En Java, una **List** es una colección ordenada que permite almacenar elementos de manera secuencial y acceder a ellos a través de un índice.

**Principales Características de List:**

**Orden**: Los elementos mantienen el orden en que fueron agregados.

**Duplicados**: Permite almacenar elementos duplicados.

**Acceso por Índice**: Los elementos pueden ser accedidos mediante su índice.

Existen varias implementaciones de la interfaz List:

**ArrayList:** Implementa una lista utilizando un Array dinámico.

**LinkedList:** Es más eficiente que Arraylist ya que cuando se realizan muchas inserciones o eliminaciones en posiciones intermedias o al principio.

**Metodos específicos de list:**

La interfaz List proporciona varios métodos útiles para manipular los elementos en la lista. A continuación, mostramos algunos de los más importantes con ejemplos.

### Add(E element) Agrega un elemento al final de la lista.

List<String> lista = new ArrayList<>();

lista.add("Java");

lista.add("Python");

### 2Get (int index): Devuelve el elemento en la posición especificada.

### String lenguaje = lista.get(0); *// Devuelve "Java"*

### 3Remove ( int index): Elimina el elemento en la posición especificada.

lista.remove(1); *// Elimina "Python"*

### 4.Size (): Devuelve el número de elementos en la lista.

int tamano = lista.size(); *// Devuelve 1*

### 5.IndexOf (Object o): Devuelve el índice de la primera aparición de un elemento, o -1 si no está presente.

int indice = lista.indexOf("Java"); *// Devuelve 0*

Ejemplo de Array list:

package Explicaciontp;  
  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class EjemploArrayList {  
 public static void main(String[] args) {  
 // Crear una lista de Strings  
 List<String> Personitas = new ArrayList<>();  
  
 // Agregar elementos a la lista  
 Personitas.add("Paco");  
 Personitas.add("Julio");  
 Personitas.add("Tomas");  
  
 // Acceder a un elemento por su índice  
 System.*out*.println("El nombre en el índice 1 es: " + Personitas.get(1));  
  
 // Recorrer la lista e imprimir los elementos  
 for (String nombre : Personitas) {  
 System.*out*.println(nombre);  
 }  
  
 // Eliminar un elemento  
 Personitas.remove(0); // Elimina "Paco"  
 System.*out*.println("Después de eliminar, la lista es: " + Personitas);  
 }  
}

Ejemplo LinkedList:

package Explicaciontp;  
  
import java.util.LinkedList;  
import java.util.List;  
  
public class EjemploLinkedList {  
  
 public static void main(String[] args) {  
 // Crear una LinkedList de enteros  
 List<Integer> Numeros = new LinkedList<>();  
  
 // Agregar elementos  
 Numeros.add(10);  
 Numeros.add(20);  
 Numeros.add(30);  
  
 // Añadir un elemento al inicio  
 ((LinkedList<Integer>) Numeros).addFirst(5);  
  
 // Recorrer la lista e imprimir los elementos  
 for (Integer numero : Numeros) {  
 System.*out*.println(numero);  
 }  
  
 // Eliminar un elemento del inicio  
 ((LinkedList<Integer>) Numeros).removeFirst();  
 System.*out*.println("Después de eliminar el primer elemento: " + Numeros);  
 }  
 }