

1 “Colección” (Collection) → el concepto general

En Java, **Collection** es una interfaz base del framework de colecciones (`java.util.Collection`) que representa un **grupo de objetos**, también llamados *elementos*.

Define las **operaciones más generales** que cualquier estructura de datos debe tener:

```
add(), remove(), contains(), size(), isEmpty(), iterator()
```

👉 En otras palabras:

Una **colección** es cualquier conjunto de elementos agrupados en un **objeto**: lista, conjunto, cola, etc.

Ejemplo:

```
Collection<String> nombres = new ArrayList<>();
nombres.add("Ana");
nombres.add("Luis");
```

Aquí `nombres` es una **colección**, pero concretamente está implementada como una **lista** (`ArrayList`).

2 “Lista” (List) → un tipo específico de colección

Una **lista** es una **subinterfaz** de `Collection`.

Es más especializada y **mantiene el orden** de los elementos, además de permitir **accesos por posición (índice)** y **duplicados**.

```
List<String> nombres = new ArrayList<>();
nombres.add("Ana");
nombres.add("Luis");
nombres.add("Ana"); // permitido, puede haber duplicados
System.out.println(nombres.get(1)); // "Luis"
```

👉 En resumen:

Una **lista** es una **colección ordenada**, donde los elementos tienen una posición y pueden repetirse.

3 Jerarquía simplificada

```
      Collection
     /   |   \
   List  Set  Queue
   /  \
ArrayList LinkedList
```

- `Collection` → interfaz general
- `List`, `Set`, `Queue` → tipos específicos de colecciones
- `ArrayList`, `LinkedList`, `HashSet`, `PriorityQueue`, etc. → implementaciones concretas

4 Diferencias clave

Característica	<code>Collection</code>	<code>List</code>
Tipo	Interfaz genérica	Subinterfaz de <code>Collection</code>
Orden	No garantiza orden	Mantiene el orden de inserción
Duplicados	Depende de la implementación	Permitidos
Acceso por índice	No	Sí (<code>get(int index)</code>)
Ejemplos de implementación	<code>HashSet</code> , <code>ArrayList</code> , <code>PriorityQueue</code>	<code>ArrayList</code> , <code>LinkedList</code> , <code>Vector</code>

5 Ejemplo visual rápido

```
Collection<Integer> numeros = new HashSet<>();
numeros.add(1);
numeros.add(2);
numeros.add(1); // Ignorado (no permite duplicados)
```

```
List<Integer> lista = new ArrayList<>();
lista.add(1);
lista.add(2);
lista.add(1); // Permitido (guarda duplicados)

System.out.println("Colección: " + numeros); // [1, 2]
System.out.println("Lista: " + lista);        // [1, 2, 1]
```

✓ En resumen

Concepto	Qué representa	Ejemplo típico
Colección	Grupo genérico de elementos	<code>Collection<Producto></code>
Lista	Colección ordenada, con duplicados y acceso por índice	<code>List<Producto></code> (<code>ArrayList</code>)