
```
import java.util.ArrayList;

public class ConjuntoADT<T> {
  private ArrayList<T> datos;

  public ConjuntoADT() {
    this.datos = new ArrayList<>();
  }
  public int longitud() {
    return this.datos.size();
  }
  public boolean contiene(T elemento) {
    return this.datos.contains(elemento);
  }
  public void agregar(T elemento) {
    if (!this.contiene(elemento)) {
      this.datos.add(elemento);
    }
  }
  public void eliminar(T elemento) {
    if (this.contiene(elemento)) {
      this.datos.remove(elemento);
    } else {
      System.out.println("El elemento a eliminar no existe.");
    }
  }
  public boolean equals(ConjuntoADT<T> otroConjunto) {
    return this.datos.containsAll(otroConjunto.datos) && otroConjunto.datos.containsAll(this.datos);
  }
  public boolean esSubConjunto(ConjuntoADT<T> otroConjunto) {
    return otroConjunto.datos.containsAll(this.datos);
  }
  public ConjuntoADT<T> union(ConjuntoADT<T> otroConjunto) {
    ConjuntoADT<T> nuevoConjunto = new ConjuntoADT<>();
    nuevoConjunto.datos.addAll(this.datos);
    for (T elemento : otroConjunto.datos) {
      if (!nuevoConjunto.contiene(elemento)) {
        nuevoConjunto.agregar(elemento);
      }
    }
    return nuevoConjunto;
```

```
}
public ConjuntoADT<T> interseccion(ConjuntoADT<T> otroConjunto) {
  ConjuntoADT<T> nuevoConjunto = new ConjuntoADT<>();
  for (T elemento : this.datos) {
    if (otroConjunto.contiene(elemento)) {
      nuevoConjunto.agregar(elemento);
    }
  }
  return nuevoConjunto;
}
public ConjuntoADT<T> diferencia(ConjuntoADT<T> otroConjunto) {
  ConjuntoADT<T> nuevoConjunto = new ConjuntoADT<>();
  for (T elemento : this.datos) {
    if (!otroConjunto.contiene(elemento)) {
      nuevoConjunto.agregar(elemento);
    }
  }
  return nuevoConjunto;
}
@Override
public String toString() {
  return datos.toString();
}
public static void main(String[] args) {

  ConjuntoADT<Integer> conjuntoA = new ConjuntoADT<>();
  ConjuntoADT<Integer> conjuntoB = new ConjuntoADT<>();

  conjuntoA.agregar(1);
  conjuntoA.agregar(2);
  conjuntoA.agregar(3);
  conjuntoB.agregar(3);
  conjuntoB.agregar(4);
  conjuntoB.agregar(5);

  System.out.println("Conjunto 1: " + conjuntoA);
  System.out.println("Conjunto 2: " + conjuntoB);

  System.out.println("Conjunto 1 contiene el 2: " + conjuntoA.contiene(2));

  // Se eliminar un elemento
  conjuntoA.eliminar(2);
  System.out.println("Conjunto 1 después de eliminar 2: " + conjuntoA);

  // Operaciones de unión, intersección y diferencia
  ConjuntoADT<Integer> unionAB = conjuntoA.union(conjuntoB);
  ConjuntoADT<Integer> interseccionAB = conjuntoA.interseccion(conjuntoB);
  ConjuntoADT<Integer> diferenciaAB = conjuntoA.diferencia(conjuntoB);
```

```
        System.out.println("Unión de 1 y 2: " + unionAB);
        System.out.println("Intersección de 1 y 2: " + interseccionAB);
        System.out.println("Diferencia de 1 y 2: " + diferenciaAB);

        // Aqui se hace la corroboracion de si si un conjunto es subconjunto de otro
        System.out.println("1 es subconjunto de 2: " + conjuntoA.esSubConjunto(conjuntoB));
    }
}
```