Código

Conjunto a:

```
package ico.fes.aragon.unam.clases;
import java.util.HashSet;
import java.util.Iterator;
import java.util.Set;
public class ConjuntoA<T> implements ConjuntoADT<T> {
  private Set<T> conjunto;
  private int tamanio;
public ConjuntoA(){
  this.conjunto = new HashSet<>();
}
  public int longitud() {
    return this.tamanio;
  }
  @Override
  public void agregar(T elemento) {
     if (!conjunto.contains(elemento)) {
       conjunto.remove(elemento);
       conjunto.add(elemento);
    }
  }
  @Override
  public T eliminar(T elemento) {
     if (conjunto.contains(elemento)) {
       conjunto.remove(elemento);
       return elemento;
    return null;
  }
  @Override
  public boolean contiene(T elemento) {
```

```
return conjunto.contains(elemento);
}
@Override
public boolean estaVacio() {
  return conjunto.isEmpty();
}
@Override
public int tamanio() {
  return conjunto.size();
}
@Override
public ConjuntoADT<T> union(ConjuntoADT<T> otroConjunto) {
  ConjuntoA<T> union = new ConjuntoA<>();
  for (T elemento : conjunto) {
     if (!otroConjunto.contiene(elemento)) {
       union.agregar(elemento);
    }
  }
  return union;
}
@Override
public ConjuntoADT interseccion(ConjuntoADT<T> otroConjunto) {
  ConjuntoA<T> interseccion = new ConjuntoA<>();
  for (T elemento : conjunto) {
    if (!otroConjunto.contiene(elemento)) {
       interseccion.agregar(elemento);
  }
  return interseccion;
}
@Override
public ConjuntoADT diferencia(ConjuntoADT<T> otroConjunto) {
  ConjuntoA<T> diferencia = new ConjuntoA<>();
  for (T elemento : conjunto) {
     if (!otroConjunto.contiene(elemento)) {
       diferencia.agregar(elemento);
    }
  return diferencia;
}
```

ConjuntoADT

```
package ico.fes.aragon.unam.clases;
public interface ConjuntoADT<T> extends Iterable<T> {
 public void agregar(T elemento);
  T eliminar(T valor);
 public boolean contiene(T elemento);
 public boolean estaVacio();
 public int tamanio();
 public ConjuntoADT<T>union(ConjuntoADT<T> otroConjunto);
 public ConjuntoADT<T>interseccion(ConjuntoADT<T> otroConjunto);
 public ConjuntoADT<T>diferencia(ConjuntoADT<T> otroConjunto);
  void eliminar();
  /*String[] Frutas= new String[5];
  Frutas[0]= "Durazno";
  Frutas[1]= "Mango";
  Frutas[2]= "Manzana";
  Frutas[3]= "Uva";
  Frutas[4]= "Fresa";
  ArrayList fruitList = new ArrayList();
    System.out.println(1);
    fruitList.add("Durazno");
    fruitList.add("Mango");
    fruitList.add("Manzana");
    fruitList.add("Uva");
```

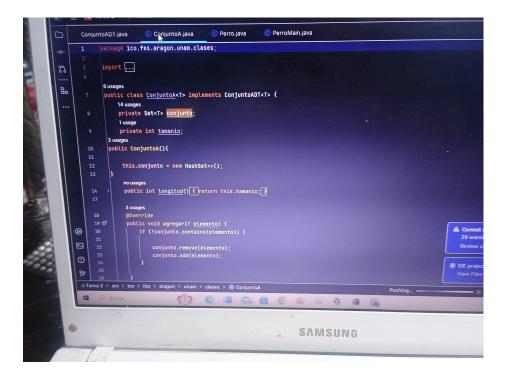
```
fruitList.add("Fresa");*/
}
Perro:
package ico.fes.aragon.unam.clases;
import java.util.Objects;
public class Perro {
  private String raza;
  private int edad;
  private String color;
  public Perro() {
  }
  public Perro(String raza, int edad, String color) {
     this.raza = raza;
     this.edad = edad;
     this.color = color;
  }
  public String getRaza() {
     return raza;
  }
  public void setRaza(String raza) {
     this.raza = raza;
  }
  public int getEdad() {
     return edad;
  }
  public void setEdad(int edad) {
     this.edad = edad;
```

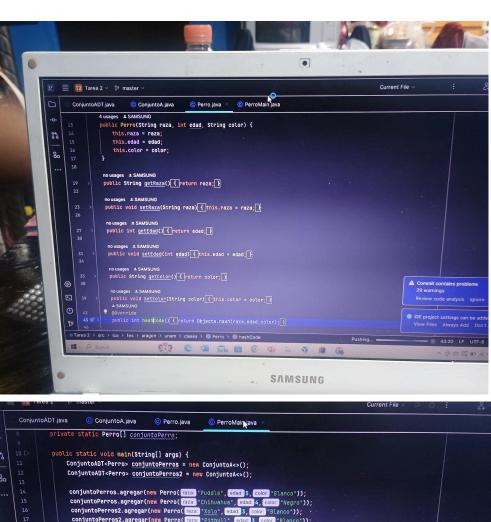
```
}
  public String getColor() {
     return color;
  }
  public void setColor(String color) {
    this.color = color;
  @Override
  public int hashCode(){
     return Objects.hash(raza,edad,color);
  }
  @Override
  public String toString() {
     return "Perro{" +
          "raza="" + raza + '\" +
          ", edad=" + edad +
          ", color="" + color + '\" +
          '}';
  }
  @Override
  public boolean equals(Object obj) {
    if (this ==obj);
     return true;
Main
package ico.fes.aragon.unam.main;
import ico.fes.aragon.unam.clases.ConjuntoA;
import ico.fes.aragon.unam.clases.ConjuntoADT;
import ico.fes.aragon.unam.clases.Perro;
```

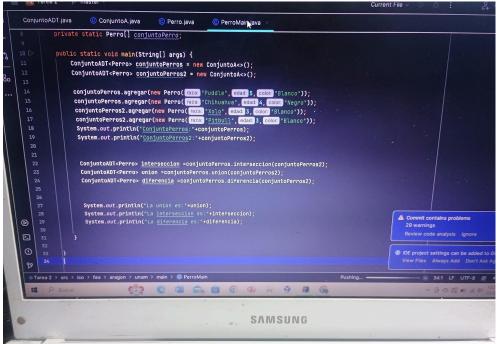
}

```
public class PerroMain {
  private static Perro[] conjuntoPerro;
  public static void main(String[] args) {
    ConjuntoADT<Perro> conjuntoPerros = new ConjuntoA<>();
    ConjuntoADT<Perro> conjuntoPerros2 = new ConjuntoA<>();
    conjuntoPerros.agregar(new Perro("Puddle",5,"Blanco"));
    conjuntoPerros.agregar(new Perro("Chihuahua",4,"Negro"));
    conjuntoPerros2.agregar(new Perro("Xolo",3,"Blanco"));
    conjuntoPerros2.agregar(new Perro("Pitbull",3,"Blanco"));
    System.out.println("ConjuntoPerros:"+conjuntoPerros);
    System.out.println("ConjuntoPerros2:"+conjuntoPerros2);
     ConjuntoADT<Perro> interseccion
=conjuntoPerros.interseccion(conjuntoPerros2);
    ConjuntoADT<Perro> union
=conjuntoPerros.union(conjuntoPerros2);
    ConjuntoADT<Perro> diferencia
=conjuntoPerros.diferencia(conjuntoPerros2);
    System.out.println("La union es:"+union);
    System.out.println("La interseccion es:"+interseccion);
     System.out.println("La diferencia es:"+diferencia);
  }
}
```

Capturas de pantalla del codigo con la corrida.







1-7