

1. Qu es una coleccin?

En programacin, una coleccin es simplemente un grupo de elementos.

Ejemplos:

- Una bolsa de frutas ["Manzana", "Banana", "Pera"]
- Una lista de nmeros [4, 7, 2, 9]

En Java, hay una interfaz llamada Collection, que define las operaciones ms comunes para cualquier coleccin:

- Agregar elementos
- Ver si un elemento est
- Saber cuntos hay

Pero no dice cmo se hace eso. Solo define qu se puede hacer.

Qu es una interfaz en Java?

Una interfaz es un contrato que dice: Si vas a ser una coleccin, tens que saber hacer estas cosas. No te dice cmo las hacs, eso lo define la clase que la implemente.

2. Qu significa `public interface List<T> extends Collection<T>`?

Vamos parte por parte:

```
public interface List<T> extends Collection<T> {
```

```
// mtodos  
}
```

- `interface List<T>`: esto define una interfaz de listas, donde `T` es un tipo genérico (puede ser `Integer`, `String`, etc.).
- `extends Collection<T>`: est diciendo una lista también es una colección.
- O sea: la interfaz `List` hereda de la interfaz `Collection`, y le agrega cosas nuevas.

Entonces una lista es un tipo de colección, que además puede:

- Agregar al principio o al final
- Buscar elementos por su posición

3. ¿Qué es un Nodo?

Cuando hablamos de listas enlazadas, cada elemento se guarda dentro de un objeto llamado `Nodo`.

Ese nodo tiene dos cosas:

```
private class Nodo {  
    int valor; // el dato que guarda este nodo  
  
    Nodo sig; // el nodo que viene después (siguiente)  
  
    Nodo(int v) {  
        valor = v; // cuando creo un nodo, le asigno su valor  
    }  
}
```

```
}
```

Entonces esta línea:

```
Nodo(int v) { valor = v; }
```

Significa:

- Estoy creando un constructor (es un método especial que se usa cuando haces `new Nodo(...)`).
- `v` es el valor que le paso como dato.
- Ese valor se guarda en el atributo `valor`.

Ejemplo:

```
Nodo n = new Nodo(5);
```

Eso crea un nodo que guarda el número 5.

4. ¿Qué es un Iterador?

Un iterador es un objeto que te permite recorrer una colección sin tener que saber cómo está hecha por dentro.

Imagina que tienes una caja con papeles y quieres leer uno por uno:

- El iterador es tu dedo que va señalando uno a uno, y después avanza al siguiente.

En Java, el iterador se representa así:

```
public interface Iterator<E> {  
  
    boolean hasNext(); // ¿Queda algo más?
```

```
E next();      // Dame el siguiente  
}
```

5. Cmo se ve todo esto en cdigo?

```
import java.util.Iterator;  
  
public class ListaDeInts implements Iterable<Integer> {  
  
    private Nodo primero;  
  
    private class Nodo {  
        int valor;  
        Nodo sig;  
  
        Nodo(int v) {  
            valor = v;  
        }  
    }  
  
    public void agregarAdelante(int elem) {  
        Nodo nuevo = new Nodo(elem);  
        nuevo.sig = primero;  
        primero = nuevo;  
    }  
}
```

```
// El iterador
```

```
public Iterator<Integer> iterator() {  
    return new Iterador();  
}
```

```
private class Iterador implements Iterator<Integer> {
```

```
    Nodo actual = primero;
```

```
    public boolean hasNext() {  
        return actual != null;  
    }
```

```
    public Integer next() {  
        int val = actual.valor;  
        actual = actual.sig;  
        return val;  
    }
```

```
}
```

```
// Ejemplo de uso
```

```
public static void main(String[] args) {  
    ListaDeInts lista = new ListaDeInts();  
    lista.agregarAdelante(5);  
    lista.agregarAdelante(10);  
    lista.agregarAdelante(20);  
  
    for (int x : lista) {
```

```
System.out.println(x); // imprime 20, 10, 5
```

```
}
```

```
}
```

```
}
```

En resumen:

Concepto	Qu es?	Ejemplo en Java
-----	-----	-----
Coleccin	Grupo de elementos	<code>Collection<T></code>
Interfaz	Contrato de comportamiento	<code>interface List<T></code>
Nodo	Elemento de una lista	<code>int valor; Nodo sig;</code>
Iterador	Objeto para recorrer	<code>Iterator<T></code>