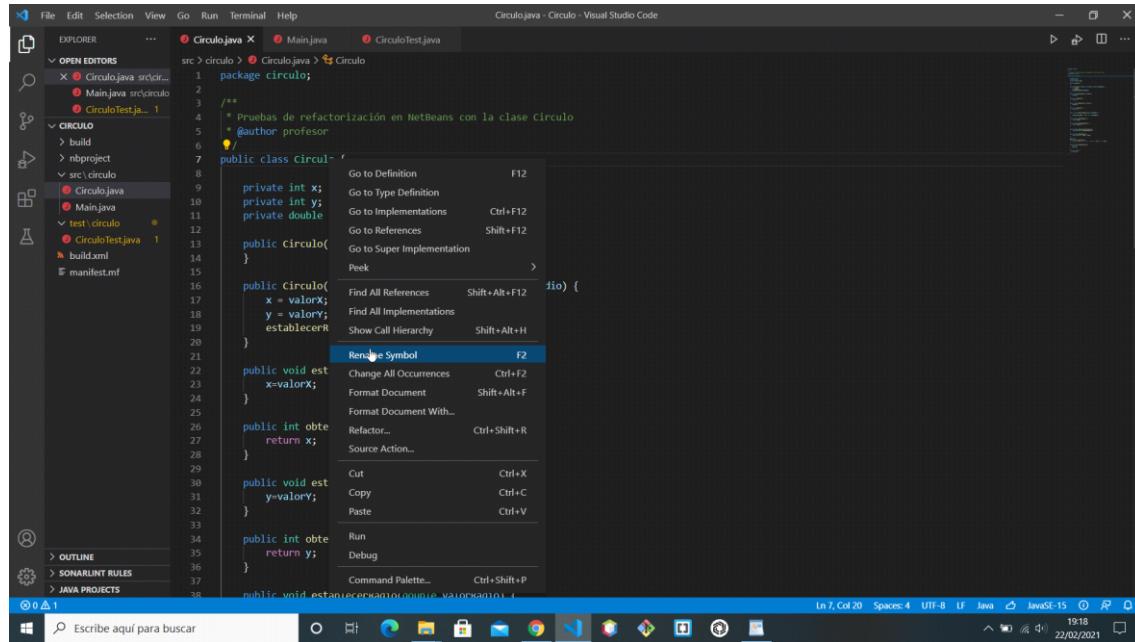


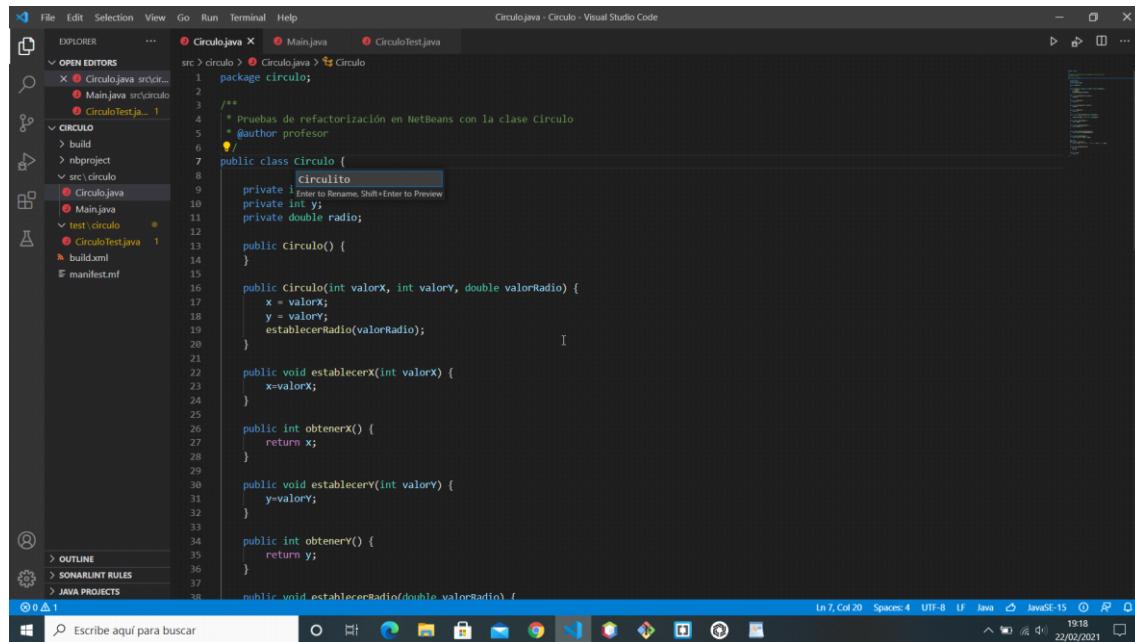
## Tarea ED04

### Ejercicio 1

Renombrar a clase Circulo por Circulito.



The screenshot shows the Visual Studio Code interface with the file `Circulo.java` open. A context menu is displayed over the word `Circulo` in the code editor. The menu item `Rename Symbol` is highlighted with a blue selection bar. Other options in the menu include `Go to Definition`, `Find All References`, `Change All Occurrences`, and `Refactor...`. The code editor shows Java code for a `Circulo` class with methods `establecerX`, `obtenerX`, `establecerY`, `obtenerY`, and `establecerRadio`.



The screenshot shows the Visual Studio Code interface with the file `Circulo.java` open. The word `Circulo` has been replaced by `Circulito` throughout the code. The code editor shows the modified Java code for the `Circulito` class.

The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer Bar:** Shows the project structure under "OPEN EDITORS".
- Code Editor:** Displays the `Círculo.java` file with the following code:

```
src > circulo > Círculo.java > CirculoTest.java
1 package circulo;
2
3 /**
4 * Pruebas de refactorización en NetBeans con la clase Circulo
5 * @author profesor
6 */
7
7 public class Circulo {
8
9     private int x;
10    private int y;
11    private double radio;
12
13    public Circulo() {
14    }
15
16    public Circulo(int valorX, int valorY, double valorRadio) {
17        x = valorX;
18        y = valorY;
19        establecerRadio(valorRadio);
20    }
21
22    public void establecerX(int valorX) {
23        x=valorX;
24    }
25
26    public int obtenerX() {
27        return x;
28    }
29
30    public void establecerY(int valorY) {
31        y=valorY;
32    }
33
34    public int obtenerY() {
35        return y;
36    }
37
38    public void establecerRadio(double valorRadio) {
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70 }
```

- Bottom Status Bar:** Línea 19, Columna 37, Espacios: 4, UTF-8, LF, Java, JavaSE-15, 19:19, 22/02/2021.

Renombrar o método ObtenerArea por ObtenerAreaCírculo.

The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer Bar:** Shows the project structure under "OPEN EDITORS".
- Code Editor:** Displays the `Círculo.java` file with the following code. A context menu is open over the `obtenerArea` method:

```
src > circulo > Círculo.java > CirculoTest.java
34     public int obtenerY() {
35         return y;
36     }
37
38     public void establecerRadio(double valorRadio) {
39         radio=(valorRadio <
40             public double obtenerRai
41             return radio;
42         }
43
44         public double obtenerRai
45             return radio * 2;
46
47         public double obtenerradio
48             return Math.PI * ob
49
50         public double obtenerarea
51             return Math.PI * ra
52
53         @Override
54         public String toString(
55             return "Centro = ["
56
57         public void trasladarCe
58             x=x + 5;
59             y=y + 5;
60
61         @Override
62         public int a(){
63
64
65
66
67
68
69
70 }
```

The context menu for the `obtenerArea` method includes the following options:

- Rename Symbol (F2)
- Change All Occurrences (Ctrl+F2)
- Format Document (Shift+Alt+F)
- Format Document With...
- Format Selection (Ctrl+K Ctrl+F)
- Refactor...
- Source Action...
- Cut (Ctrl+X)
- Copy (Ctrl+C)
- Paste (Ctrl+V)
- Run
- Debug

- Bottom Status Bar:** Línea 56, Columna 11 (seleccionado), Espacios: 4, UTF-8, LF, Java, JavaSE-15, 19:21, 22/02/2021.

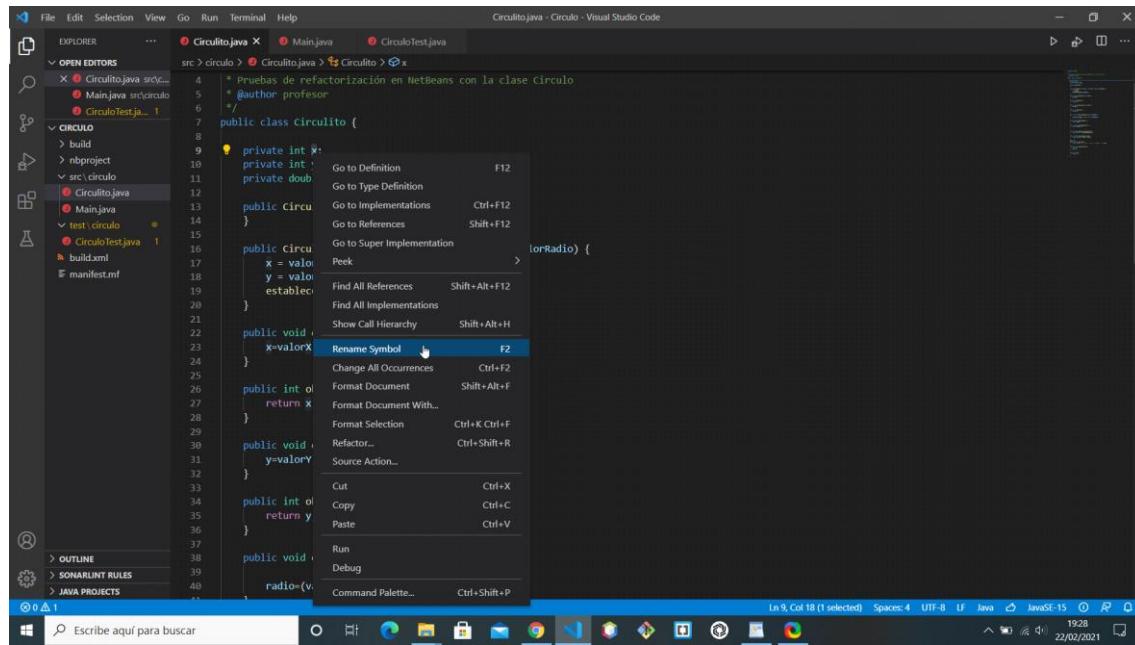
The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor Area:** Shows the `Circulo.java` file content. The cursor is at line 56, column 37, selecting the method `obtenerAreaCirculo()`.

```
public int obtenerY() {  
    return y;  
}  
  
public void establecerRadio(double valorRadio) {  
    radio=(valorRadio < 0.0 ? 0.0 : valorRadio);  
}  
  
public double obtenerRadio() {  
    return radio;  
}  
  
public double obtenerDiametro() {  
    return radio * 2;  
}  
  
public double obtenerCircunferencia() {  
    return Math.PI * obtenerDiametro();  
}  
  
public double obtenerArea() {  
    return Math.PI * obtenerAreaCirculo();  
}  
} @Override  
public String toString() {  
    return "Centro = [" + x + "," + y + "]; Radio = " + radio;  
}  
  
public void trasladarCentro(){  
    x=x + 5;  
    y=y + 5;  
}  
  
public int a(){
```
- Explorer Bar:** Shows the project structure under the `CIRULO` folder, including `Circulo.java`, `Main.java`, and `CirculoTest.java`.
- Bottom Status Bar:** Shows the current line (Ln 56), column (Col 37), and character count (11 selected). It also displays encoding (UTF-8), file type (Java), Java version (JavaSE-15), and the date and time (19:22, 22/02/2021).

This screenshot is nearly identical to the one above, showing the same Java code for the `Circulo.java` file in Visual Studio Code. The cursor is again at line 56, column 37, selecting the method `obtenerAreaCirculo()`. The project structure in the Explorer bar and the status bar at the bottom are also identical.

Renombrar os campos x e y por coordenadaX e coordenadaY.



The screenshot shows the Visual Studio Code interface with the file `Circuito.java` open. A context menu is displayed over the variable `x` at line 9, column 18. The menu includes options like `Rename Symbol`, `Change All Occurrences`, and `Format Document`. The variable `x` is highlighted in blue.

```
File Edit Selection View Go Run Terminal Help
Circuito.java - Visual Studio Code

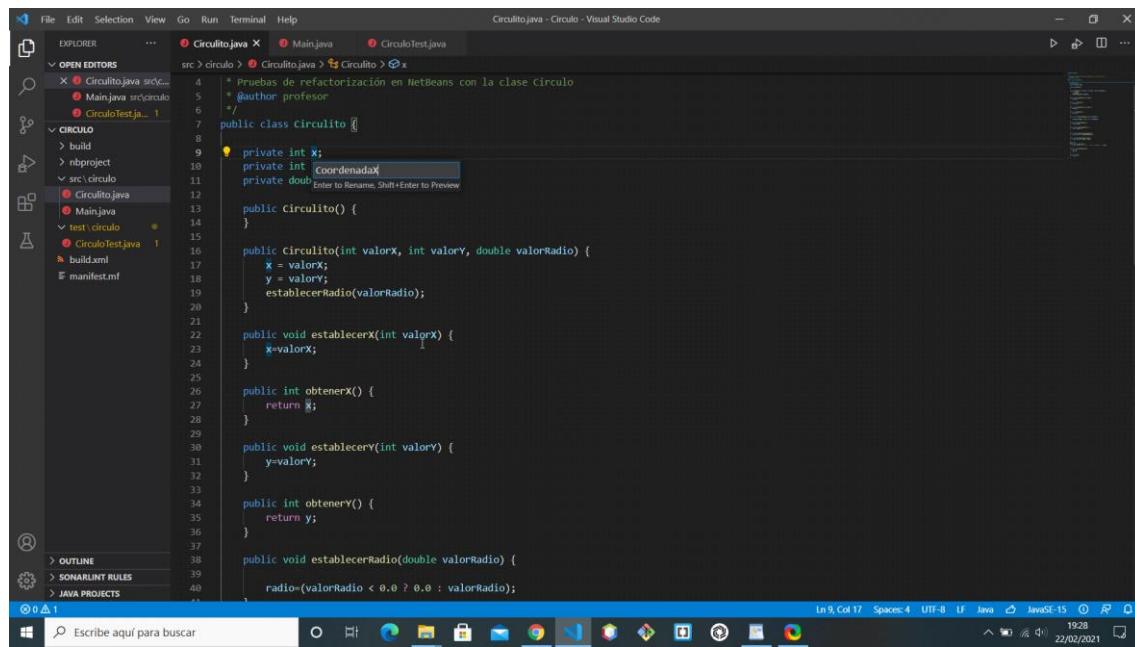
OPEN EDITORS
  Circuito.java
    Main.java
    CircuitoTest.java

  CIRCULO
    build
    nbproject
    src\circulo
      Circuito.java
        Main.java
        CircuitoTest.java
      build.xml
      manifest.mf

  OUTLINE
  SONARLINT RULES
  JAVA PROJECTS

Escribe aquí para buscar
In 9, Col 18 (1 selected) Spaces: 4 UTF-8 LF Java ⌂ JavaSE-15 19:28 22/02/2021
```

```
public class Circuito {
    private int x;
    private int y;
    private double radio;
    public Circuito() {
    }
    public Circuito(int valorX, int valorY, double valorRadio) {
        x = valorX;
        y = valorY;
        establecerRadio(valorRadio);
    }
    public void establecerX(int valorX) {
        x=valorX;
    }
    public int obtenerX() {
        return x;
    }
    public void establecerY(int valorY) {
        y=valorY;
    }
    public int obtenerY() {
        return y;
    }
    public void establecerRadio(double valorRadio) {
        radio=(valorRadio < 0.0 ? 0.0 : valorRadio);
    }
}
```



The screenshot shows the Visual Studio Code interface with the file `Circuito.java` open. A renamer dialog is displayed over the variable `x` at line 9, column 18. The placeholder text `Enter to Rename, Shift+Enter to Preview` is visible. The variable `x` is highlighted in blue.

```
File Edit Selection View Go Run Terminal Help
Circuito.java - Visual Studio Code

OPEN EDITORS
  Circuito.java
    Main.java
    CircuitoTest.java

  CIRCULO
    build
    nbproject
    src\circulo
      Circuito.java
        Main.java
        CircuitoTest.java
      build.xml
      manifest.mf

  OUTLINE
  SONARLINT RULES
  JAVA PROJECTS

Escribe aquí para buscar
In 9, Col 17 (1 selected) Spaces: 4 UTF-8 LF Java ⌂ JavaSE-15 19:28 22/02/2021
```

```
public class Circuito {
    private int x;
    private int CoordenadaX;
    private double radio;
    public Circuito() {
    }
    public Circuito(int valorX, int valorY, double valorRadio) {
        x = valorX;
        y = valorY;
        establecerRadio(valorRadio);
    }
    public void establecerX(int valorX) {
        x=valorX;
    }
    public int obtenerX() {
        return x;
    }
    public void establecerY(int valorY) {
        y=valorY;
    }
    public int obtenerY() {
        return y;
    }
    public void establecerRadio(double valorRadio) {
        radio=(valorRadio < 0.0 ? 0.0 : valorRadio);
    }
}
```

Screenshot of Visual Studio Code showing the context menu for variable 'y' in the file Circuito.java. The variable is highlighted with a yellow circle. The context menu is open, showing options like 'Rename Symbol' (F2), 'Change All Occurrences' (Ctrl+F2), 'Format Document' (Shift+Alt+F), and 'Format Selection' (Ctrl+K Ctrl+F). Other options include 'Cut' (Ctrl+X), 'Copy' (Ctrl+C), 'Paste' (Ctrl+V), 'Run', and 'Debug'. The status bar at the bottom shows 'Ln 10, Col 18 (1 selected) Spaces: 4 UTF-8 LF Java JavaSE-15 19:30 22/02/2021'.

Screenshot of Visual Studio Code showing the same code editor after renaming. The variable 'y' has been renamed to 'CoordenadaY'. A tooltip 'Enter to Rename, Shift+Enter to Preview' is visible near the cursor. The status bar at the bottom shows 'Ln 10, Col 18 (1 selected) Spaces: 4 UTF-8 LF Java JavaSE-15 19:30 22/02/2021'.

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor:** The main window displays the `Circuito.java` file. The code defines a class `Circuito` with private fields `CoordenadaX` and `CoordenadaY`, and a double field `radio`. It includes methods to set and get coordinates and radius, and a constructor to initialize them.
- Explorer:** Shows the project structure with files like `Circuito.java`, `Main.java`, `CirculoTest.java`, and build-related files.
- Search Bar:** Escribe aquí para buscar.
- Bottom Status Bar:** Ln 10, Col 28, Spaces: 4, UTF-8, LF, Java, 19:30, 22/02/2021.

Introducir constante LIMITERADIO de tipo “double” con el valor 0.0.

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor:** The main window displays the `Circuito.java` file. A context menu is open over the line `radio=(valorRadio < 0.0 ? 0.0 : valorRadio);`, with the "Refactor..." option highlighted.
- Explorer:** Shows the project structure with files like `Circuito.java`, `Main.java`, `CirculoTest.java`, and build-related files.
- Search Bar:** Escribe aquí para buscar.
- Bottom Status Bar:** Ln 40, Col 32 (3 selected), Spaces: 4, UTF-8, LF, Java, 19:32, 22/02/2021.

```
src > circulo > Circuito.java > CircuitoTest.java
1 package circulo;
2
3 /**
4 * Pruebas de refactorización en NetBeans con la clase Circulo
5 * @author profesor
6 */
7
8 public class Circuito {
9
10     private static final double LIMITERADIO = 0.0;
11     private int CoordenadaX;
12     private int CoordenadaY;
13     private double radio;
14
15     public Circuito() {
16
17     }
18
19     public Circuito(int valorX, int valorY, double valorRadio) {
20         CoordenadaX = valorX;
21         CoordenadaY = valorY;
22         establecerRadio(valorRadio);
23
24     }
25
26     public void establecerX(int valorX) {
27         CoordenadaX = valorX;
28     }
29
30     public int obtenerX() {
31         return CoordenadaX;
32     }
33
34     public void establecerY(int valorY) {
35         CoordenadaY = valorY;
36     }
37
38     public int obtenerY() {
39         return CoordenadaY;
40     }
41 }
```

```
src > circulo > Circuito.java > CircuitoTest.java
1 package circulo;
2
3 /**
4 * Pruebas de refactorización en NetBeans con la clase Circulo
5 * @author profesor
6 */
7
8 public class Circuito {
9
10     private static final double LIMITERADIO = 0.0;
11     private int CoordenadaX;
12     private int CoordenadaY;
13     private double radio;
14
15     public Circuito() {
16
17     }
18
19     public Circuito(int valorX, int valorY, double valorRadio) {
20         CoordenadaX = valorX;
21         CoordenadaY = valorY;
22         establecerRadio(valorRadio);
23
24     }
25
26     public void establecerX(int valorX) {
27         CoordenadaX = valorX;
28     }
29
30     public int obtenerX() {
31         return CoordenadaX;
32     }
33
34     public void establecerY(int valorY) {
35         CoordenadaY = valorY;
36     }
37
38     public int obtenerY() {
39         return CoordenadaY;
40     }
41 }
```

Eliminar de forma segura los métodos obtenerX, obtenerY, establecerX, establecerY e establecerRadio que agora son innecesarios haciendo los cambios necesarios en el código para que sean substituidos por los correspondientes métodos tipo get e set creados.

Seleccionamos lo que vamos a borrar.

```
File Edit Selection View Go Run Terminal Help
Circuito.java - Circulo - Visual Studio Code
OPEN EDITORS
src > circulo > Circuito.java > Circulo > establecerRadio()
  public CIRCULO(int valorX, int valorY, double valorRadio) {
    CoordenadaX = valorX;
    CoordenadaY = valorY;
    establecerRadio(valorRadio);
  }
  public void establecerX(int valorX) {
    CoordenadaX=valorX;
  }
  public int obtenerX() {
    return CoordenadaX;
  }
  public void establecerY(int valorY) {
    CoordenadaY=valorY;
  }
  public int obtenerY() {
    return CoordenadaY;
  }
  public void establecerRadio(double valorRadio) {
    radio=valorRadio < LIMITERADIO ? 0.0 : valorRadio;
  }
  public double obtenerRadio() {
    return radio;
  }
  public double obtenerDiametro() {
    return radio * 2;
  }
  public double obtenerCircunferencia() {
    return Math.PI * obtenerDiametro();
  }

```

```
File Edit Selection View Go Run Terminal Help
Circuito.java - Circulo - Visual Studio Code
OPEN EDITORS
src > circulo > Circuito.java > Circulo
  package circulo;
  /**
   * Pruebas de refactorización en NetBeans con la clase Circulo
   * @author profesor
   */
  public class Circulo {
    private static final double LIMITERADIO = 0.0;
    private int CoordenadaX;
    private int CoordenadaY;
    private double radio;
    public Circulo() {
    }
    public Circulo(int valorX, int valorY, double valorRadio) {
      CoordenadaX = valorX;
      CoordenadaY = valorY;
      establecerRadio(valorRadio);
    }
    public double obtenerDiametro() {
      return radio * 2;
    }
    public double obtenerCircunferencia() {
      return Math.PI * obtenerDiametro();
    }
    public double obtenerAreaCirculo() {
      return Math.PI * radio * radio;
    }
  
```

Cambiamos el método establecerradio por la asignación directa de la variable.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "CIRULO".
- Editor:** Displays the `Circuito.java` file content:

```
src > circulo > Circuito.java > Circuito > Circuito(int, int, double)
1 package circulo;
2
3 /**
4 * Pruebas de refactorización en NetBeans con la clase Circulo
5 * @author profesor
6 */
7
8 public class Circuito {
9
10     private static final double LIMITERADIO = 0.0;
11     private int CoordenadaX;
12     private int CoordenadaY;
13     private double radio;
14
15     public Circuito() {
16
17     }
18
19     public Circuito(int valorX, int valorY, double valorRadio) {
20         CoordenadaX = valorX;
21         CoordenadaY = valorY;
22         radio=valorRadio;
23     }
24
25     public double obtenerDiametro() {
26         return radio * 2;
27     }
28
29     public double obtenerCircunferencia() {
30         return Math.PI * obtenerDiametro();
31     }
32
33     public double obtenerAreaCirculo() {
34         return Math.PI * radio * radio;
35     }
36
37     @Override
38 }
```
- Bottom Bar:** Shows the status bar with "Line 20, Col 26 (17 selected)", "Spaces: 4", "UTF-8", "LF", "Java", "JavaScrip", and the date "26/02/2021".

Generamos automaticamente los getters and setters de las variables.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "CIRULO".
- Editor:** Displays the `Circuito.java` file content, identical to the previous screenshot.
- Context Menu:** A context menu is open over the code, with the "Generate Getters and Setters..." option highlighted.
- Bottom Bar:** Shows the status bar with "Line 23, Col 5", "Spaces: 4", "UTF-8", "LF", "Java", "JavaScrip", and the date "26/02/2021".

Circuito.java - Circulo - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER CIRULO

OPEN EDITORS Circuito.java Main.java CirculoTest.java

src > circulo > Circuito.java CirculoTest.java

src\circulo Circuito.java Main.java CirculoTest.java

nbproject build.xml manifest.mf

test\circulo Circuito.java Main.java CirculoTest.java

CIRULO Circuito.java Main.java CirculoTest.java

src\circulo Circuito.java Main.java CirculoTest.java

nbproject build.xml manifest.mf

test\circulo Circuito.java Main.java CirculoTest.java

Circuito.java

```
public class Circuito {  
    private static final double LIMITERADIO = 0.0;  
    private int CoordenadaX;  
    private int CoordenadaY;  
    private double radio;  
  
    public Circuito() {  
    }  
  
    public Circuito(int valorX, int valorY, double valorRadio) {  
        CoordenadaX = valorX;  
        CoordenadaY = valorY;  
        radio = valorRadio;  
    }  
  
    public double obtenerDiametro() {  
        return radio * 2;  
    }  
  
    public double obtenerCircunferencia() {  
        return Math.PI * obtenerDiametro();  
    }  
  
    public double obtenerAreaCirculo() {  
        return Math.PI * radio * radio;  
    }  
  
    @Override  
    public String toString() {  
        return "Centro [" + CoordenadaX + "," + CoordenadaY + "]; Radio = " + radio;  
    }  
}
```

Outline SonarLint Rules Java Projects

Escribe aquí para buscar

In 23, Col 5 Spaces: 4 UTF-8 LF Java 18:40 26/02/2021

Circuito.java - Circulo - Visual Studio Code

File Edt Selection View Go Run Terminal Help

EXPLORER CIRULO

OPEN EDITORS Circuito.java Main.java CirculoTest.java

src > circulo > Circuito.java CirculoTest.java

src\circulo Circuito.java Main.java CirculoTest.java

nbproject build.xml manifest.mf

test\circulo Circuito.java Main.java CirculoTest.java

CIRULO Circuito.java Main.java CirculoTest.java

src\circulo Circuito.java Main.java CirculoTest.java

nbproject build.xml manifest.mf

test\circulo Circuito.java Main.java CirculoTest.java

Circuito.java

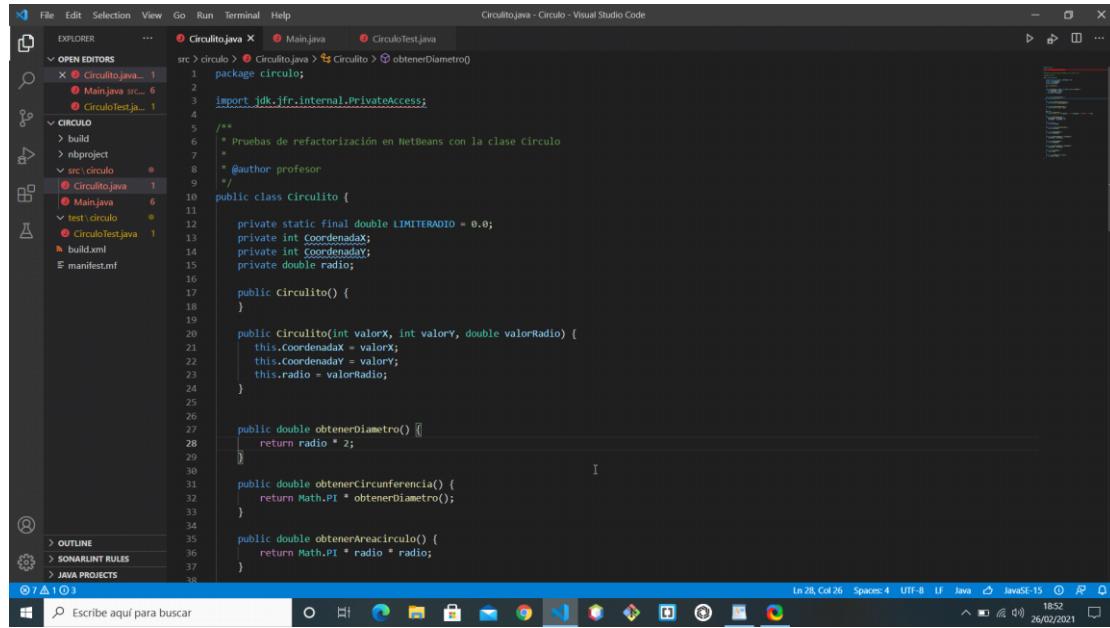
```
public class Circuito {  
    private static final double LIMITERADIO = 0.0;  
    private int CoordenadaX = CoordenadaX + 5;  
    private int CoordenadaY = CoordenadaY + 5;  
  
    public int a() {  
        return CoordenadaX;  
    }  
  
    public static double getLimiteroadio() {  
        return LIMITERADIO;  
    }  
  
    public int getCoordenadaX() {  
        return CoordenadaX;  
    }  
  
    public void setCoordenadaX(int coordenadaX) {  
        CoordenadaX = coordenadaX;  
    }  
  
    public int getCoordenadaY() {  
        return CoordenadaY;  
    }  
  
    public void setCoordenadaY(int coordenadaY) {  
        CoordenadaY = coordenadaY;  
    }  
  
    public double getRadio() {  
        return radio;  
    }  
  
    public void setRadio(double radio) {  
        this.radio = radio;  
    }  
}
```

Outline SonarLint Rules Java Projects

Escribe aquí para buscar

In 77, Col 6 (50 selected) Spaces: 4 UTF-8 LF Java 18:40 26/02/2021

Preparamos o constructor circulito para que siga enviando los datos del circulo a nustra clase.

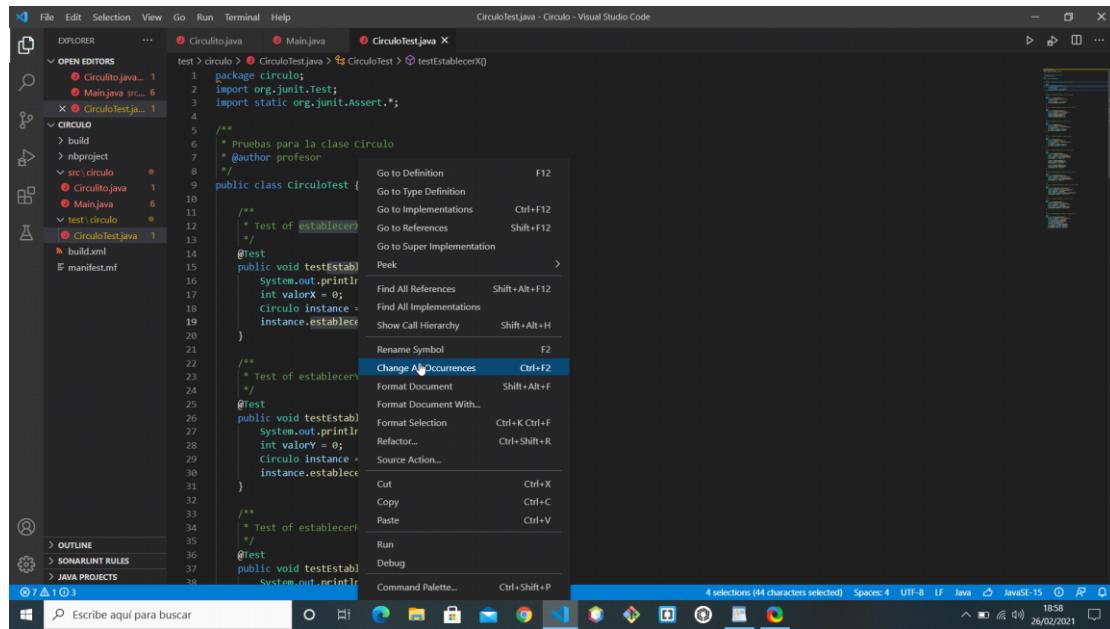


```
File Edit Selection View Go Run Terminal Help Circuito.java - Circulo - Visual Studio Code

EXPLORER ... Circuito.java Main.java CirculoTest.java
src > circulo > Circuito.java > Circulo > obtenerDiametro()
1 package circulo;
2
3 import jdk.jfr.internal.PrivateAccess;
4
5 /**
6 * Pruebas de refactorización en NetBeans con la clase Circulo
7 *
8 * @author profesor
9 */
10 public class Circulo {
11
12     private static final double LIMITERADIO = 0.0;
13     private int CoordenadaX;
14     private int CoordenadaY;
15     private double radio;
16
17     public Circulo() {
18
19
20     }
21
22     public Circulo(int valorX, int valorY, double valorRadio) {
23         this.CoordenadaX = valorX;
24         this.CoordenadaY = valorY;
25         this.radio = valorRadio;
26
27     }
28
29     public double obtenerDiametro() {
30         return radio * 2;
31     }
32
33     public double obtenerCircunferencia() {
34         return Math.PI * obtenerDiametro();
35     }
36
37     public double obtenerAreaCirculo() {
38         return Math.PI * radio * radio;
39     }
}

Outline SonarLint Rules Java Projects
Escrive aquí para buscar
In 2B, Col 26 - Spaces: 4 - UTF-8 - LF - Java - JavaStl-15 - 1852 - 26/02/2021
```

Cambiamos los nombres de los métodos usados no resto do programa para que siga funcionando.



```
File Edit Selection View Go Run Terminal Help CirculoTest.java - Circulo - Visual Studio Code

EXPLORER ... Circuito.java Main.java CirculoTest.java
test > circulo > CirculoTest.java > Circulo > establecerX()
1 package circulo;
2 import org.junit.Test;
3 import static org.junit.Assert.*;
4
5 /**
6 * Pruebas para la clase Circulo
7 * @author profesor
8 */
9 public class CirculoTest {
10
11     /**
12      * Test of establecerX
13      */
14     @Test
15     public void testEstablecerX() {
16         System.out.println("int ValorX = 0; ");
17         int ValorX = 0;
18         Circulo instance = instance.establecerX(ValorX);
19     }
20
21     /**
22      * Test of establecerY
23      */
24     @Test
25     public void testEstablecerY() {
26         System.out.println("int ValorY = 0; ");
27         int ValorY = 0;
28         Circulo instance = instance.establecerY(ValorY);
29     }
30
31     /**
32      * Test of establecerRadio
33      */
34     @Test
35     public void testEstablecerRadio() {
36         System.out.println("double Radio = 0.0; ");
37     }
}

Outline SonarLint Rules Java Projects
Escrive aquí para buscar
In 2B, Col 26 - Spaces: 4 - UTF-8 - LF - Java - JavaStl-15 - 1858 - 26/02/2021
```

```

File Edit Selection View Go Run Terminal Help
CirculoTest.java - Circulo - Visual Studio Code
EXPLORER OPEN EDITORS CírculoTest.java Main.java Circulo.java
Círculo.java... 1 Main.java... 6 CirculoTest.java...
Círculo.java... 1 Main.java... 6 CirculoTest.java...
src/círculo build nbproject test/círculo Circulo.java Main.java CirculoTest.java
build.xml manifest.mf
CírculoTest.java 13
Main.java 6
Círculo.java 1
CirculoTest.java 1
build.xml
manifest.mf
Outline Sonarlint Rules Java Projects
Escribe aquí para buscar
1 package circulo;
2 import org.junit.Test;
3 import static org.junit.Assert.*;
4
5 /**
6  * Pruebas para la clase Circulo
7  * @author profesor
8 */
9 public class CirculoTest {
10
11     /**
12      * Test of getCoordenadaX method, of class Circulo.
13     */
14     @Test
15     public void testgetCoordenadaX() {
16         System.out.println("getCoordenadaX()");
17         int valorX = 0;
18         Circulo instance = new Circulo();
19         instance.getCoordenadaX(valorX);
20     }
21
22     /**
23      * Test of establecerY method, of class Circulo.
24     */
25     @Test
26     public void testestablecerY() {
27         System.out.println("establecerY");
28         int valorY = 0;
29         Circulo instance = new Circulo();
30         instance.establecerY(valorY);
31     }
32
33     /**
34      * Test of establecerRadio method, of class Circulo.
35     */
36     @Test
37     public void testestablecerRadio() {
38         System.out.println("establecerRadio");
39     }
}

```

4 selections Spaces: 4 UTF-8 LF Java 18:58 26/02/2021

etc....

Tamen no main

```

File Edit Selection View Go Run Terminal Help
Main.java - Circulo - Visual Studio Code
EXPLORER OPEN EDITORS Main.java Circulo.java CirculoTest.java
Círculo.java... 1 Main.java... 6 CirculoTest.java...
Círculo.java... 1 Main.java... 6 CirculoTest.java...
src/círculo build nbproject test/círculo Circulo.java Main.java CirculoTest.java
build.xml manifest.mf
Main.java 6
Círculo.java 1
CirculoTest.java 1
build.xml
manifest.mf
Outline Sonarlint Rules Java Projects
Escribe aquí para buscar
1 package circulo;
2
3 import java.text.DecimalFormat;
4
5 /**
6  * Pruebas de refactorización en NetBeans con la clase Circulo
7  * @author profesor
8 */
9 public class Main {
10
11     public static void main(String[] args) {
12         Circulo circulo = new Circulo(37,43,2.5);
13         String salida =
14             "La coordenada X es "+circulo.getCoordenadaX()+"\n"
15             +"La coordenada Y es "+circulo.getCoordenadaY()+" : int\n"
16             +"El radio es "+circulo.getRadio()+" : double\n"
17             +"El diámetro es "+circulo.getDiametro()+" : double\n"
18             +"El perímetro es "+circulo.getPerimetro()+" : double\n"
19             +"El área es "+circulo.getArea()+" : double\n"
20             +"El hashcode es "+circulo.hashCode()+" : int\n"
21             +"DecimalFormat dosDigitos = new DecimalFormat("###.##")\n"
22             +"salida = dosDigitos.format(circulo.getPerimetro()) : double\n"
23             +"salida = dosDigitos.format(circulo.getArea()) : double\n"
24             +"salida = dosDigitos.format(circulo.getCoordenadaX()) : string\n"
25             +"System.out.println(salida); : void\n"
26             +"System.out.println(salida); : void\n"
27         }
28
29     }
}

```

In 13, Col 47 Spaces: 4 UTF-8 LF Java 19:08 26/02/2021

File Edit Selection View Go Run Terminal Help

EXPLORER CIRULO

```
src > circulo > Main.java > Main > main(String[])
1 package circulo;
2
3 import java.text.DecimalFormat;
4
5 /**
6 * Pruebas de refactorización en NetBeans con la clase Circulo
7 * @author profesor
8 */
9
public class Main {
    Run | Debug
    public static void main(String[] args) {
        Circulo circulo = new Circulo(17,43,2.5);
        String salida =
            "La coordenada X es "+circulo.getCoordenadaX()+
            "La coordenada Y es "+circulo.getCoordenadaY()+
            "El radio es "+circulo.obtenerRadio();
        DecimalFormat dosDigitos = new DecimalFormat("###.##");
        int circulo establecerX(35); int a[]; int getCoordenadaX(); int getCoordenadaY(); int establecerRadio(4,2); double establecerRadio(); double DecimalFormat dosDigitos(); hashCode(); int salida=""; int diametro en "dosDigitos(); obtenerArea(circulo); double salida=""; int circunferencia es "dosDigitos(); obtenerCircunferencia(); double salida=""; int area es "dosDigitos.forma(); obtenerDiametro(); double System.out.println(salida); String toString(); int getImiteradio(); double equals(Object obj); boolean getClass(); Class<>"
        System.out.println(salida);
        System.exit(0);
    }
}
```

OUTLINE SONARLINT RULES JAVA PROJECTS

Escribe aquí para buscar

File Edt Selection View Go Run Terminal Help

EXPLORER CIRULO

```
src > circulo > Main.java > Main > main(String[])
1 package circulo;
2
3 import java.text.DecimalFormat;
4
5 /**
6 * Pruebas de refactorización en NetBeans con la clase Circulo
7 * @author profesor
8 */
9
public class Main {
    Run | Debug
    public static void main(String[] args) {
        Circulo circulo = new Circulo(17,43,2.5);
        String salida =
            "La coordenada X es "+circulo.getCoordenadaX()+
            "La coordenada Y es "+circulo.getCoordenadaY()+
            "El radio es "+circulo.getRadio();
        DecimalFormat dosDigitos = new DecimalFormat("###.##");
        int circulo establecerX(35); int a[]; int getCoordenadaX(); int getCoordenadaY(); int establecerRadio(4,2); double establecerRadio(); double DecimalFormat dosDigitos(); hashCode(); int salida=""; int diametro en "dosDigitos(); obtenerArea(circulo); double salida=""; int circunferencia es "dosDigitos(); obtenerCircunferencia(); double salida=""; int area es "dosDigitos.forma(); obtenerDiametro(); double System.out.println(salida); String toString(); int getImiteradio(); double equals(Object obj); boolean getClass(); Class<>"
        System.out.println(salida);
        System.exit(0);
    }
}
```

OUTLINE SONARLINT RULES JAVA PROJECTS

Escribe aquí para buscar

The screenshot shows the Visual Studio Code interface with a Java project named 'Circulo'. The 'Main.java' file is open in the editor. The code defines a class 'Main' with a main method that creates a 'Circulo' object, sets its coordinates and radius, and prints its circumference and area. The code uses Java 8 features like String interpolation and try-with-resources.

```
File Edit Selection View Go Run Terminal Help
File Explorer Open Editors Circuito.java Main.java CirculoTest.java
src > circulo > Main.java > Main > main(String[])
1 package circulo;
2
3 import java.text.DecimalFormat;
4
5 /**
6 * Pruebas de refactorización en NetBeans con la clase Circulo
7 * @author profesor
8 */
9 public class Main {
10     public static void main(String[] args) {
11         Circulo circulo = new Circulo(17,43,2.5);
12         String salida =
13             "La coordenada X es "+circulo.getCoordenadaX()+
14             "La coordenada Y es "+circulo.getCoordenadaY()+
15             "El radio es "+circulo.getRadio();
16
17         circulo.setCoordenadaX(35);
18         circulo.setCoordenadaY(45);
19         DecimalFormat formatoRadio = new DecimalFormat("###.##");
20         DecimalFormat formatoCircunferencia = new DecimalFormat("###.##");
21         String salida="";
22         salida+="La circunferencia es "+formatoCircunferencia.format(circulo.obtenerCircunferencia());
23         salida+="El área es "+formatoRadio.format(circulo.obtenerAreaCirculo());
24         System.out.println(salida);
25
26         System.out.println(salida);
27         System.exit(0);
28     }
29 }
30
```

etc...

Optativo. Encapsular los tres campos del método (coordenadaX coordenadaY, radio).  
Investigar la funcionalidad de encapsular.

The screenshot shows the Visual Studio Code interface with the same Java project 'Circulo'. The 'Circulo.java' file is open, showing the class definition. The fields 'coordenadaX', 'coordenadaY', and 'radio' are now declared as private. The constructor and methods remain the same as in the previous screenshot.

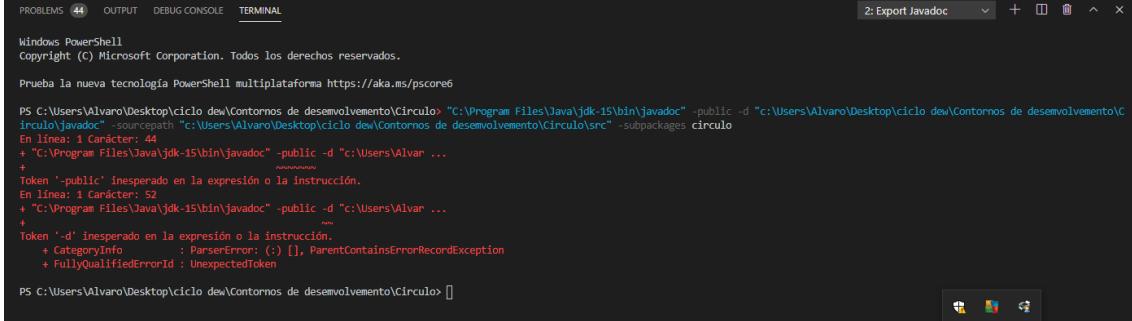
```
File Edit Selection View Go Run Terminal Help
File Explorer Open Editors Circuito.java Main.java CirculoTest.java
src > circulo > Circuito.java > Main.java > Circulo > obtenerDiametro()
1 package circulo;
2
3 import jdk.jfr.internal.PrivateAccess;
4
5 /**
6 * Pruebas de refactorización en NetBeans con la clase Circulo
7 * @author profesor
8 */
9 public class Circulo {
10     private static final double LIMITERADIO = 0.0;
11     private int CoordenadaX;
12     private int CoordenadaY;
13     private double radio;
14
15     public Circulo() {
16     }
17
18     public Circulo(int valorX, int valorY, double valorRadio) {
19         this.CoordenadaX = valorX;
20         this.CoordenadaY = valorY;
21         this.radio = valorRadio;
22     }
23
24
25     public double obtenerDiametro() {
26         return radio * 2;
27     }
28
29     public double obtenerCircunferencia() {
30         return Math.PI * obtenerDiametro();
31     }
32
33     public double obtenerAreaCirculo() {
34         return Math.PI * radio * radio;
35     }
36 }
```

Observado el programa los campos del método ya están encapsulados o denominados "private" y de este modo solo se pueden modificar dende los métodos getter y setter imposibilitando que sean accesibles desde otras clases de nuestro programa, o por otro desarrollador que use a nuestra clase modifique los parámetros de manera inadecuada.

## Ejercicio 2

Crear a documentación con javadoc.

En el visual me da este error.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

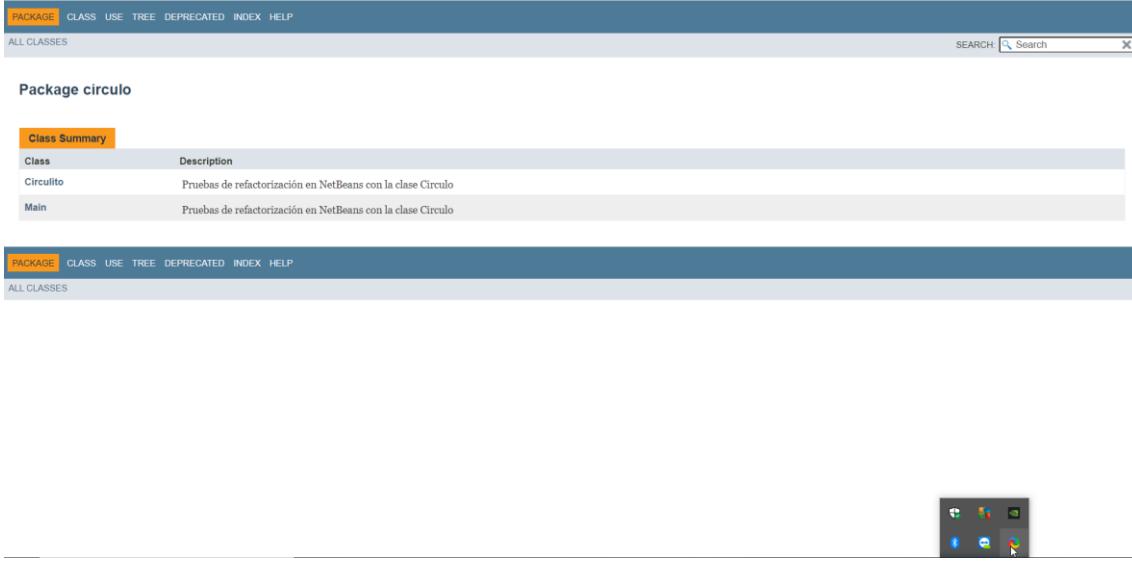
Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/powershell

PS C:\Users\Alvaro\Desktop\ciclo dev\Contornos de desarrollo\Círculo> "C:\Program Files\Java\jdk-15\bin\javadoc" -public -d "c:\Users\Alvaro\Desktop\ciclo dev\Contornos de desarrollo\Círculo" -sourcepath "c:\Users\Alvaro\Desktop\ciclo dev\Contornos de desarrollo\Círculo\src" -subpackages círculo
En linea: 1 Carácter: 44
+ "C:\Program Files\Java\jdk-15\bin\javadoc" -public -d "c:\Users\Alvaro ...
+ ~~~~~
Token '-public' inesperado en la expresión o la instrucción.
En linea: 1 Carácter: 52
+ "C:\Program Files\Java\jdk-15\bin\javadoc" -public -d "c:\Users\Alvaro ...
+ ~~~~~
Token '-d' inesperado en la expresión o la instrucción.
+ CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
+ FullyQualifiedErrorId : UnexpectedToken
PS C:\Users\Alvaro\Desktop\ciclo dev\Contornos de desarrollo\Círculo> []
```

Creamos el javadoc con el netbeans.

Probé con el netbeans y genera una página correcta como el api de java.

Se adjuntan los HTML con el proyecto.

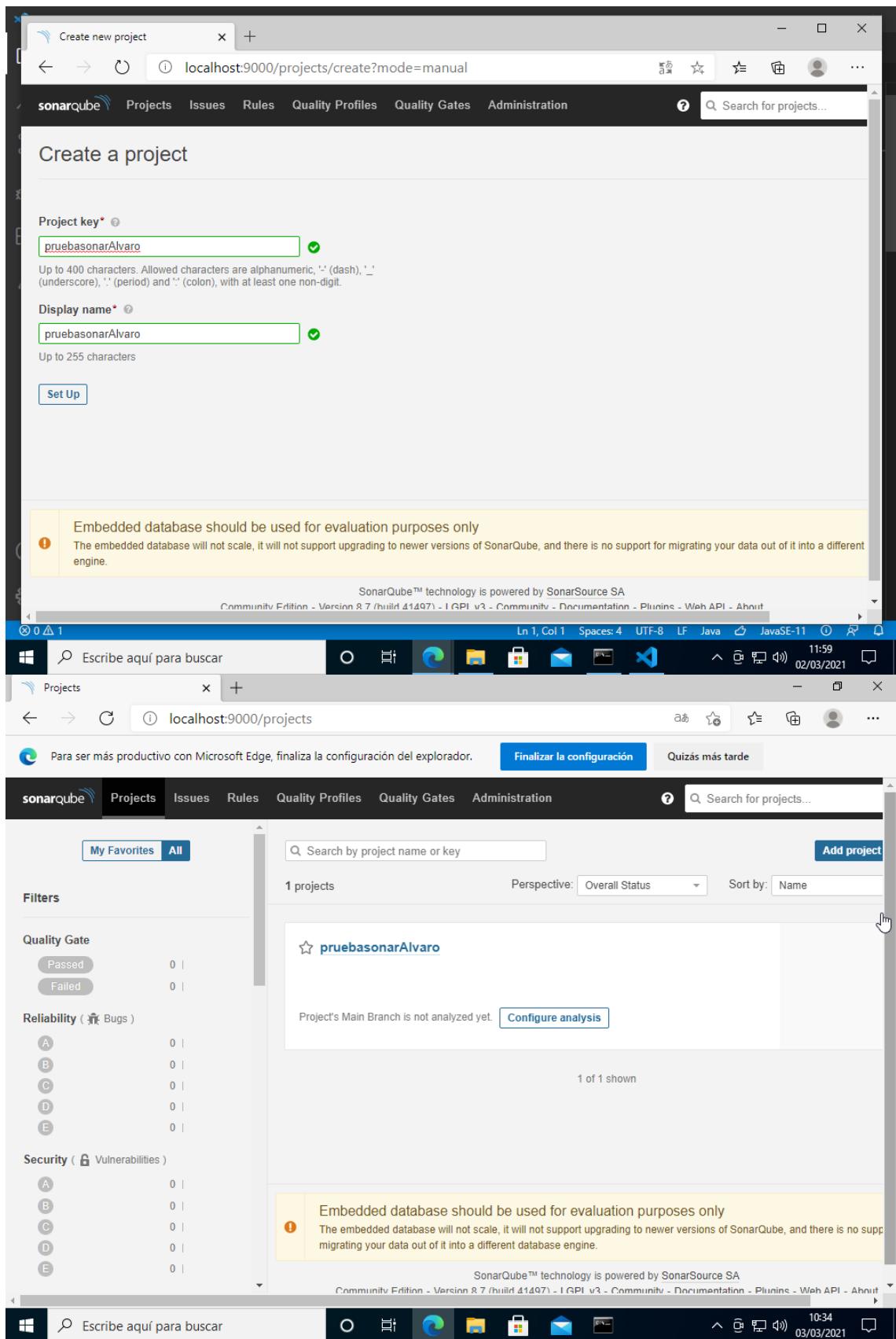


The screenshot shows the NetBeans Java API documentation interface. The top navigation bar includes links for PACKAGE, CLASS, USE, TREE, DEPRECATED, INDEX, and HELP, with PACKAGE selected. A search bar is also present. Below the navigation, there is a link to ALL CLASSES. The main content area displays a table titled "Package circulo". The table has two rows:

Class	Description
Circulito	Pruebas de refactorización en NetBeans con la clase Circulo
Main	Pruebas de refactorización en NetBeans con la clase Circulo

## Ejercicio 3

Se inicia le sonarque pero hay varios fallos que no he conseguido solucionar de momento.



The image shows two screenshots of the SonarQube interface. The top screenshot is titled 'Create a project' and shows fields for 'Project key' (pruebasonarAlvaro) and 'Display name' (pruebasonarAlvaro). A note at the bottom states: 'Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different engine.' The bottom screenshot shows the 'Projects' dashboard with one project listed: 'pruebasonarAlvaro'. It displays metrics for Quality Gate (Passed: 0, Failed: 0), Reliability (A: 0, B: 0, C: 0, D: 0, E: 0), and Security (A: 0, B: 0, C: 0, D: 0, E: 0). A note at the bottom of the dashboard is identical to the one in the creation screen.

Create new project

localhost:9000/projects/create?mode=manual

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration Search for projects...

## Create a project

Project key\*  ✓  
Up to 400 characters. Allowed characters are alphanumeric, '-' (dash), '\_' (underscore), '.' (period) and ':' (colon), with at least one non-digit.

Display name\*  ✓  
Up to 255 characters

[Set Up](#)

Embedded database should be used for evaluation purposes only  
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different engine.

SonarQube™ technology is powered by SonarSource SA  
Community Edition - Version 8.7 (build 41497) - I API v3 - Community - Documentation - Plugins - Web API - About

Projects

localhost:9000/projects

Finalizar la configuración Quizás más tarde

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration Search for projects...

My Favorites All

Filters

Search by project name or key

1 projects Perspective: Overall Status Sort by: Name

Quality Gate

Passed	0
Failed	0

Reliability ( Bugs )

A	0
B	0
C	0
D	0
E	0

Security ( Vulnerabilities )

A	0
B	0
C	0
D	0
E	0

pruebasonarAlvaro

Project's Main Branch is not analyzed yet. [Configure analysis](#)

1 of 1 shown

Embedded database should be used for evaluation purposes only  
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

SonarQube™ technology is powered by SonarSource SA  
Community Edition - Version 8.7 (build 41497) - I API v3 - Community - Documentation - Plugins - Web API - About

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Main.java - Circulo - Visual Studio Code.
- Editor:** Three tabs are open: Circulito.java, Main.java (active), and CirculoTest.java. The Main.java code is as follows:

```
src > main > circulo > Main.java > {} circulo
9  /*
10  public class Main {
11      Run | Debug
12      public static void main(String[] args) {
13          Circulito circulo = new Circulito(37,43,2.5);
14          String salida =
15              "La coordenada X es "+circulo.getCoordenadaX()+
16      }
17 }
```

- Terminal:** Shows Maven build output:

```
[INFO] Finished at: 2021-03-04T18:04:16+01:00
[INFO]
[ERROR] Failed to execute goal org.sonarsource.scanner.maven:sonar-maven-plugin:3.8.0.2131:sonar (default-cli) on project pruebaSonarAlvaro: Not authorized. Analyzing this project requires authentication. Please provide a user token in sonar.login or other credentials in sonar.login and sonar.password. -> [Help 1]
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions, please read the following articles:
[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/MojoExecutionException
-Dsonar : El término '-Dsonar' no se reconoce como nombre de un cmdlet, función, archivo de script o programa ejecutable. Compruebe si escribió correctamente el nombre o, si incluyó una ruta de acceso, compruebe que dicha ruta es correcta e intételo de nuevo.
En línea: 2 Carácter: 3
+ ~~~~~~-Dsonar.projectKey=pruebaSonarAlvaro \
+ ~~~~~~+ CategoryInfo : ObjectNotFound: (-Dsonar:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

-Dsonar : El término '-Dsonar' no se reconoce como nombre de un cmdlet, función, archivo de script o programa ejecutable. Compruebe si escribió correctamente el nombre o, si incluyó una ruta de acceso, compruebe que dicha ruta es correcta e intételo de nuevo.
En línea: 3 Carácter: 3
+ ~~~~~~-Dsonar.host.url=http://localhost:9000 \
+ ~~~~~~
```
- Bottom Bar:** Shows the number of problems (5), Java Process Console, and various icons for file operations.

Igual con el intento de pasar el sonaque con mi programa del ejercicio de selenium y junit.

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** TestCasesjava - nuevocirculo - Visual Studio Code.
- Terminal:** Shows Maven build output:

```
[INFO] -----
[INFO] BUILD FAILURE
[INFO] -----
[INFO] total time: 0.341 s
[INFO] Finished at: 2021-03-04T18:38:22+01:00
[INFO] -----
[ERROR] The goal you specified requires a project to execute but there is no POM in this directory (C:\Users\Mosquera_Rial\Desktop\nuevocirculo). Please verify you invoked Maven from the correct directory. -> [Help 1]
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions, please read the following articles:
[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/MissingProjectException
-Dsonar : El término '-Dsonar' no se reconoce como nombre de un cmdlet, función, archivo de script o programa ejecutable. Compruebe si escribió correctamente el nombre o, si incluyó una ruta de acceso, compruebe que dicha ruta es correcta e intételo de nuevo.
En línea: 2 Carácter: 3
+ ~~~~~~-Dsonar.projectKey=pruebaSonarAlvaro \
+ ~~~~~~+ CategoryInfo : ObjectNotFound: (-Dsonar:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

-Dsonar : El término '-Dsonar' no se reconoce como nombre de un cmdlet, función, archivo de script o programa ejecutable. Compruebe si escribió correctamente el nombre o, si incluyó una ruta de acceso, compruebe que dicha ruta es correcta e intételo de nuevo.
En línea: 3 Carácter: 3
+ ~~~~~~-Dsonar.host.url=http://localhost:9000 \
+ ~~~~~~+ CategoryInfo : ObjectNotFound: (-Dsonar:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

-Dsonar : El término '-Dsonar' no se reconoce como nombre de un cmdlet, función, archivo de script o programa ejecutable. Compruebe si escribió correctamente el nombre o, si incluyó una ruta de acceso, compruebe que dicha ruta es correcta e intételo de nuevo.
En línea: 4 Carácter: 3
+ ~~~~~~-Dsonar.login=8d0307d23458ede9e1d8ed3dfa5b39f0e0b3a646
```
- Bottom Bar:** Shows the number of problems (56), Java Debug Console, and various icons for file operations.