



Universidad Nacional Autónoma de México

Facultad de Estudios Superiores Aragón

Estructura de Datos

Tarea 2

Laguna Velasco Elizabeth

Grupo: 1360

Fecha de entrega: 23 de agosto de 2024

```
run:
Tarea 2
-----
Amy ya pertenece al conjunto, no se puede volver a agregar
El conjunto 1 es: [Ander, Elizabeth, Isaac, Amy, Any]
El conjunto 2 es: [Aaron, Paulo, Ander, Lucio, Erick, Esteban]
-----
El conjunto 1 tiene: 5 elementos
El conjunto 2 tiene: 6 elementos
-----
La union de los dos conjuntos es: [Aaron, Paulo, Ander, Elizabeth, Isaac, Lucio, Erick, Amy, Any, Esteban]
La interseccion de los dos conjuntos es: [Ander]
-----
La diferencia del conjunto 1 y el 2 es: [Elizabeth, Isaac, Amy, Any]
La diferencia del conjunto 2 y el 1 es: [Aaron, Paulo, Lucio, Erick, Esteban]
-----
El nuevo conjunto 1 es: [Axel, Ander, Elizabeth, Isaac, Laura, Amy, Any]
El nuevo conjunto 2 es: [Aaron, Ander, Erick, Esteban]
-----
Enrique no pertenece al conjunto 1
Esteban no pertenece al conjunto 1
Enrique no pertenece al conjunto 2
Esteban pertenece al conjunto 2
-----
El conjunto 1 no es igual al conjunto 2
-----
El conjunto 1 no es subconjunto de 2

El conjunto 2 no es subconjunto del 1
-----
BUILD SUCCESSFUL (total time: 0 seconds)
```

```
package tarea2;

import java.util.List;

public class Tarea2 {

    public static void main(String[] args) {
        ConjuntoADT conjunto1 = new ConjuntoADT();
        ConjuntoADT conjunto2 = new ConjuntoADT();

        System.out.println("Tarea 2");
        System.out.println("-----");

        //Para crear el primer conjunto
        List<String> lista1 = List.of("Amy", "Ander", "Elizabeth", "Any", "Isaac", "Amy");
        for (String item: lista1){
            conjunto1.agregar(item);
        }
        System.out.println("El conjunto 1 es: " + conjunto1.toString());

        //Para crear el segundo conjunto
        List<String> lista2 = List.of("Aaron", "Lucio", "Erick", "Paulo", "Esteban", "Ander");
        for (String item: lista2){
            conjunto2.agregar(item);
        }
        System.out.println("El conjunto 2 es: " + conjunto2.toString());

        System.out.println("-----");

        System.out.print("El conjunto 1 tiene: " + conjunto1.longitud() + " elementos");
        System.out.println();
        System.out.print("El conjunto 2 tiene: " + conjunto2.longitud() + " elementos");
        System.out.println();
        System.out.println("-----");
    }
}
```

```

System.out.println("La union de los dos conjuntos es: " + conjunto1.union(conjunto2).toString());
System.out.println("La interseccion de los dos conjuntos es: " + conjunto1.interseccion(conjunto2).toString());

System.out.println("-----");

System.out.println("La diferencia del conjunto 1 y el 2 es: " + conjunto1.diferencia(conjunto2).toString());
System.out.println("La diferencia del conjunto 2 y el 1 es: " + conjunto2.diferencia(conjunto1).toString());

System.out.println("-----");

//Para agregar elementos al conjunto 1
List<String> lista3 = List.of("Laura", "Axel");
for (String item: lista3){
    conjunto1.agregar(item);
}
System.out.println("El nuevo conjunto 1 es: " + conjunto1.toString());

//Para eliminar elementos del conjunto 2
List<String> lista4 = List.of("Lucio", "Paulo");
for (String item: lista4){
    conjunto2.eliminar(item);
}
System.out.println("El nuevo conjunto 2 es: " + conjunto2.toString());

System.out.println("-----");

//Para comprobar si un elemento pertenece al conjunto 1
String nombre1 = "Enrique";
String nombre2 = "Esteban";

if (conjunto1.contiene(nombre1)){
    System.out.println(nombre1 + " pertenece al conjunto 1");
}else{
    System.out.println(nombre1 + " no pertenece al conjunto 1");
}

```



```

if (conjunto1.contiene(nombre2)) {
    System.out.println(nombre2 + " pertenece al conjunto 1");
} else {
    System.out.println(nombre2 + " no pertenece al conjunto 1");
}

//Para comprobar si un elemento pertenece al conjunto 2
if (conjunto2.contiene(nombre1)) {
    System.out.println(nombre1 + " pertenece al conjunto 2");
} else {
    System.out.println(nombre1 + " no pertenece al conjunto 2");
}

if (conjunto2.contiene(nombre2)) {
    System.out.println(nombre2 + " pertenece al conjunto 2");
} else {
    System.out.println(nombre2 + "no pertenece al conjunto 2");
}

System.out.println("-----");

//Para saber si el conjunto 1 es igual al conjunto 2
if (conjunto1.equals(conjunto2)) {
    System.out.println("El conjunto 1 es igual al conjunto 2");
} else {
    System.out.println("El conjunto 1 no es igual al conjunto 2");
}

System.out.println("-----");

//Para saber si un conjunto es subconjunto del otro conjunto

if (conjunto1.esSubConjunto(conjunto2)) {
    System.out.println("El conjunto 1 es subconjunto del 2");
} else {
    System.out.println("El conjunto 1 no es subconjunto de 2");
}

```

```
package tarea2;

import java.util.Set;
import java.util.HashSet;

public class ConjuntoADT<T> {
    private final Set<T> conjunto;
    public ConjuntoADT() {
        conjunto = new HashSet<>();
    }

    public int longitud() {
        return conjunto.size();
    }

    public boolean contiene(T elemento) {
        return conjunto.contains(elemento);
    }

    public void agregar(T elemento) {
        if (conjunto.contains(elemento) == false) {
            conjunto.add(elemento);
        } else {
            System.out.println(elemento + " ya pertenece al conjunto, no se puede volver a agregar");
        }
    }

    public void eliminar(T elemento) {
        if (conjunto.contains(elemento) == true) {
            conjunto.remove(elemento);
        } else {
            System.out.println(elemento + " ya no pertenece al conjunto");
        }
    }
}
```

```
public boolean equals(ConjuntoADT<T> otroConjunto) {  
    return conjunto.equals(otroConjunto.conjunto);  
}
```

```
public boolean esSubConjunto(ConjuntoADT<T> otroConjunto) {  
    return otroConjunto.conjunto.containsAll(conjunto);  
}
```

```
public ConjuntoADT<T> union(ConjuntoADT<T> otroConjunto) {  
    ConjuntoADT u = new ConjuntoADT<>();  
    u.conjunto.addAll(this.conjunto);  
    u.conjunto.addAll(otroConjunto.conjunto);  
    return u;  
}
```

```
public ConjuntoADT<T> interseccion(ConjuntoADT<T> otroConjunto) {  
    ConjuntoADT<T> i = new ConjuntoADT<>();  
    i.conjunto.addAll(this.conjunto);  
    i.conjunto.retainAll(otroConjunto.conjunto);  
    return i;  
}
```

```
public ConjuntoADT<T> diferencia(ConjuntoADT<T> otroConjunto) {  
    ConjuntoADT<T> d = new ConjuntoADT<>();  
    d.conjunto.addAll(this.conjunto);  
    d.conjunto.removeAll(otroConjunto.conjunto);  
    return d;  
}
```

```
@Override  
public String toString() {  
    return conjunto.toString();  
}
```