

4.1. Practicar a reestruturación de código Java en NetBeans. Debes indicar en texto a entrada de menú ou menú contextual empregada, unha captura de pantalla das opcións da acción e outra da situación despois da reestruturación.

Realizar as seguintes reestruturacións sobre o proxecto Circulo que dispón dunha clase de probas unitarias JUnit. Recorda que despois de realizar a reestruturación de cada exercicio, débese comprobar coas probas unitarias que a clase segue pasando as probas.

Renomear a clase Circulo por Circulito

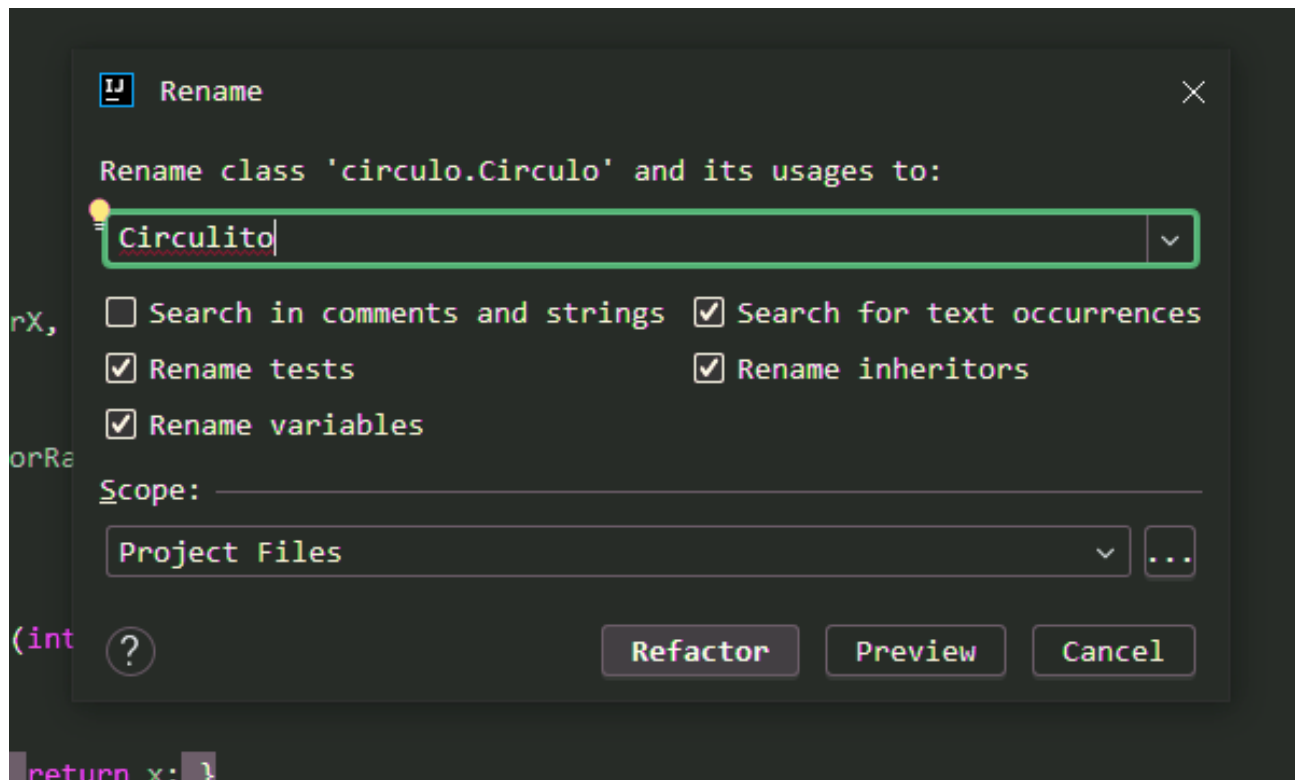


Figura 1

Renomear o método ObterArea por ObterAreaCirculo

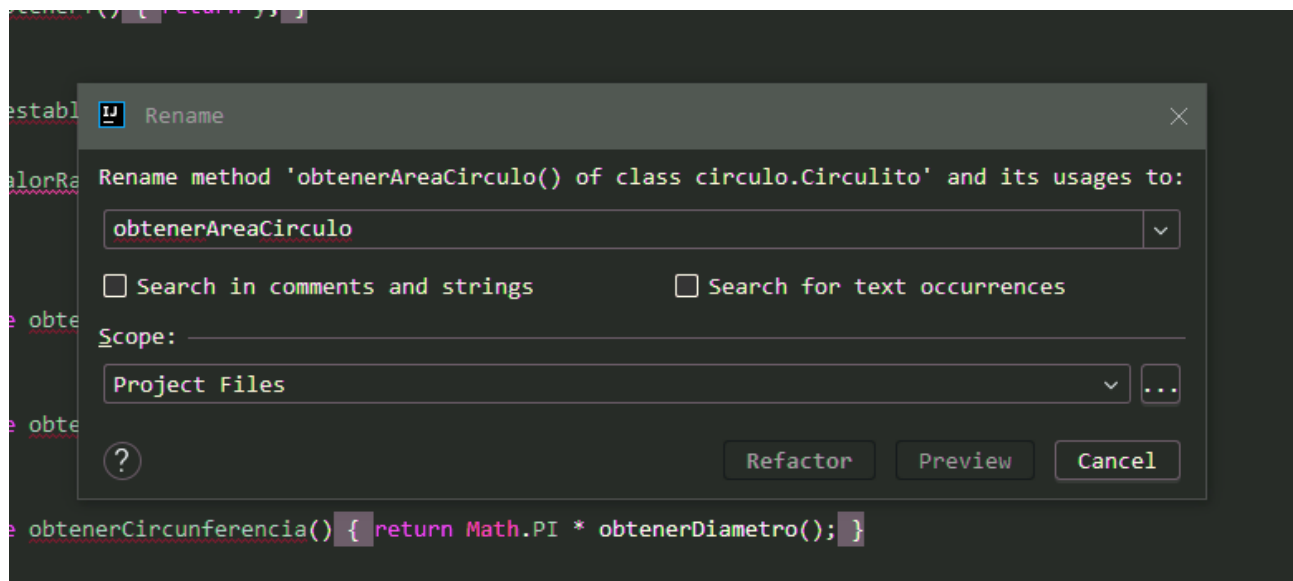


Figura 2

Renomear os campo x e y por coordenadaX e coordenadaY.

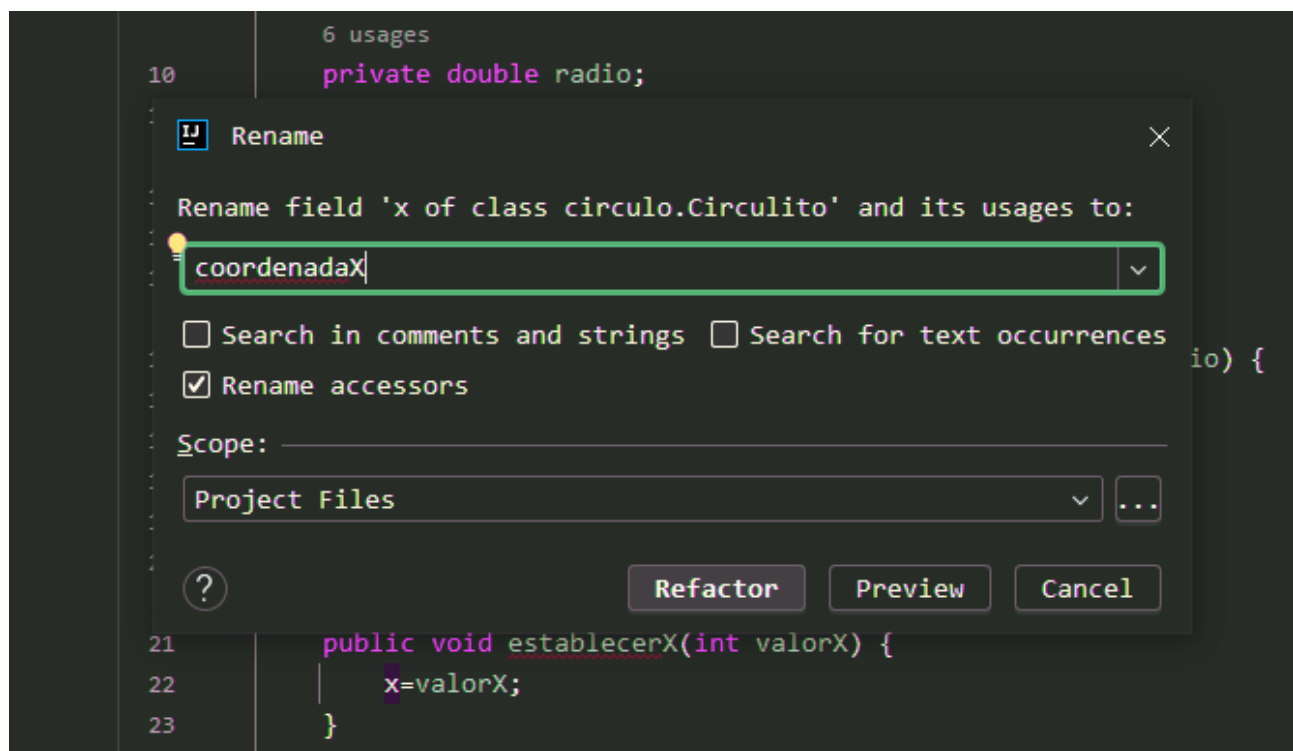


Figura 3

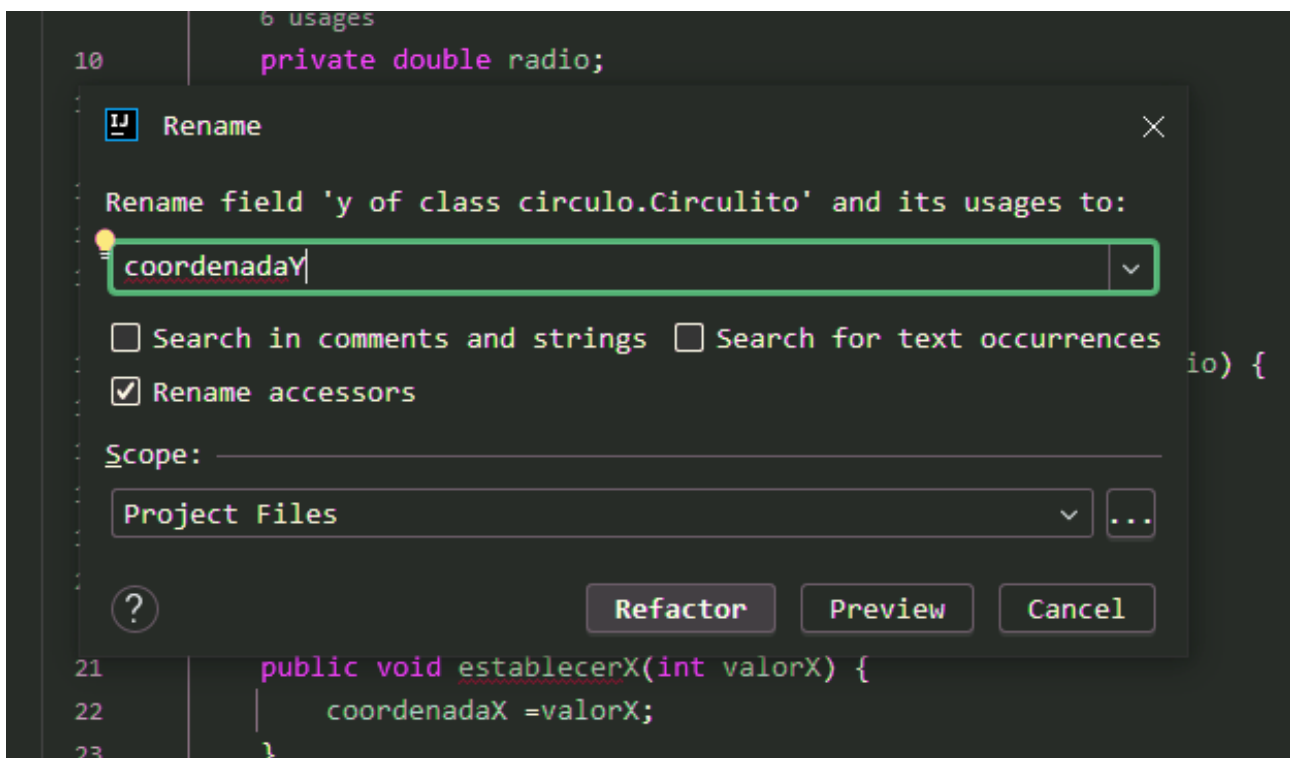


Figura 4

Introducir constante LIMITERADIO de tipo double co valor 0.0

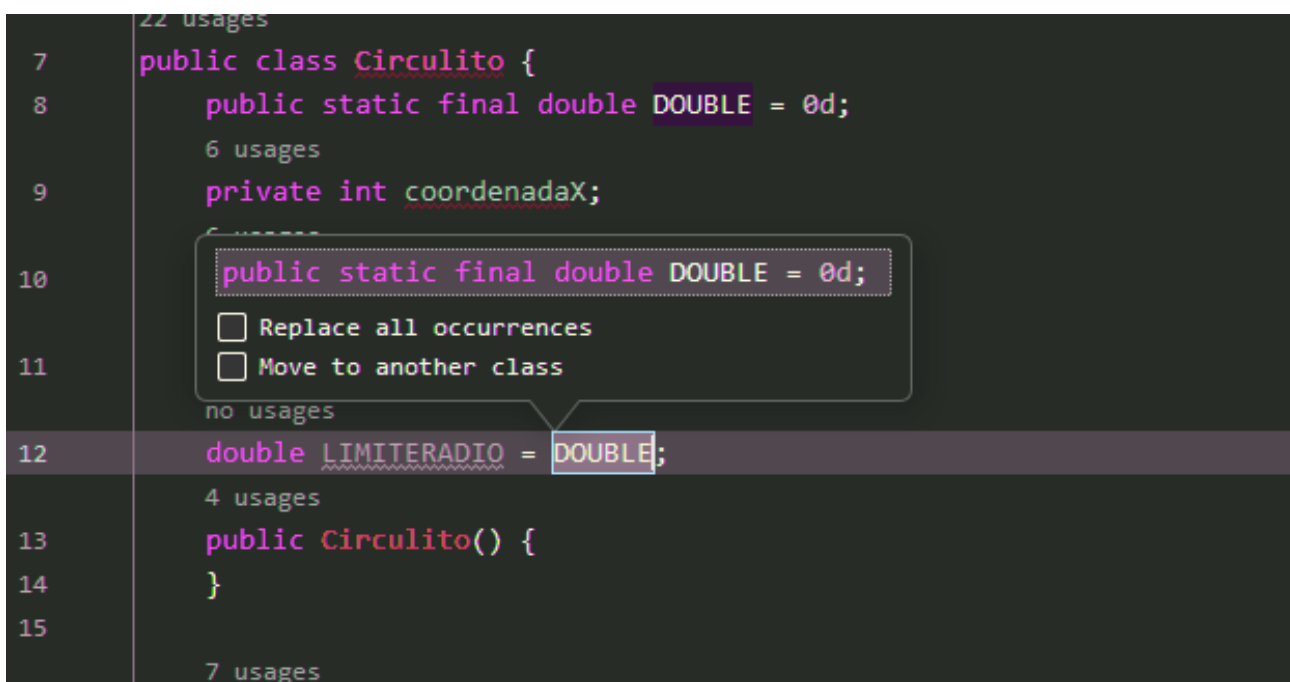


Figura 5

Cambiar parámetros do método trasladarCentro para que teña dous parámetros trasladarX e trasladarY de tipo int. Facer os cambios necesarios para que o código do método permita engadir á coordenada x o valor de trasladarX e o engadir á coordenada y o valor de trasladarY.

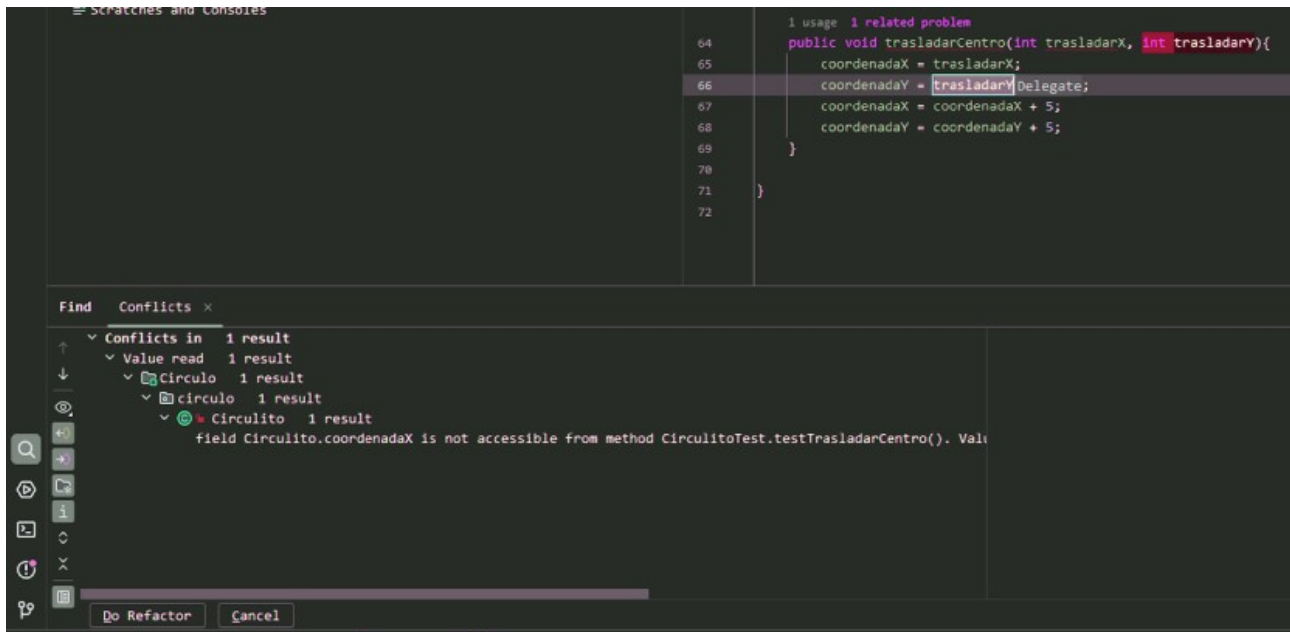


Figura 6

Encapsular os tres atributos da classe: coordenadaX coordenadaY, radio.

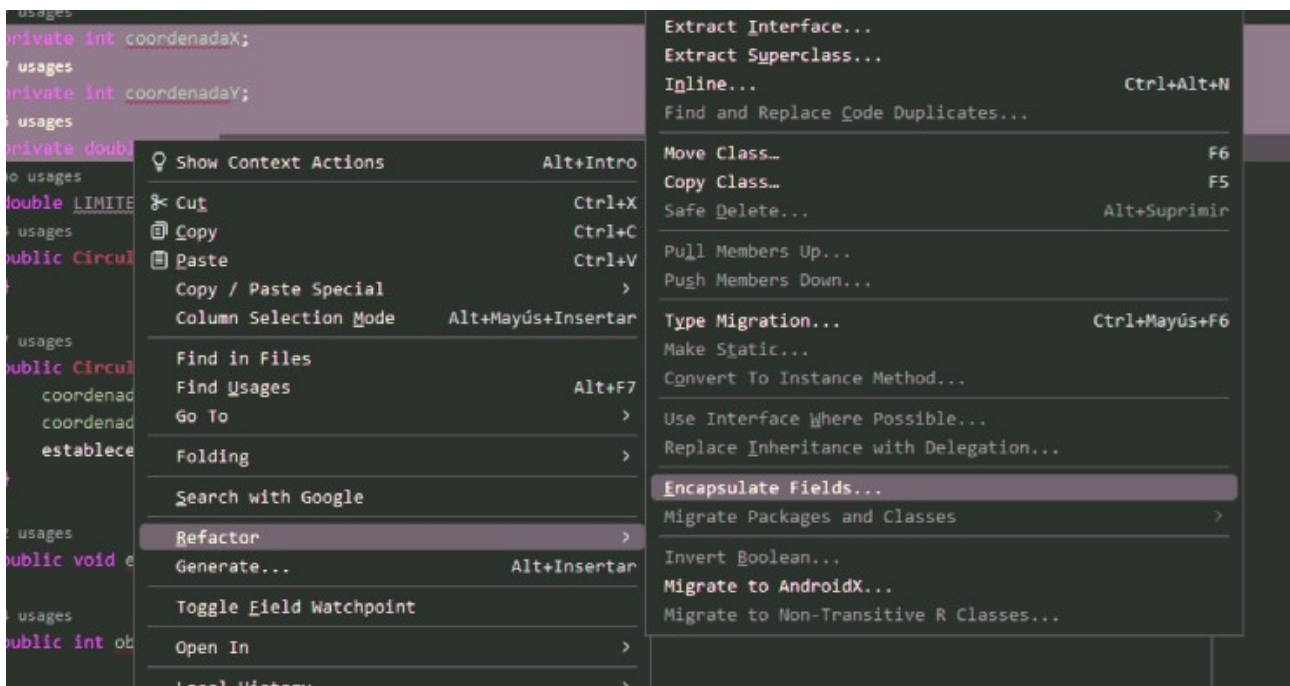


Figura 7

Eliminar de forma segura os métodos obtenerX, obtenerY, obtenerRadio, establecerX, establecerY e establecerRadio que agora son innecesarios facendo os cambios necesarios no código da clase, da clase Main e das probas para que sexan substituídos polos correspondentes métodos tipo get e set creados.

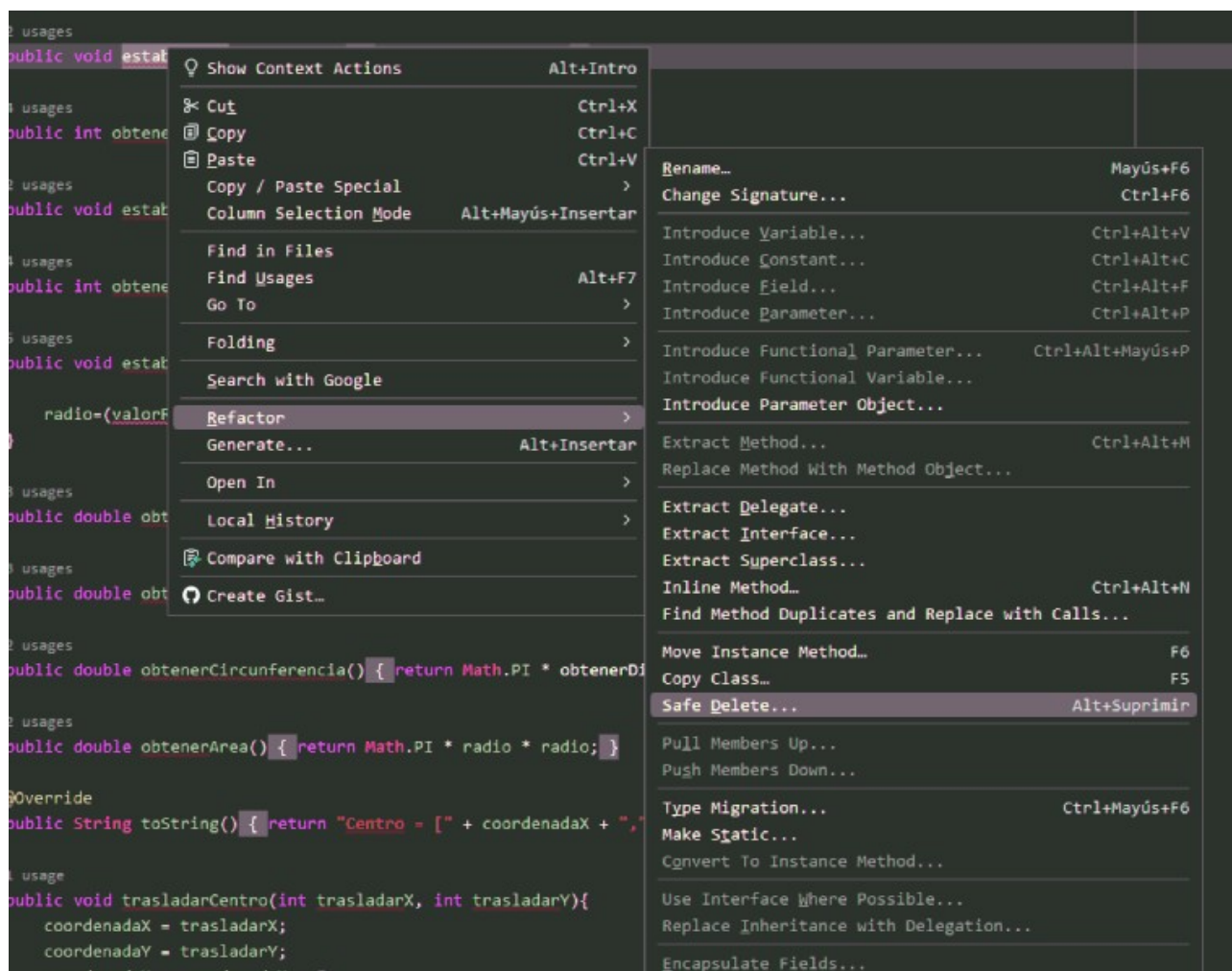
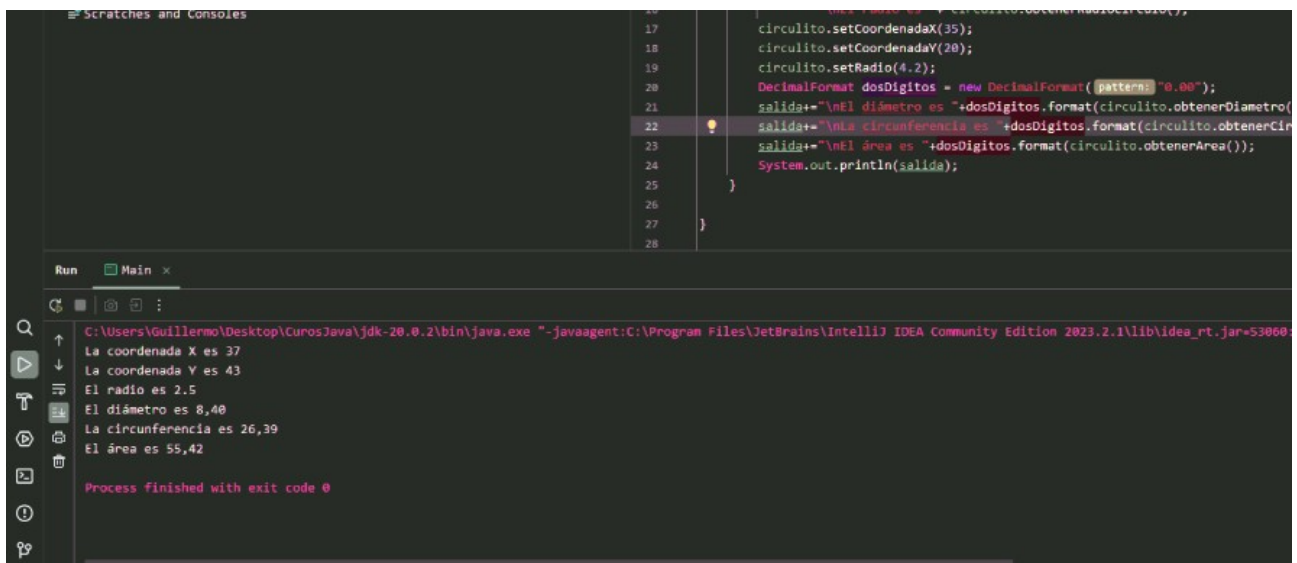


Figura 8

Ao rematar comproba que as probas seguen funcionando (achega captura de pantalla co resultado)



The screenshot shows the IntelliJ IDEA interface. The top pane displays a Java class named `Circulo` with methods for setting coordinates, radius, and calculating diameter, circumference, and area. The bottom pane shows the output of the program, which matches the expected results.

```
20:    (new Radio(4.2) y el circulo).obtenerRadio().getRadio();
21:    circulo.setCoordenadaX(35);
22:    circulo.setCoordenadaY(20);
23:    circulo.setRadio(4.2);
24:    DecimalFormat dosDigitos = new DecimalFormat("0.00");
25:    salida+="\nEl diámetro es "+dosDigitos.format(circulo.obtenerDiametro());
26:    salida+="\nLa circunferencia es "+dosDigitos.format(circulo.obtenerCircunferencia());
27:    salida+="\nEl área es "+dosDigitos.format(circulo.obtenerArea());
28:    System.out.println(salida);
29: }
```

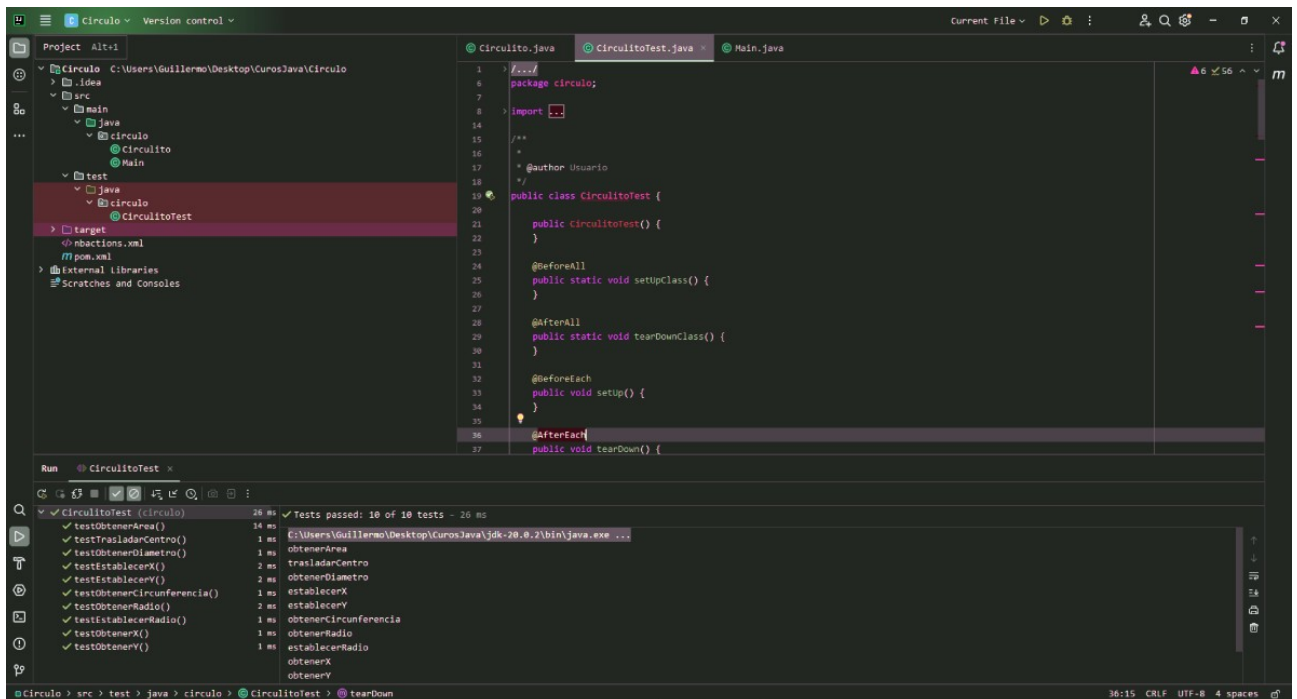
Run Main x

C:\Users\Guillermo\Desktop\CurosJava\jdk-20.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.2.1\lib\idea_rt.jar=53060:..."

La coordenada X es 37
La coordenada Y es 43
El radio es 2.5
El diámetro es 8,40
La circunferencia es 26,39
El área es 55,42

Process finished with exit code 0

Figura 9



The screenshot shows the IntelliJ IDEA interface with a project named `Circulo`. The `CirculoTest` class is open, showing JUnit 5 test methods. The bottom pane displays the test results, indicating that all 10 tests passed.

```
1: //src
2: package circulo;
3:
4: import org.junit.jupiter.api.*;
5:
6: /**
7:  * @author Usuario
8:  */
9: public class CirculoTest {
10:
11:     public CirculoTest() {
12:     }
13:
14:     @BeforeAll
15:     public static void setUpClass() {
16:     }
17:
18:     @AfterAll
19:     public static void tearDownClass() {
20:     }
21:
22:     @BeforeEach
23:     public void setUp() {
24:     }
25:
26:     @AfterEach
27:     public void tearDown() {
28:     }
29: }
```

Run CirculoTest x

✓ CirculoTest (circulo) 26 ms ✓ Tests passed: 10 of 10 tests - 26 ms

- ✓ testObtenerArea() 14 ms
- ✓ testTrasladarCentro() 1 ms
- ✓ testObtenerDiametro() 1 ms
- ✓ testEstablecerX() 2 ms
- ✓ testEstablecerY() 2 ms
- ✓ testObtenerCircunferencia() 1 ms
- ✓ testObtenerRadio() 2 ms
- ✓ testEstablecerRadio() 1 ms
- ✓ testObtenerX() 1 ms
- ✓ testObtenerY() 1 ms

36:15 CRLF UTF-8 4 spaces

Figura 10

Realizar as seguintes reestruturacións sobre o proxecto Alumno, modificando se é o caso, os tests implementados.

Move o atributo ‘telefono’ da clase Alumno a súa superclase Persona.

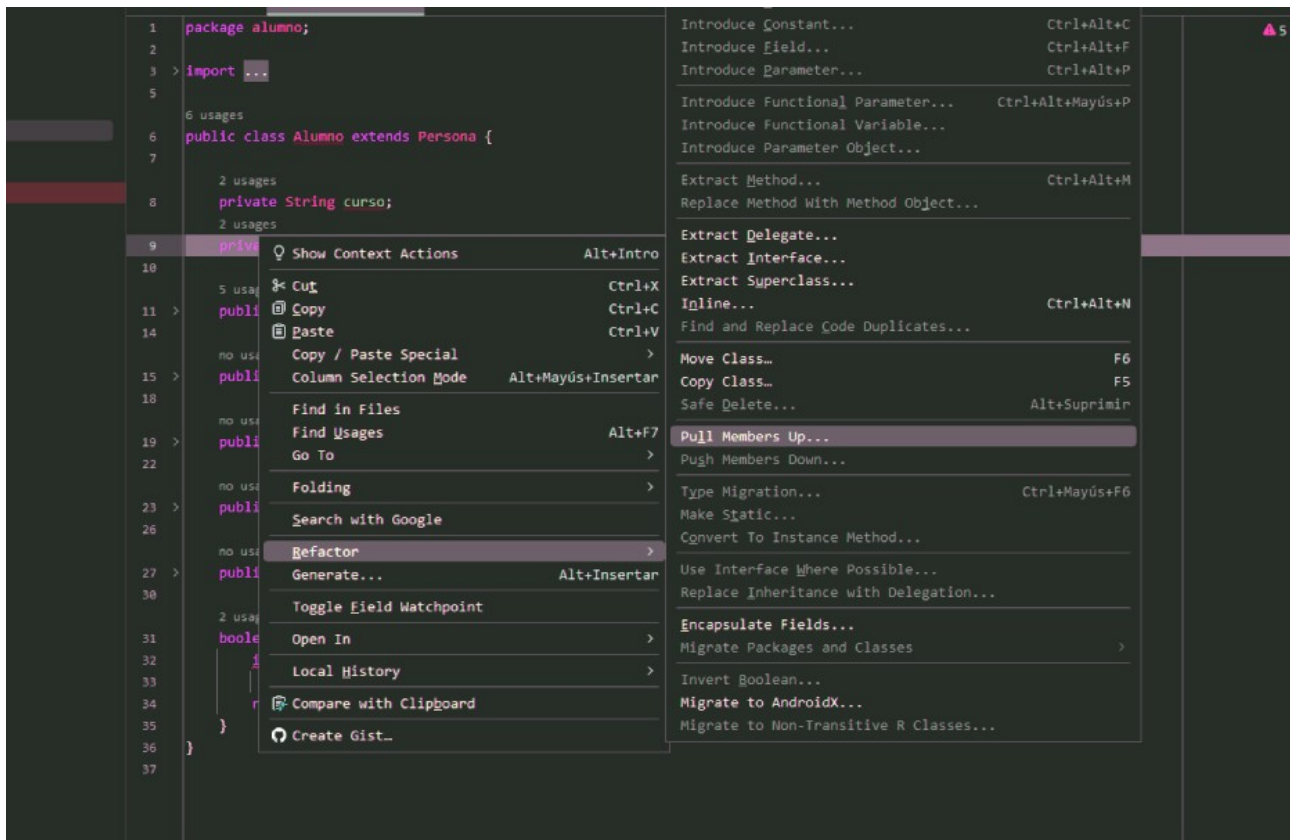


Figura 11

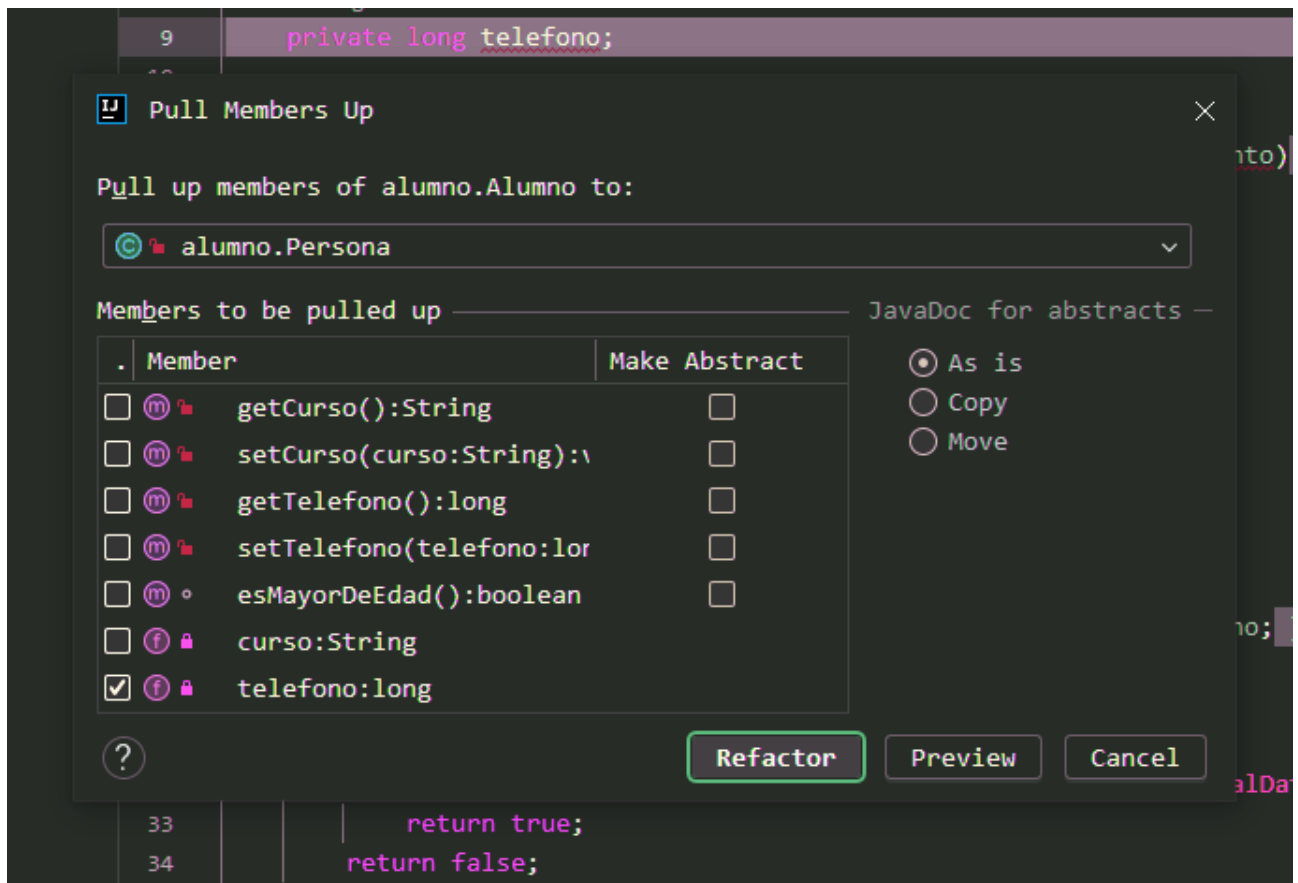


Figura 12

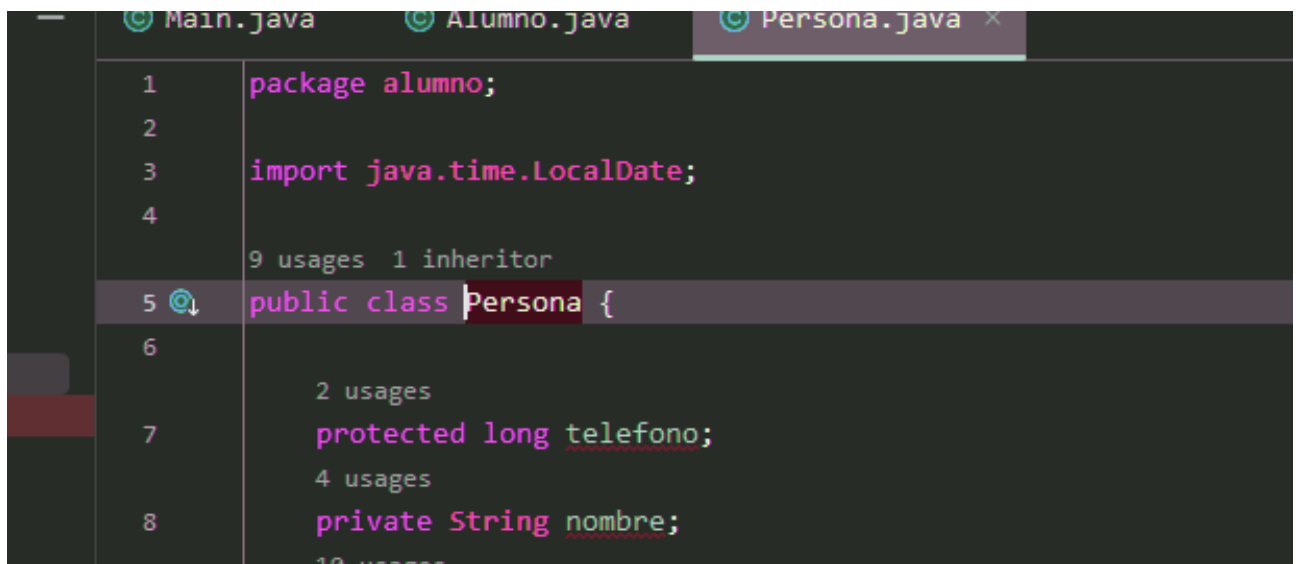


Figura 13

Move o método 'esMayorDeEdad' da classe Alumno a sua superclase Persona.

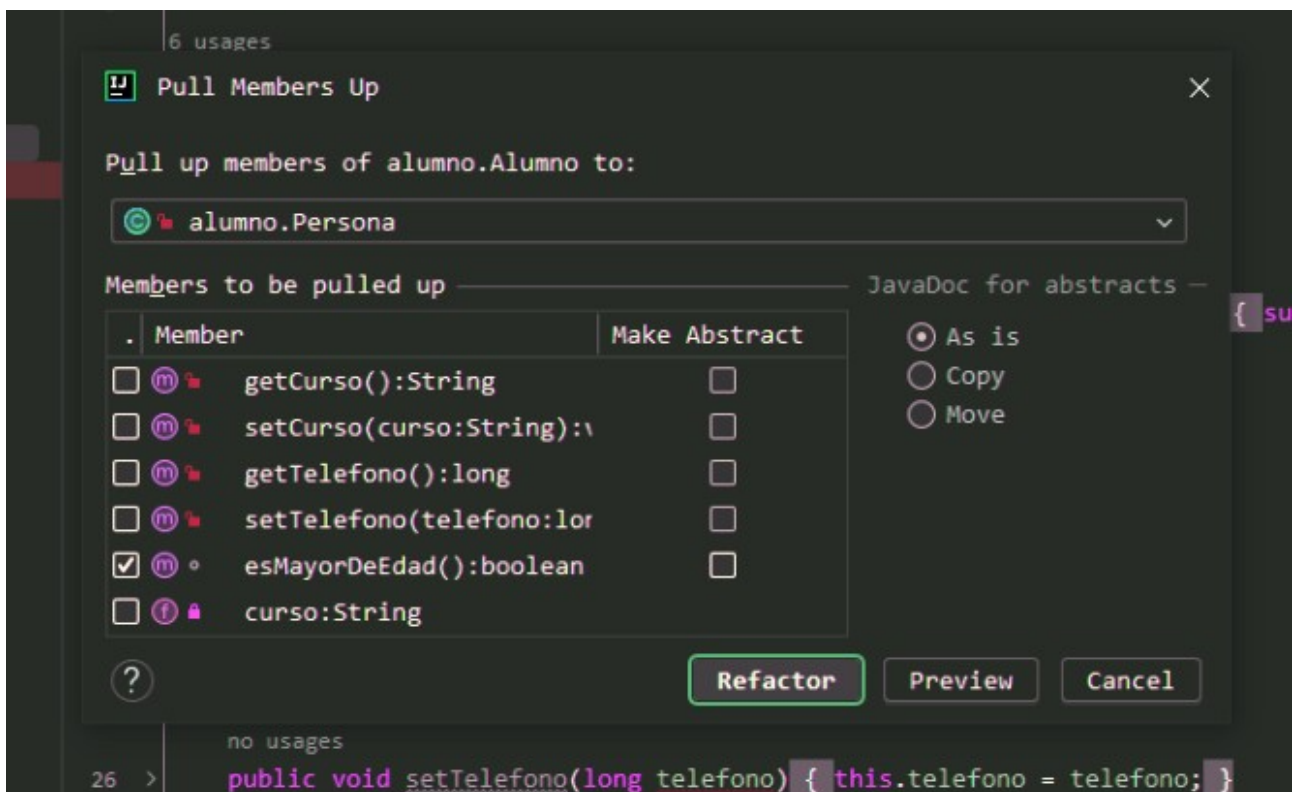


Figura 14

```

Main.java  Alumno.java  Persona.java x
54         String numero = this.dni.substring(0,this.dni.length()-1);
55
56         char letraCorrecta = ' ';
57         try {
58             num = Integer.parseInt(numero);
59         } catch (NumberFormatException numberFormatException) { numeroValido = false;}
60         String letras = "TRWAGYFPDXBNJZSQVHLCKE";
61         if (numeroValido)
62             letraCorrecta= letras.charAt(num%23);
63
64         if (letraDNI == letraCorrecta) return true;
65         else return false;
66     }
67
68     2 usages
69     public void asignarLetraDNI () {
70         int num=0;
71         boolean numeroValido = true;
72         String numero= this.dni;
73
74         char letraCorrecta = ' ';
75         try {
76             num = Integer.parseInt(numero);
77         } catch (NumberFormatException numberFormatException) { numeroValido = false;}
78         String letras = "TRWAGYFPDXBNJZSQVHLCKE";
79         if (numeroValido) {
80             letraCorrecta=letras.charAt(num%23);
81         }
82         if (letraCorrecta != ' ') this.dni+=letraCorrecta;
83     }
84     2 usages
85     boolean esMayorDeEdad () {
86         if (ChronoUnit.YEARS.between(this.getFechaNacimiento(), LocalDate.now()) >= 18)
87             return true;
88         return false;
89     }
90

```

Figura 15

Na classe 'Persona' substituir o número 18 por unha constante estática.

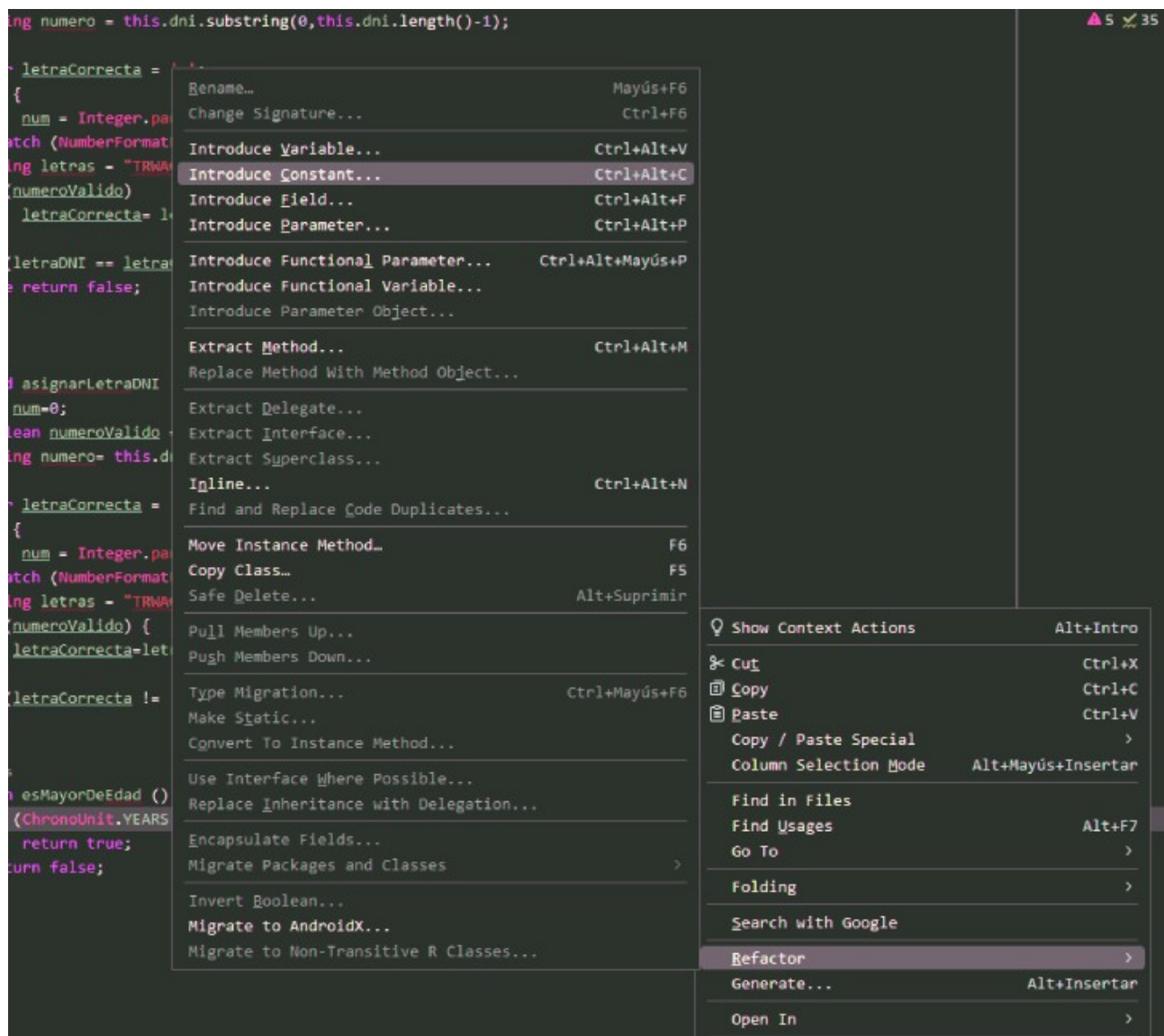


Figura 16

```

1 package alumno;
2
3 import java.time.LocalDate;
4 import java.time.temporal.ChronoUnit;
5
6 public class Persona {
7
8     public static final int INT = 18;
9
10    protected long telefono;

```

Figura 17

Nos métodos comprobarDNI e asignarLetraDNI repítase o código. Crea un método privado con ese código que sexa invocado desde os outro dous métodos.

```

61 String letras = "TRMAGMYFPDXBNJZSQVHLCKE";
62 if (numeroValido)
63     letraCorrecta = letras.charAt(num%23);
64
65 if (letraDNI == letraCorrecta) return true;
66 else return false;
67 }
68
69 public void asignarLetraDNI () {
70     int num=0;
71     boolean numeroValido = true;
72     String numero= this.dni;
73
74     char letraCorrecta = ' ';
75     try {
76         num = Integer.parseInt(numero);
77     } catch (NumberFormatException e) {
78         String letras = "TRMAGMYFPDXBNJZSQVHLCKE";
79         if (numeroValido) {
80             letraCorrecta = letras.charAt(num%23);
81         }
82         if (letraCorrecta != letraDNI) {
83             return false;
84         }
85     }
86     boolean esMayorDeEdad () {
87         if (ChronoUnit.YEARS.between(LocalDate.now(), LocalDate.of(2000, 1, 1)) > 18)
88             return true;
89         return false;
90     }
91     private void comprobacion () {
92         // ...
93     }
94 }

```

Figura 18

```

public void asignarLetraDni() {
    char letraCorrecta = getLetraCorrecta();
    if (letraCorrecta != ' ') this.dni+=letraCorrecta;
}

1 usage
private char getLetraCorrecta() {
    int num=0;
    boolean numeroValido = true;
    String numero= this.dni;

    char letraCorrecta = ' ';
    try {
        num = Integer.parseInt(numero);
    } catch (NumberFormatException numberFormatException) { numeroValido = false;}
    String letras = "TRWAGMYFPDXBNJZSQVHLCKE";
    if (numeroValido) {
        letraCorrecta=letras.charAt(num%23);
    }
    return letraCorrecta;
}

```

Figura 19

Na classe Main, converte a classe anônima que está como segundo parâmetro do método Collections.sort nunha clase membro.

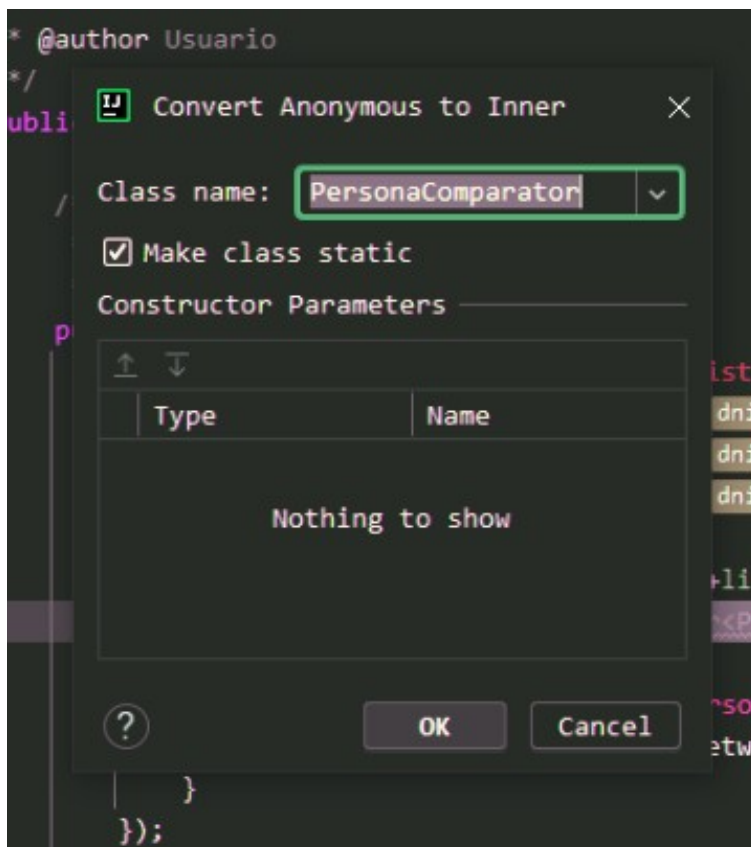


Figura 20

Realizar as seguintes reestruturacións sobre o proxecto MTB.

Move o atributo marcha, o método getMarcha e o método setMarcha dende a superclase Bicicleta á subclase MTB ca opción de menú Reestructurar/Descender probablemente quede algún problema que solucionar e non se faga todo automaticamente soluciona e explica dito problema.

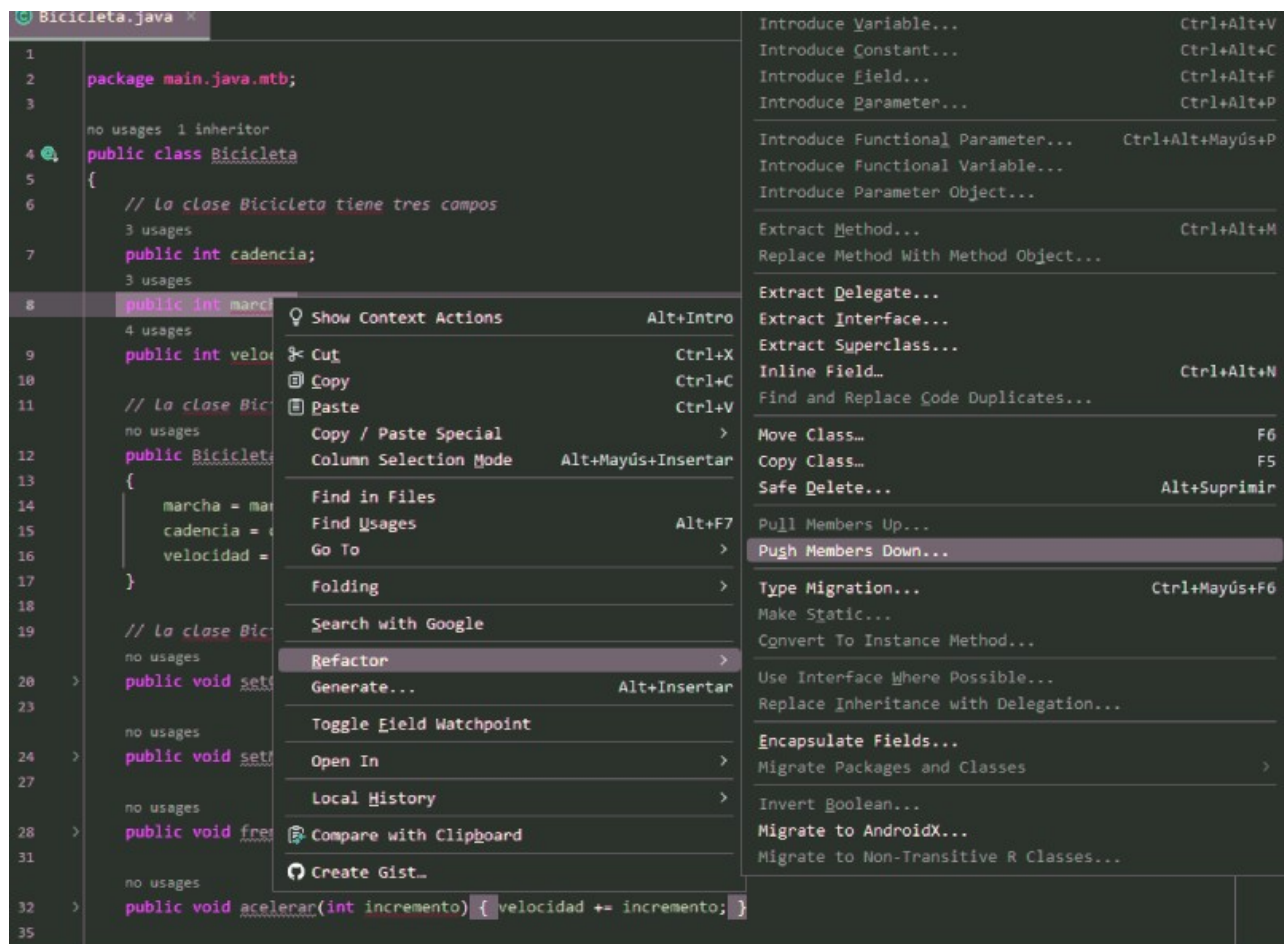


Figura 21

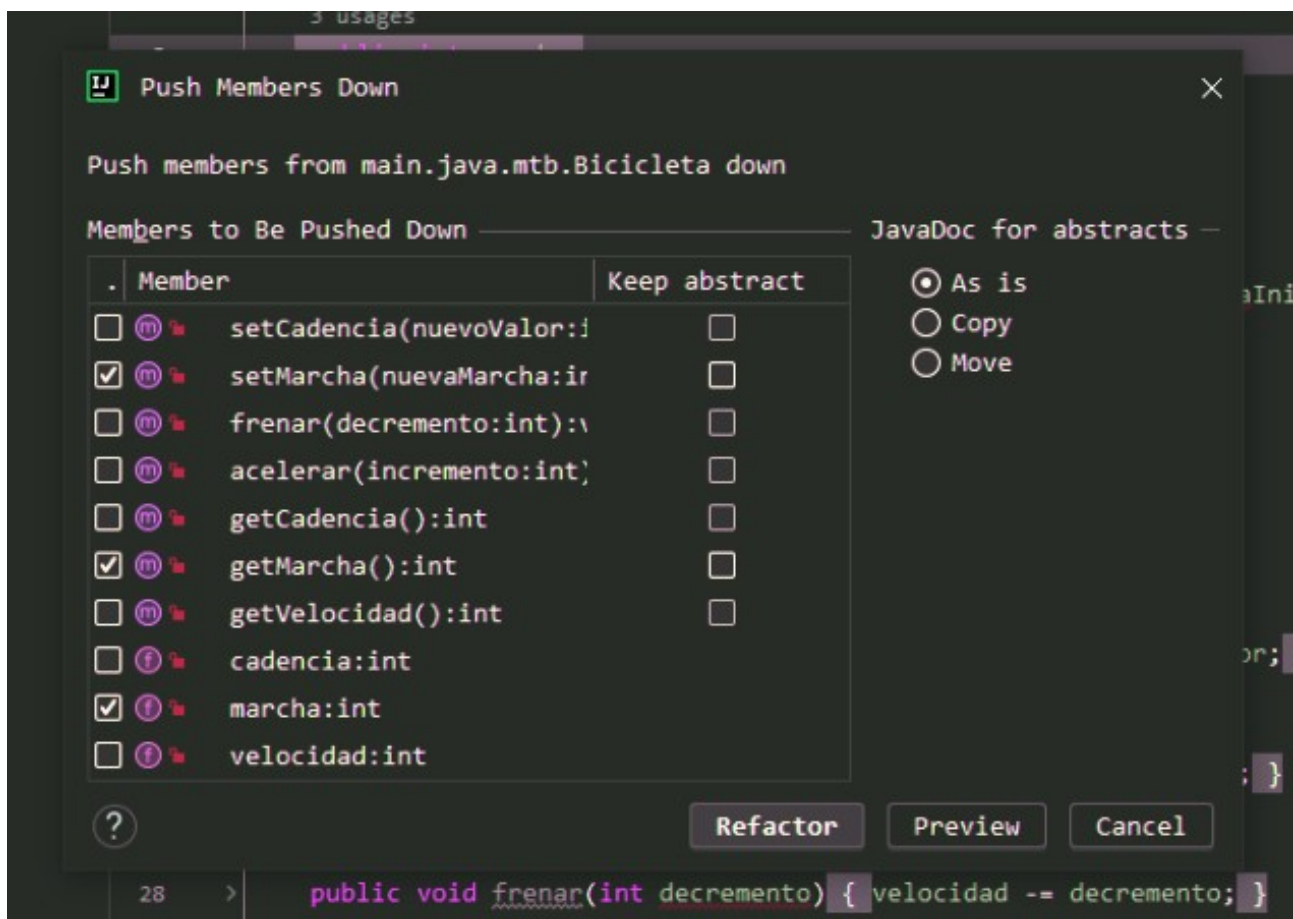


Figura 22

El problema que se produce es que la clase `Bicicleta` utiliza el atributo `marcha`, por lo que si lo cambiamos de clase, el constructor se queda sin este atributo.

Extrae un interface para os métodos getAltoAsiento e setAltoAsiento, usando a opción de menú Reestructurar/Extraer Interface.

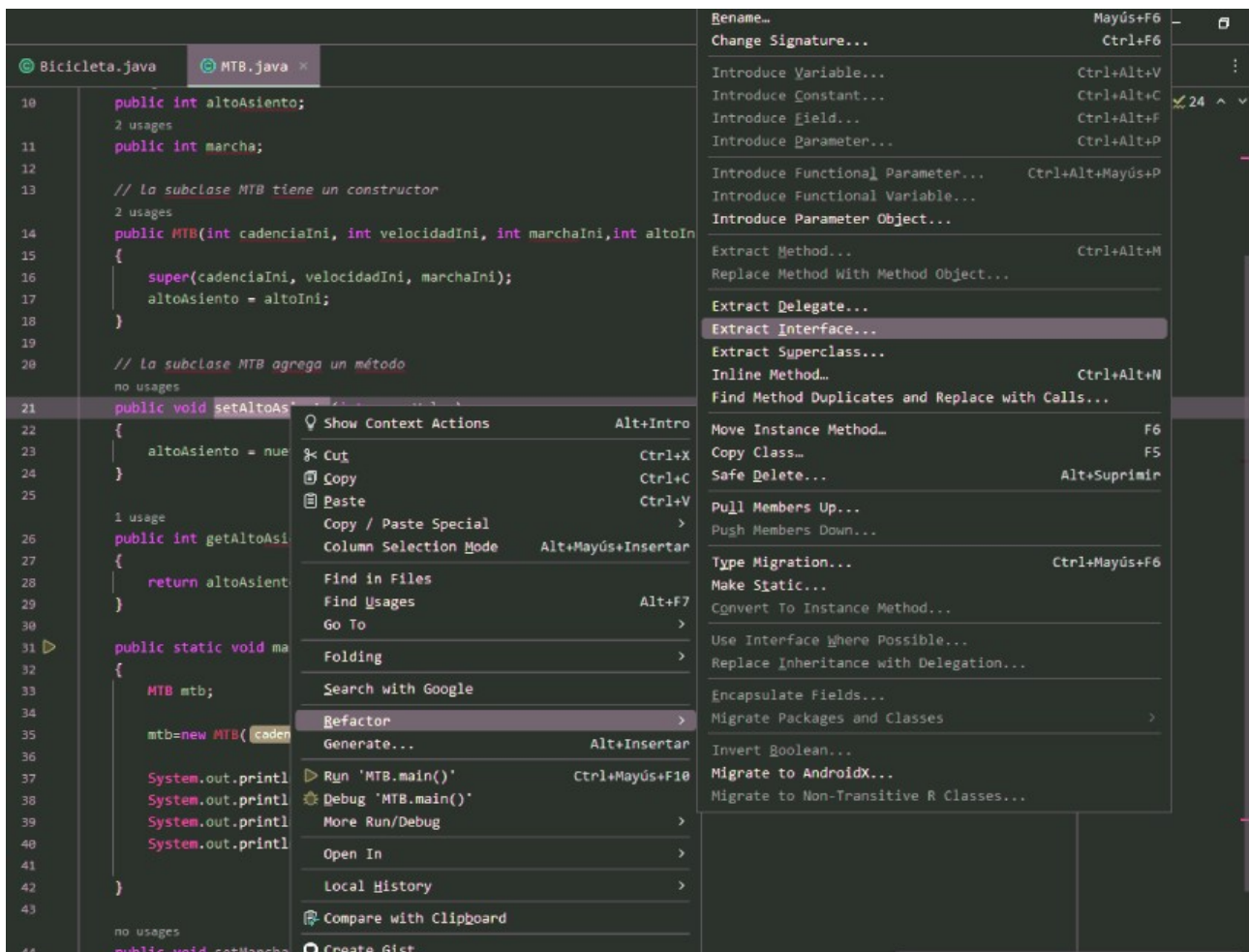


Figura 23

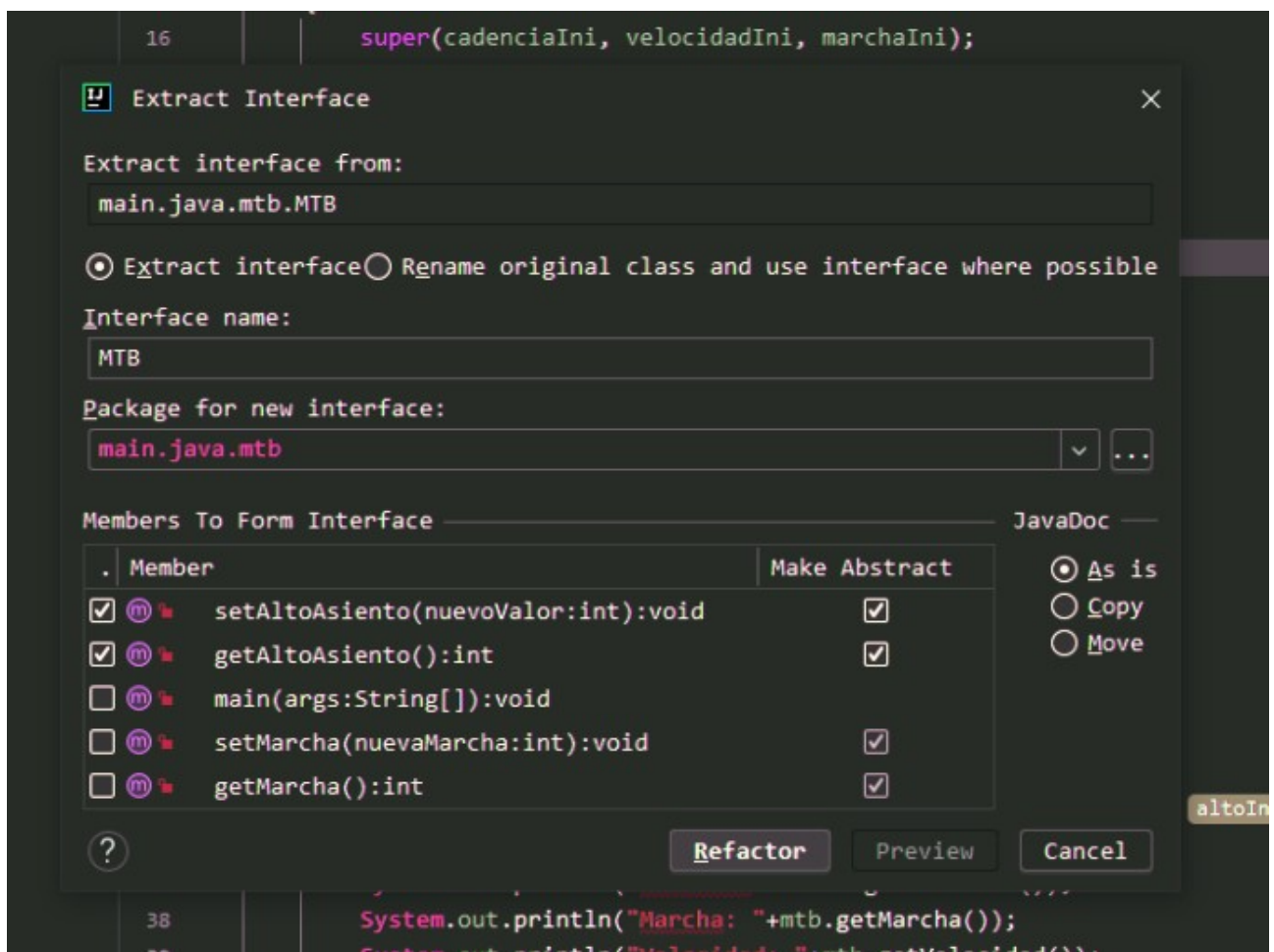


Figura 24

Extrae unha superclase co campo velocidade e os métodos getVelocidad, acelerar e frenar usando a opción de menú Reestructurar/Extraer Superclase.

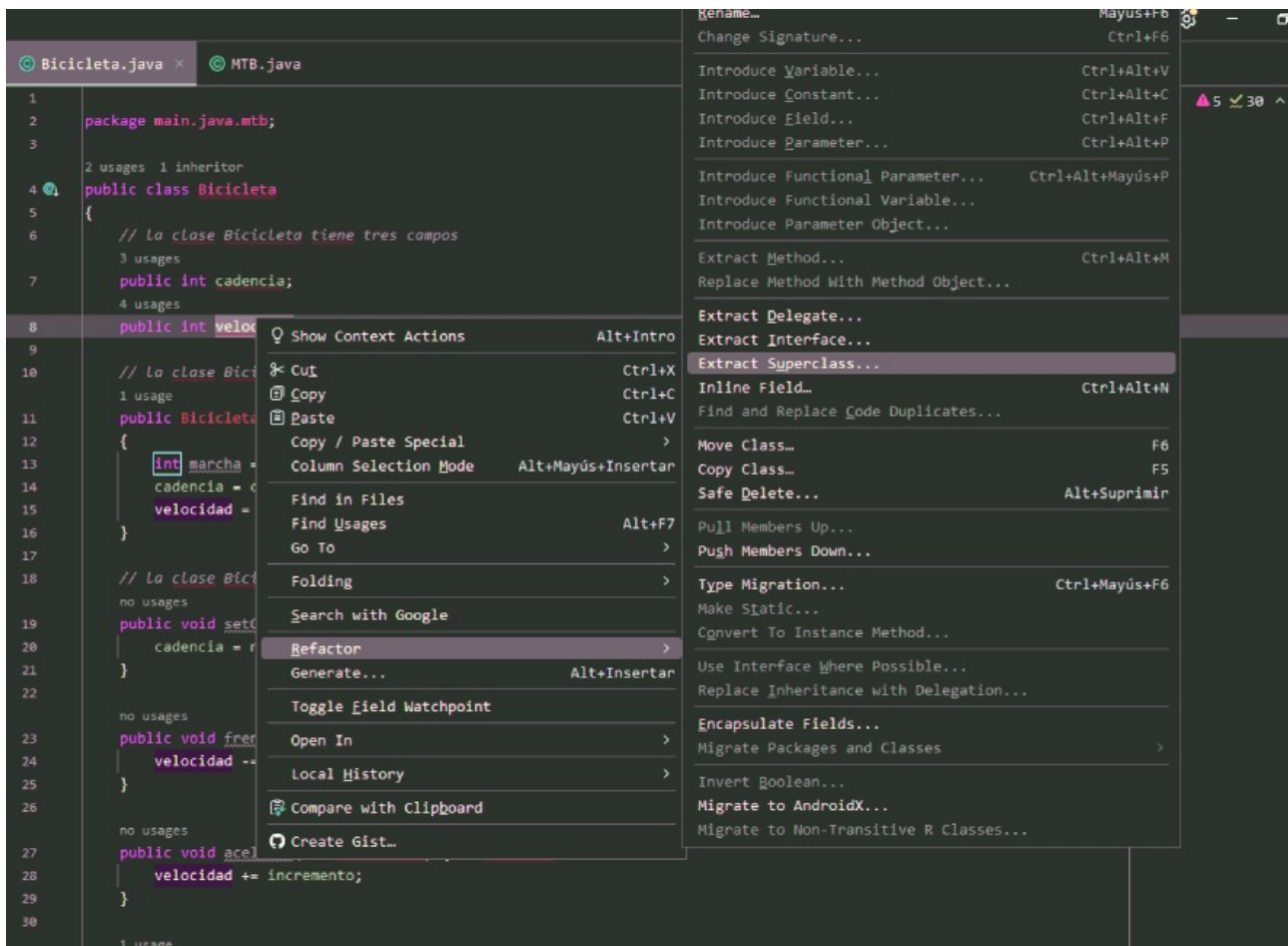


Figura 25

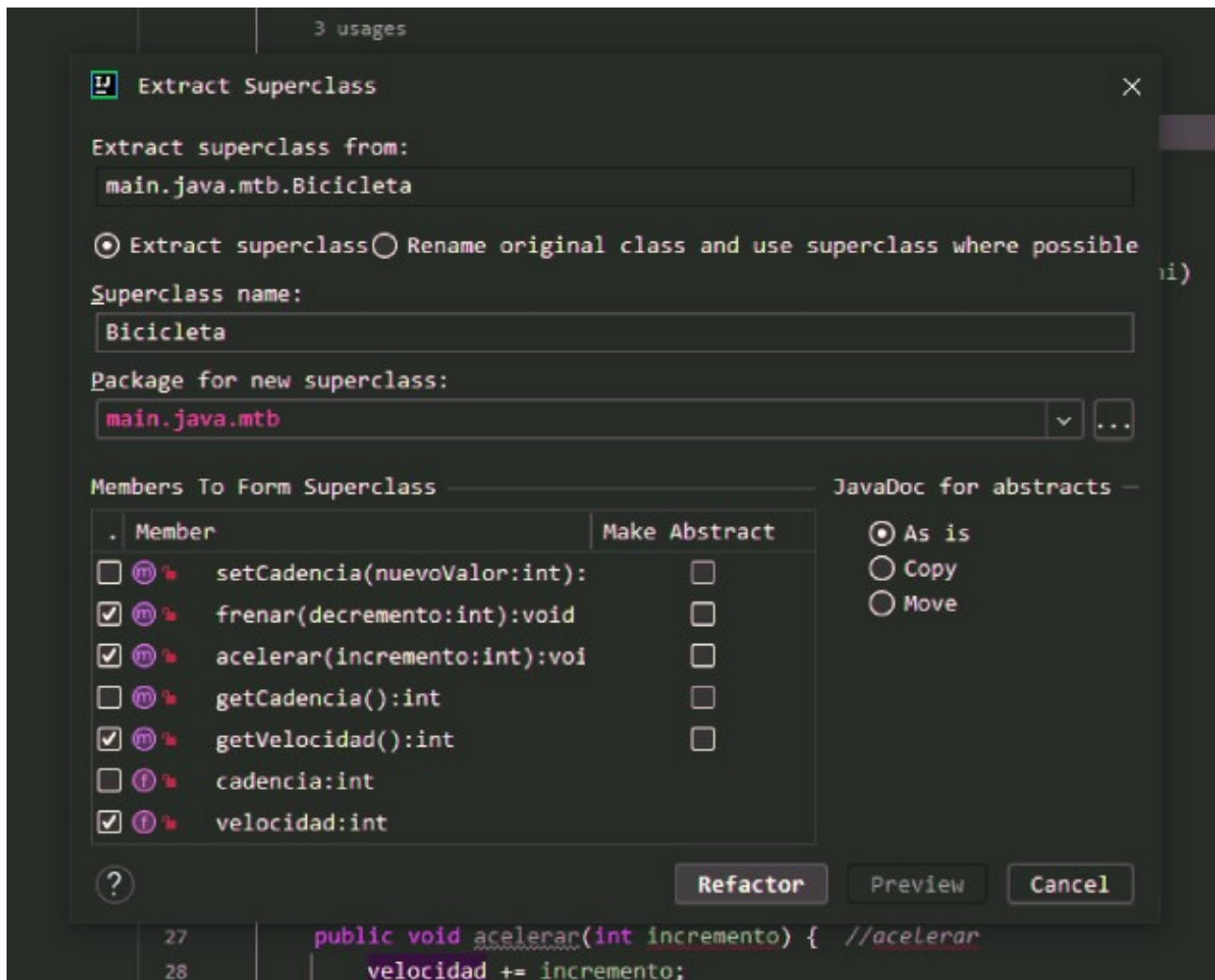


Figura 26

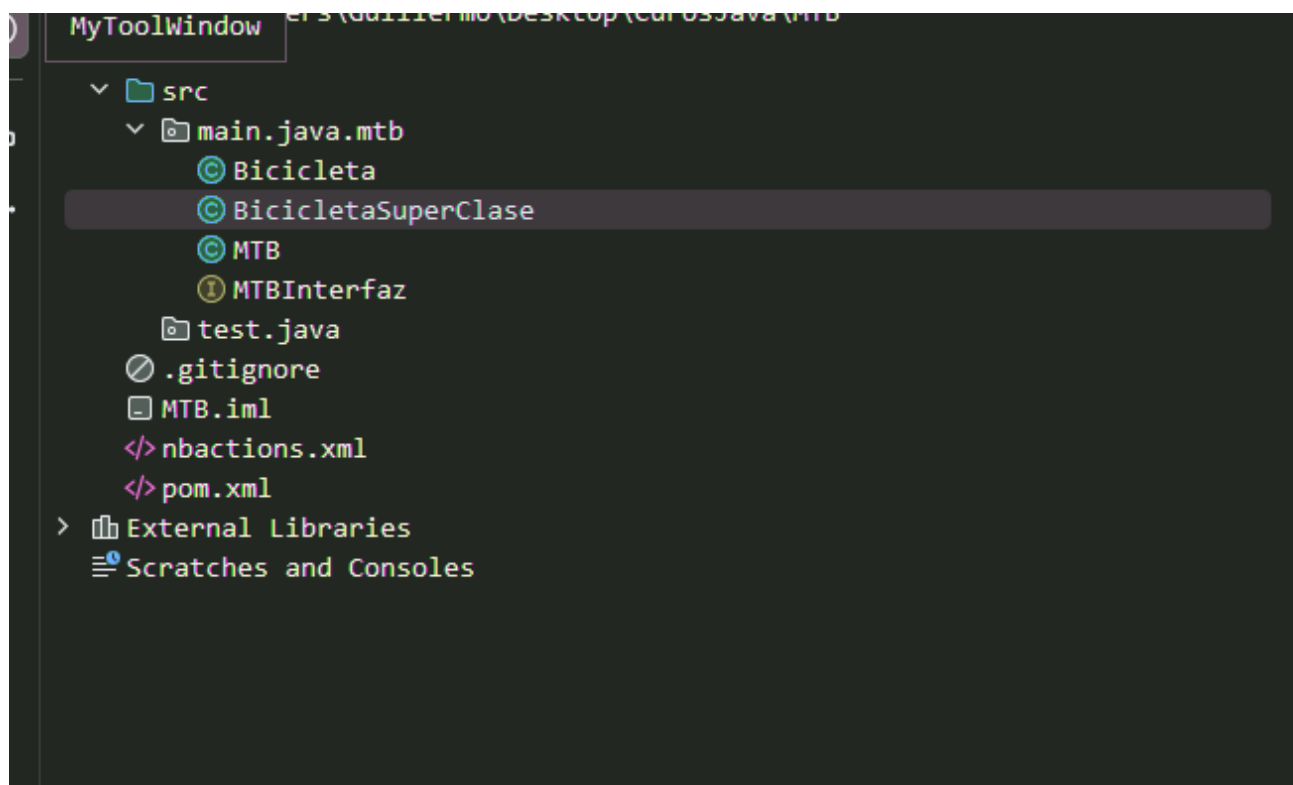


Figura 27