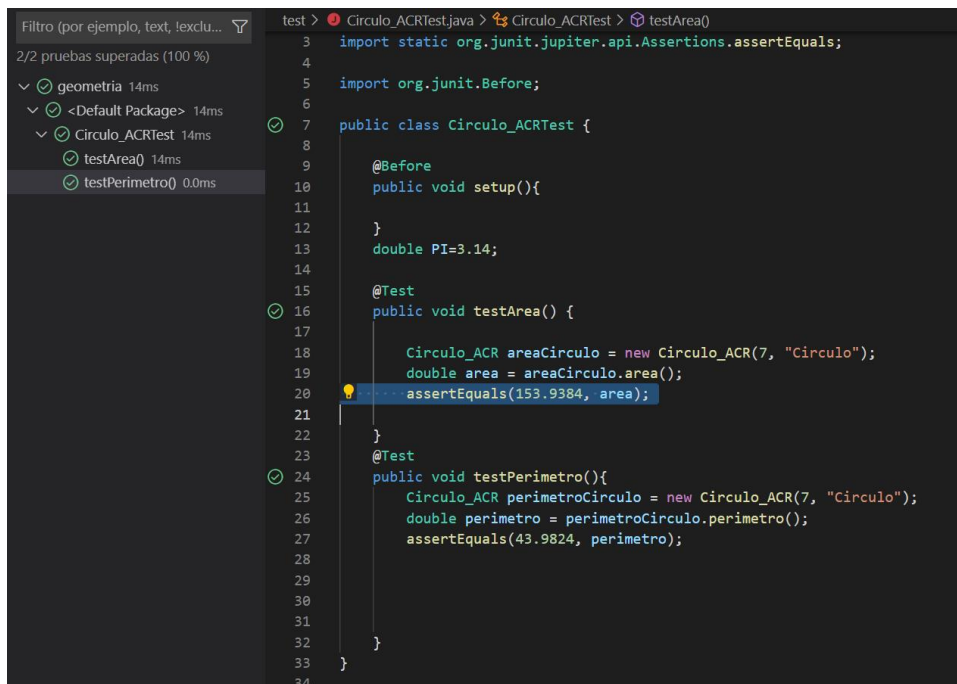


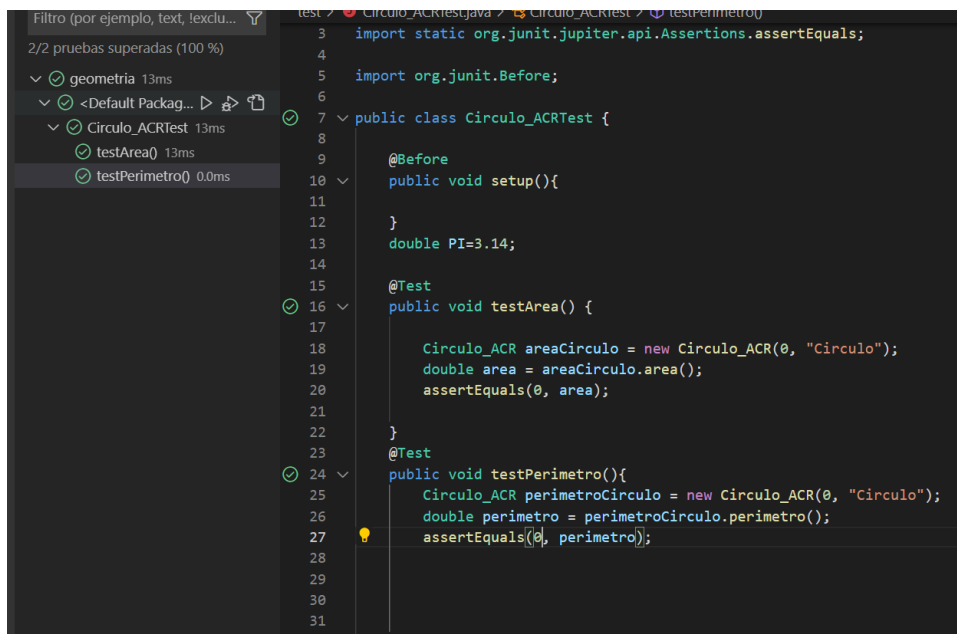
Radio = 7



The screenshot shows an IDE with a test file named `Circulo_ACRTest.java`. The left sidebar displays a test runner with 2/2 tests passed (100%). The test results list includes `geometria` (14ms), `<Default Package>` (14ms), `Circulo_ACRTest` (14ms), `testArea()` (14ms), and `testPerimetro()` (0.0ms). The main editor shows the following code:

```
1  import static org.junit.jupiter.api.Assertions.assertEquals;
2
3  import org.junit.Before;
4
5  public class Circulo_ACRTest {
6
7      @Before
8      public void setup(){
9
10     }
11
12     double PI=3.14;
13
14     @Test
15     public void testArea() {
16
17         Circulo_ACR areaCirculo = new Circulo_ACR(7, "Circulo");
18         double area = areaCirculo.area();
19         assertEquals(153.9384, area);
20     }
21
22     @Test
23     public void testPerimetro(){
24         Circulo_ACR perimetroCirculo = new Circulo_ACR(7, "Circulo");
25         double perimetro = perimetroCirculo.perimetro();
26         assertEquals(43.9824, perimetro);
27     }
28 }
29
30
31
32
33
34
```

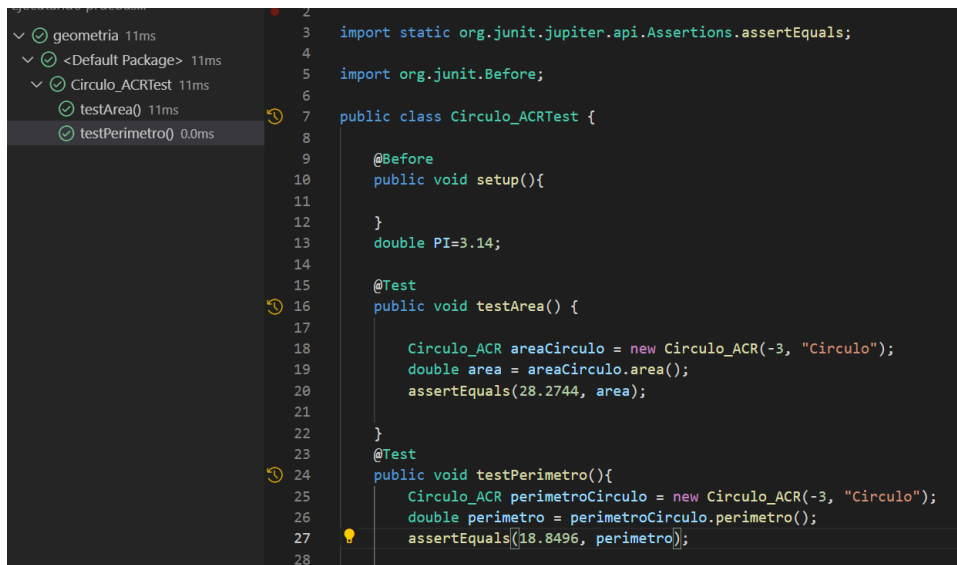
Radio = 0



The screenshot shows the same IDE with the test file `Circulo_ACRTest.java`. The left sidebar shows the test runner with 2/2 tests passed (100%). The test results list includes `geometria` (13ms), `<Default Packag...>` (13ms), `Circulo_ACRTest` (13ms), `testArea()` (13ms), and `testPerimetro()` (0.0ms). The main editor shows the following code:

```
1  import static org.junit.jupiter.api.Assertions.assertEquals;
2
3  import org.junit.Before;
4
5  public class Circulo_ACRTest {
6
7      @Before
8      public void setup(){
9
10     }
11
12     double PI=3.14;
13
14     @Test
15     public void testArea() {
16
17         Circulo_ACR areaCirculo = new Circulo_ACR(0, "Circulo");
18         double area = areaCirculo.area();
19         assertEquals(0, area);
20     }
21
22     @Test
23     public void testPerimetro(){
24         Circulo_ACR perimetroCirculo = new Circulo_ACR(0, "Circulo");
25         double perimetro = perimetroCirculo.perimetro();
26         assertEquals(0, perimetro);
27     }
28 }
29
30
31
32
33
34
```

Radio = -3



The screenshot shows an IDE with a test results pane on the left and a source code editor on the right. The test results pane shows a successful test run for the `Circulo_ACRTest` class, with two test methods: `testArea()` (11ms) and `testPerimetro()` (0.0ms). The source code editor shows the implementation of the `Circulo_ACRTest` class, which includes imports for JUnit, a `@Before` method for setup, and two `@Test` methods: `testArea()` and `testPerimetro()`. Both tests use a `Circulo_ACR` object with a radius of -3 and assert the area and perimeter values.

```
2
3 import static org.junit.jupiter.api.Assertions.assertEquals;
4
5 import org.junit.Before;
6
7 public class Circulo_ACRTest {
8
9     @Before
10     public void setup(){
11
12     }
13     double PI=3.14;
14
15     @Test
16     public void testArea() {
17
18         Circulo_ACR areaCirculo = new Circulo_ACR(-3, "Circulo");
19         double area = areaCirculo.area();
20         assertEquals(28.2744, area);
21     }
22
23     @Test
24     public void testPerimetro(){
25
26         Circulo_ACR perimetroCirculo = new Circulo_ACR(-3, "Circulo");
27         double perimetro = perimetroCirculo.perimetro();
28         assertEquals(18.8496, perimetro);
29     }
30 }
```