecciones en Java

TreeMap

Consultar elementos

```
edades.get("Luis");           // Devuelve 40
edades.containsKey("Carlos"); // true
edades.containsValue(28);     // true
edades.isEmpty();            // false
edades.size();                // Cantidad de pares clave-valor
```

Recorrer claves (ordenadas)

```
for (String nombre : edades.keySet()) {
    System.out.println(nombre);
}
```

Recorrer valores

```
for (Integer edad : edades.values())
{
    System.out.println(edad);
}
```

Recorrer pares clave-valor

```
for (Map.Entry<String, Integer> entrada : edades.entrySet()) {
    System.out.println(entrada.getKey() + " -> " + entrada.getValue());
}
```