

```

1 package ico.fes.aragon.unam.conjustosADT;
2
3 import java.util.Collection;
4 import java.util.HashSet;
5 import java.util.Iterator;
6 import java.util.Set;
7
8 public class ConjuntoADT<T> { 19 usages  ▲ Charly2002
9     //Atributos
10     private Set<T> conjunto; 23 usages
11
12     //Constructor
13     public ConjuntoADT() { 5 usages  ▲ Charly2002
14         this.conjunto = new HashSet<>();
15     }
16
17     //Metodos
18     public int Longitud() { no usages  ▲ Charly2002
19         return conjunto.size();
20     }
21
22     public boolean Contiene(T elemento) { 6 usages  ▲ Charly2002
23         return conjunto.contains(elemento);
24     }
25
26     public void Agregar(T elemento) { 10 usages  ▲ Charly2002
27         if(!(this.conjunto.contains(elemento))){...}
28     }
29
30
31
32     public void Eliminar(T elemento) { 4 usages  ▲ Charly2002
33         conjunto.remove(elemento);
34     }
35
36

```

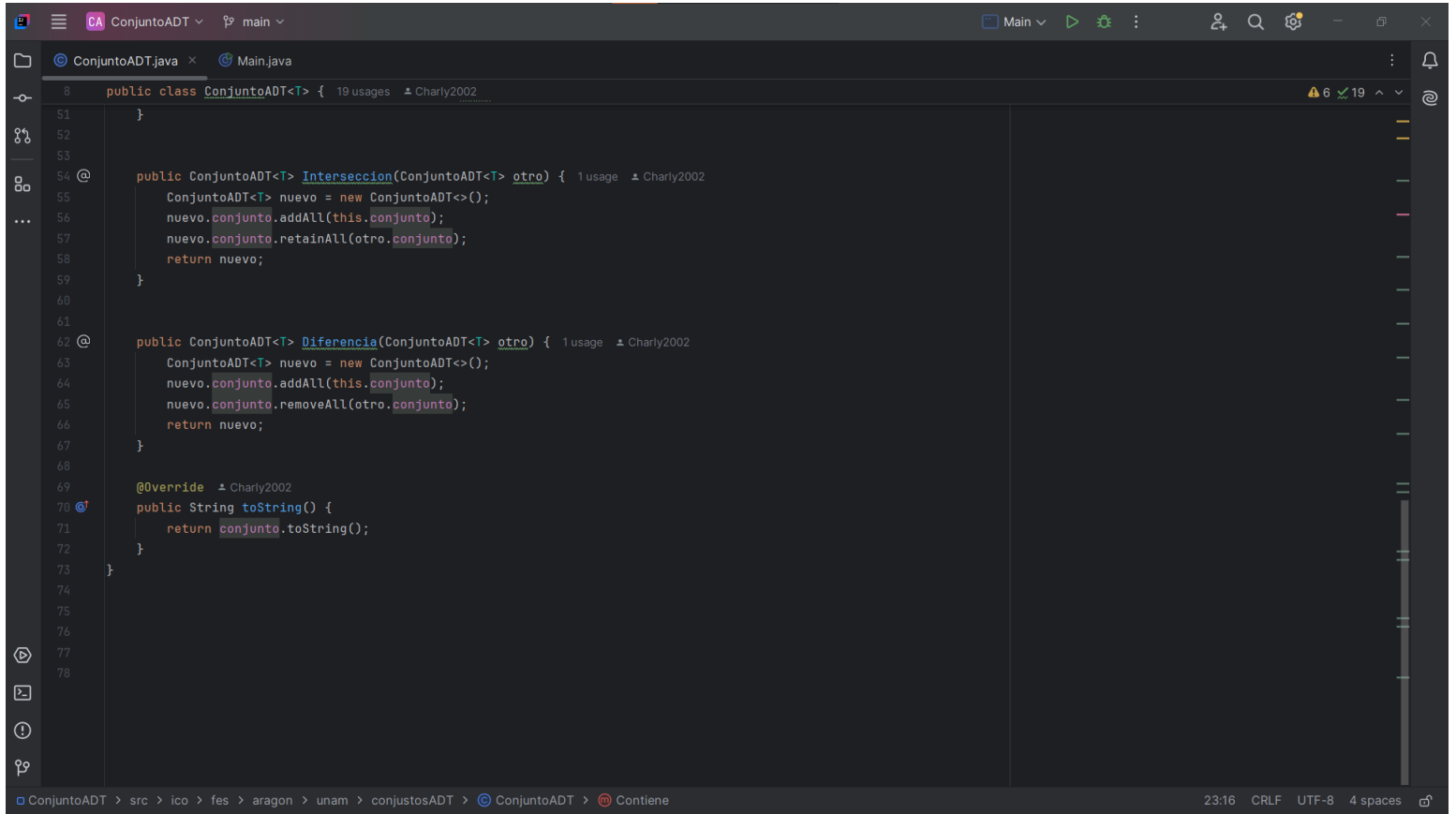
ConjuntoADT > src > ico > fes > aragon > unam > conjustosADT > ConjuntoADT > Contiene

```

36     public class ConjuntoADT<T> { 19 usages  ▲ Charly2002
37
38     public boolean Equals(ConjuntoADT<T> otro){ no usages  ▲ Charly2002
39         return this.conjunto.equals(otro.conjunto);
40     }
41
42     public boolean esSubconjunto(ConjuntoADT<T> otro) { no usages  ▲ Charly2002
43         return otro.conjunto.containsAll(this.conjunto);
44     }
45
46     public ConjuntoADT<T> Union(ConjuntoADT<T> otro) { 1 usage  ▲ Charly2002
47         ConjuntoADT<T> nuevo = new ConjuntoADT<>();
48         nuevo.conjunto.addAll(this.conjunto);
49         nuevo.conjunto.addAll(otro.conjunto);
50         return nuevo;
51     }
52
53
54     public ConjuntoADT<T> Interseccion(ConjuntoADT<T> otro) { 1 usage  ▲ Charly2002
55         ConjuntoADT<T> nuevo = new ConjuntoADT<>();
56         nuevo.conjunto.addAll(this.conjunto);
57         nuevo.conjunto.retainAll(otro.conjunto);
58         return nuevo;
59     }
60
61
62     public ConjuntoADT<T> Diferencia(ConjuntoADT<T> otro) { 1 usage  ▲ Charly2002
63         ConjuntoADT<T> nuevo = new ConjuntoADT<>();
64         nuevo.conjunto.addAll(this.conjunto);
65         nuevo.conjunto.removeAll(otro.conjunto);
66         return nuevo;
67     }
68

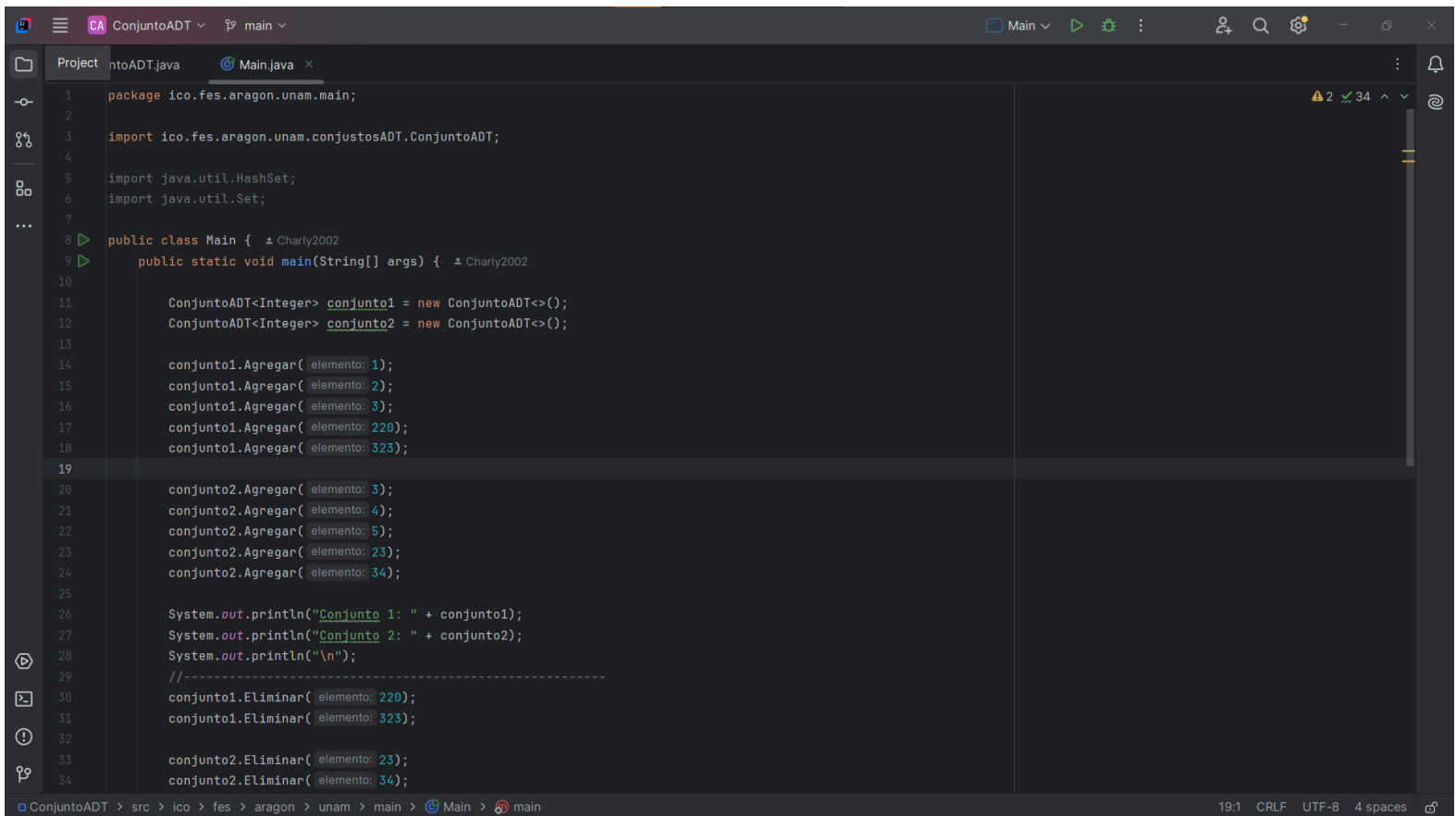
```

ConjuntoADT > src > ico > fes > aragon > unam > conjustosADT > ConjuntoADT > Contiene



```
8 public class ConjuntoADT<T> { 19 usages Charly2002
51 }
52
53
54 @
55 public ConjuntoADT<T> Interseccion(ConjuntoADT<T> otro) { 1 usage Charly2002
56     ConjuntoADT<T> nuevo = new ConjuntoADT<>();
57     nuevo.conjunto.addAll(this.conjunto);
58     nuevo.conjunto.retainAll(otro.conjunto);
59     return nuevo;
60 }
61
62 @
63 public ConjuntoADT<T> Diferencia(ConjuntoADT<T> otro) { 1 usage Charly2002
64     ConjuntoADT<T> nuevo = new ConjuntoADT<>();
65     nuevo.conjunto.addAll(this.conjunto);
66     nuevo.conjunto.removeAll(otro.conjunto);
67     return nuevo;
68 }
69
70 @Override Charly2002
71 public String toString() {
72     return conjunto.toString();
73 }
74
75
76
77
78 }
```

ConjuntoADT > src > ico > fes > aragon > unam > conjuntosADT > ConjuntoADT > Contiene 23:16 CRLF UTF-8 4 spaces



```
1 package ico.fes.aragon.unam.main;
2
3 import ico.fes.aragon.unam.conjuntosADT.ConjuntoADT;
4
5 import java.util.HashSet;
6 import java.util.Set;
7
8 public class Main { Charly2002
9     public static void main(String[] args) { Charly2002
10
11         ConjuntoADT<Integer> conjunto1 = new ConjuntoADT<>();
12         ConjuntoADT<Integer> conjunto2 = new ConjuntoADT<>();
13
14         conjunto1.Agregar( elemento: 1);
15         conjunto1.Agregar( elemento: 2);
16         conjunto1.Agregar( elemento: 3);
17         conjunto1.Agregar( elemento: 220);
18         conjunto1.Agregar( elemento: 323);
19
20         conjunto2.Agregar( elemento: 3);
21         conjunto2.Agregar( elemento: 4);
22         conjunto2.Agregar( elemento: 5);
23         conjunto2.Agregar( elemento: 23);
24         conjunto2.Agregar( elemento: 34);
25
26         System.out.println("Conjunto 1: " + conjunto1);
27         System.out.println("Conjunto 2: " + conjunto2);
28         System.out.println("\n");
29         //-----
30         conjunto1.Eliminar( elemento: 220);
31         conjunto1.Eliminar( elemento: 323);
32
33         conjunto2.Eliminar( elemento: 23);
34         conjunto2.Eliminar( elemento: 34);
35     }
36 }
```

ConjuntoADT > src > ico > fes > aragon > unam > main > Main > main 19:1 CRLF UTF-8 4 spaces

```
ConjuntoADT.java Main.java x
8 public class Main { // Charly2002
9     public static void main(String[] args) { // Charly2002
34         conjunto2.Eliminar( elemento: 34);
35
36         System.out.println("Conjuntos depues de eliminar elementos: ");
37         System.out.println("Conjunto 1: " + conjunto1);
38         System.out.println("Conjunto 2: " + conjunto2);
39         System.out.println("\n");
40         //-----
41         System.out.println("Comprobar si los elementos pertenecen a los conjuntos: ");
42         System.out.println("El conjunto1 tiene el elemento 20: "+conjunto1.Contiene( elemento: 20));
43         System.out.println("El conjunto1 tiene el elemento 2: "+conjunto1.Contiene( elemento: 2));
44         System.out.println("El conjunto1 tiene el elemento 345: "+conjunto1.Contiene( elemento: 345));
45
46         System.out.println("El conjunto2 tiene el elemento 5: "+conjunto2.Contiene( elemento: 5));
47         System.out.println("El conjunto2 tiene el elemento 578: "+conjunto2.Contiene( elemento: 578));
48         System.out.println("El conjunto2 tiene el elemento 4: "+conjunto2.Contiene( elemento: 4));
49
50         System.out.println("\n");
51         //-----
52         System.out.println("Operaciones de conjuntos: ");
53         System.out.println("Unión: " + conjunto1.Union(conjunto2));
54         System.out.println("Intersección: " + conjunto1.Interseccion(conjunto2));
55         System.out.println("Diferencia: " + conjunto1.Diferencia(conjunto2));
56
57     }
58 }
59 }
60
```

ConjuntoADT > src > ico > fes > aragon > unam > main > Main > main Saving 'ConjuntoADT' settings 19:1 CRLF UTF-8 4 spaces

```
Project ntoADT.java Main.java x
Run Main x
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.0.1\lib\idea_rt.jar=53393:C:\Program Files\JetBrains\I
Conjunto 1: [1, 2, 3, 323, 220]
Conjunto 2: [34, 3, 4, 5, 23]

Conjuntos depues de eliminar elementos:
Conjunto 1: [1, 2, 3]
Conjunto 2: [3, 4, 5]

Comprobar si los elementos pertenecen a los conjuntos:
El conjunto1 tiene el elemento 20: false
El conjunto1 tiene el elemento 2: true
El conjunto1 tiene el elemento 345: false
El conjunto2 tiene el elemento 5: true
El conjunto2 tiene el elemento 578: false
El conjunto2 tiene el elemento 4: true

Operaciones de conjuntos:
Unión: [1, 2, 3, 4, 5]
Intersección: [3]
Diferencia: [1, 2]

Process finished with exit code 0

ConjuntoADT > src > ico > fes > aragon > unam > main > Main > main 19:1 CRLF UTF-8 4 spaces
```

```

package ico.fes.aragon.unam.conjustosADT;

import java.util.Collection;
import java.util.HashSet;
import java.util.Iterator;
import java.util.Set;

public class ConjuntoADT<T> {
    //Atributos
    private Set<T> conjunto;

    //Constructor
    public ConjuntoADT() {
        this.conjunto = new HashSet<>();
    }

    //Metodos
    public int Longitud() {
        return conjunto.size();
    }

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

    public void Agregar(T elemento) {
        if(!(this.conjunto.contains(elemento))){
            conjunto.add(elemento);
        }
    }

    public void Eliminar(T elemento) {
        conjunto.remove(elemento);
    }

    public boolean Equals(ConjuntoADT<T> otro){
        return this.conjunto.equals(otro.conjunto);
    }

    public boolean esSubconjunto(ConjuntoADT<T> otro) {
        return otro.conjunto.containsAll(this.conjunto);
    }

    public ConjuntoADT<T> Union(ConjuntoADT<T> otro) {
        ConjuntoADT<T> nuevo = new ConjuntoADT<>();
        nuevo.conjunto.addAll(this.conjunto);
        nuevo.conjunto.addAll(otro.conjunto);
    }
}

```

```

        return nuevo;
    }

    public ConjuntoADT<T> Interseccion(ConjuntoADT<T> otro) {
        ConjuntoADT<T> nuevo = new ConjuntoADT<>();
        nuevo.conjunto.addAll(this.conjunto);
        nuevo.conjunto.retainAll(otro.conjunto);
        return nuevo;
    }

    public ConjuntoADT<T> Diferencia(ConjuntoADT<T> otro) {
        ConjuntoADT<T> nuevo = new ConjuntoADT<>();
        nuevo.conjunto.addAll(this.conjunto);
        nuevo.conjunto.removeAll(otro.conjunto);
        return nuevo;
    }

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

public class Main {
    public static void main(String[] args) {

        ConjuntoADT<Integer> conjunto1 = new ConjuntoADT<>();
        ConjuntoADT<Integer> conjunto2 = new ConjuntoADT<>();

        conjunto1.Agregar(1);
        conjunto1.Agregar(2);
        conjunto1.Agregar(3);
        conjunto1.Agregar(220);
        conjunto1.Agregar(323);

        conjunto2.Agregar(3);
        conjunto2.Agregar(4);
        conjunto2.Agregar(5);
        conjunto2.Agregar(23);
        conjunto2.Agregar(34);
    }
}

```

```

System.out.println("Conjunto 1: " + conjunto1);
System.out.println("Conjunto 2: " + conjunto2);
System.out.println("\n");
//-----
conjunto1.Eliminar(220);
conjunto1.Eliminar(323);

conjunto2.Eliminar(23);
conjunto2.Eliminar(34);

System.out.println("Conjuntos depues de eliminar elementos: ");
System.out.println("Conjunto 1: " + conjunto1);
System.out.println("Conjunto 2: " + conjunto2);
System.out.println("\n");
//-----
System.out.println("Comprobar si los elementos pertenesen a los conjuntos: ");
System.out.println("El conjunto1 tiene el elemento 20: "+conjunto1.Contiene(20));
System.out.println("El conjunto1 tiene el elemento 2: "+conjunto1.Contiene(2));
System.out.println("El conjunto1 tiene el elemento 345: "+conjunto1.Contiene(345));

System.out.println("El conjunto2 tiene el elemento 5: "+conjunto2.Contiene(5));
System.out.println("El conjunto2 tiene el elemento 578: "+conjunto2.Contiene(578));
System.out.println("El conjunto2 tiene el elemento 4: "+conjunto2.Contiene(4));

System.out.println("\n");
//-----
System.out.println("Operaciones de conjuntos: ");
System.out.println("Unión: " + conjunto1.Union(conjunto2));
System.out.println("Intersección: " + conjunto1.Interseccion(conjunto2));
System.out.println("Diferencia: " + conjunto1.Diferencia(conjunto2));

}
}

```

