

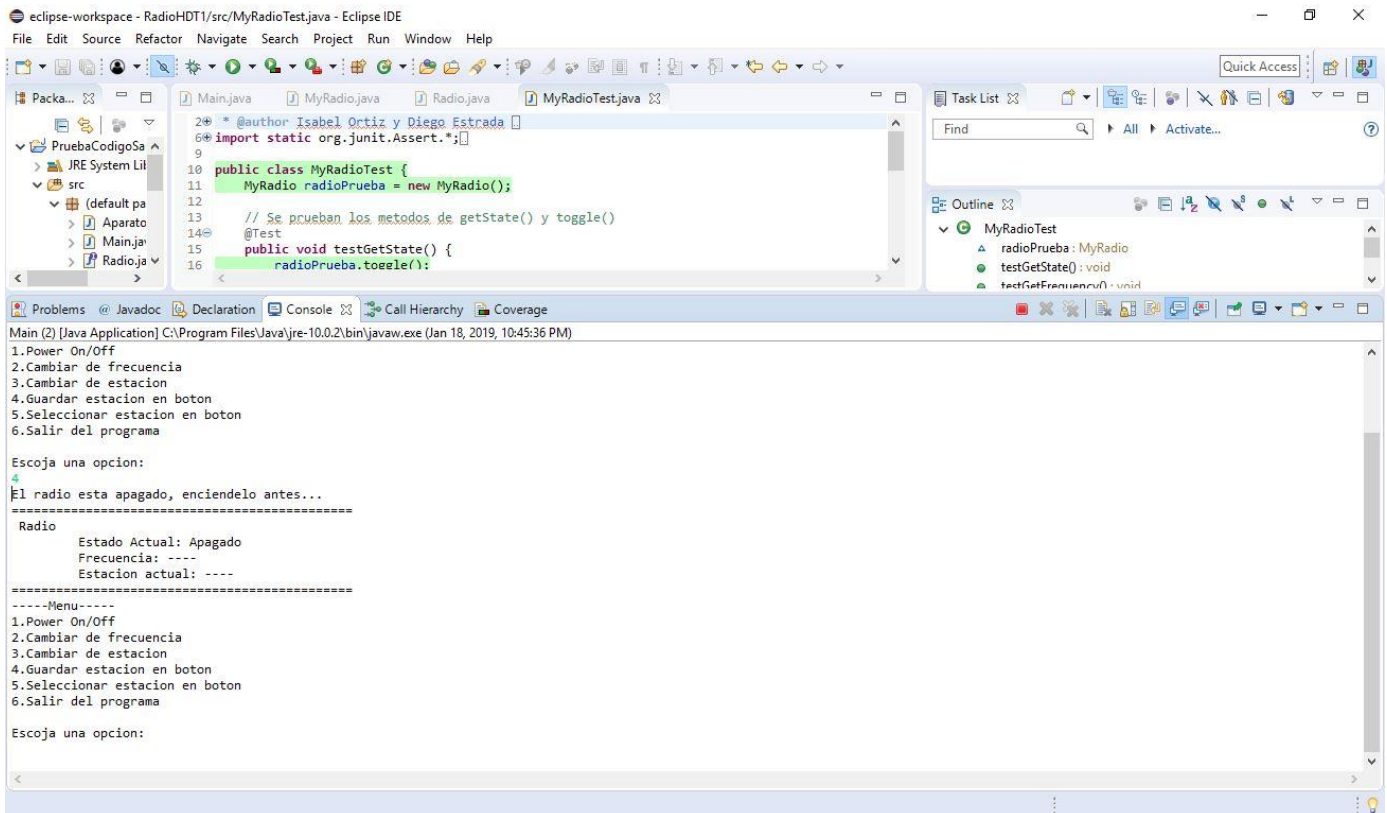
Diego Estrada – Carne 18540

María Isabel Ortiz Naranjo – Carne 18176

Hoja de trabajo 1

Funcionamiento del programa

Opción 1



The screenshot shows the Eclipse IDE interface. The top menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The toolbar contains various icons for file operations, running, and debugging. The left sidebar shows the Project Explorer with a tree view of the project structure, including a package named 'PruebaCodigoSa' and a class 'Main.java'. The central editor displays the source code of 'MyRadioTest.java', which includes a package declaration, an import statement for JUnit, and a test class with a constructor and a test method. The right sidebar shows the Outline view, which lists the classes and methods in the project. The bottom panel shows the Console view, which displays the output of the program, including a list of options, a prompt to choose an option, and the state of the radio (apagado).

```
eclipse-workspace - RadioHDT1/src/MyRadioTest.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

Packa...
PruebaCodigoSa
  JRE System Li
  src
    (default pa
      Aparato
      Main.ja
      Radio.ja

Main.java
MyRadioTest.java
2 * @author Isabel Ortiz y Diego Estrada
3 import static org.junit.Assert.*;
4
5 public class MyRadioTest {
6     MyRadio radioPrueba = new MyRadio();
7
8     // Se prueban los metodos de getState() y toggle()
9     @Test
10    public void testGetState() {
11        radioPrueba.toggle();
12    }
13
14
15
16

Task List
Find
All
Activate...

Outline
MyRadioTest
  radioPrueba: MyRadio
  testGetState(): void
  testGetFrecuencia(): void

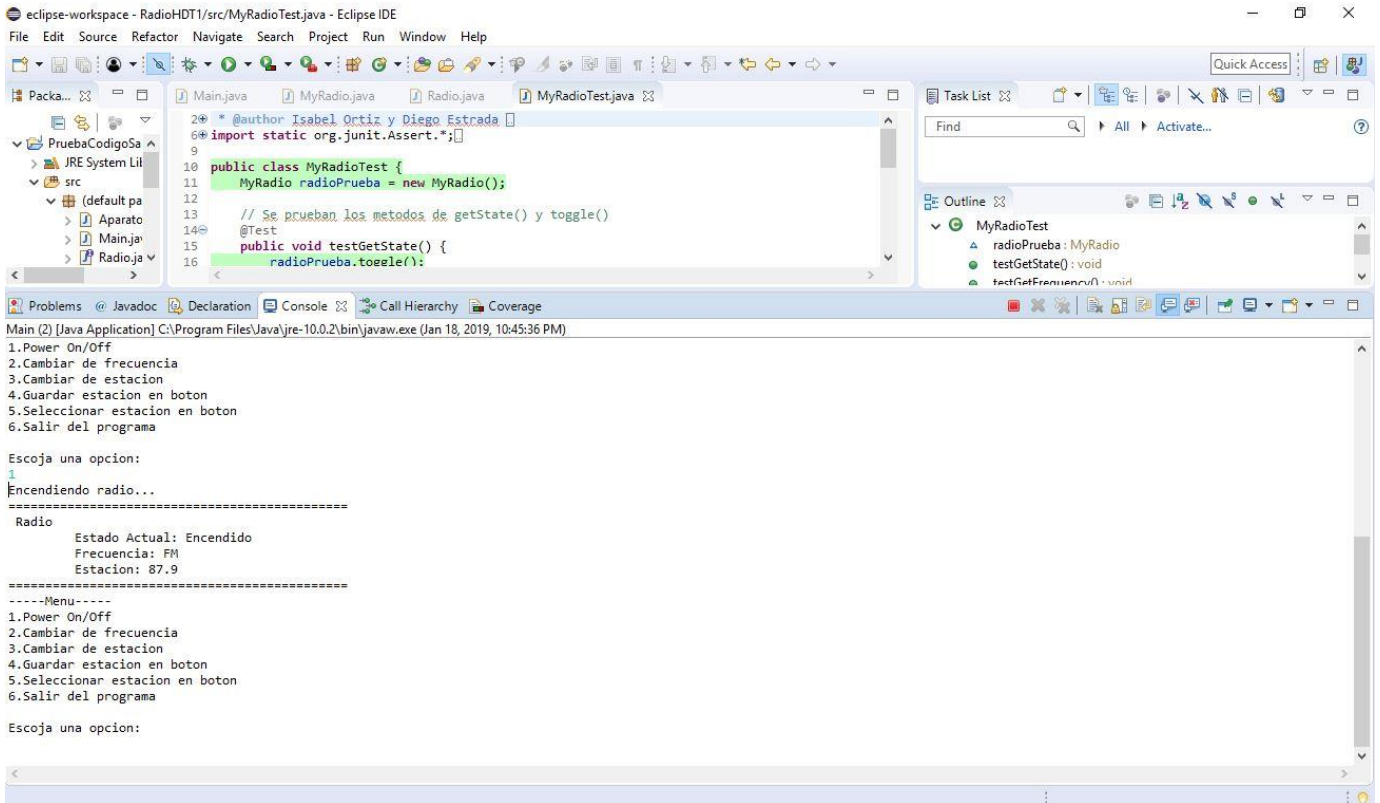
Problems
Javadoc
Declaration
Console
Call Hierarchy
Coverage

Main (2) [Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (Jan 18, 2019, 10:45:36 PM)
1.Power On/Off
2.Cambiar de frecuencia
3.Cambiar de estacion
4.Guardar estacion en boton
5.Seleccionar estacion en boton
6.Salir del programa

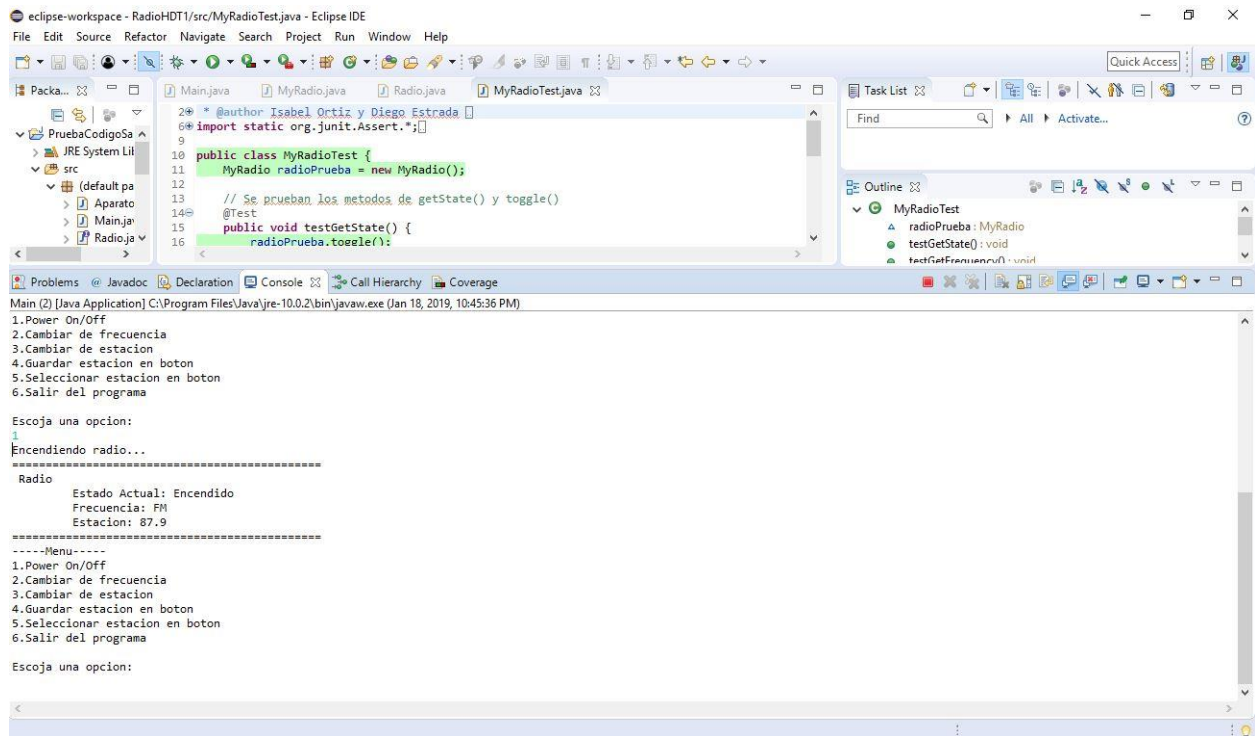
Escoja una opcion:
4
El radio esta apagado, enciendolo antes...
-----
Radio
Estado Actual: Apagado
Frecuencia: ----
Estacion actual: ----
-----
-----Menu-----
1.Power On/Off
2.Cambiar de frecuencia
3.Cambiar de estacion
4.Guardar estacion en boton
5.Seleccionar estacion en boton
6.Salir del programa

Escoja una opcion:
```

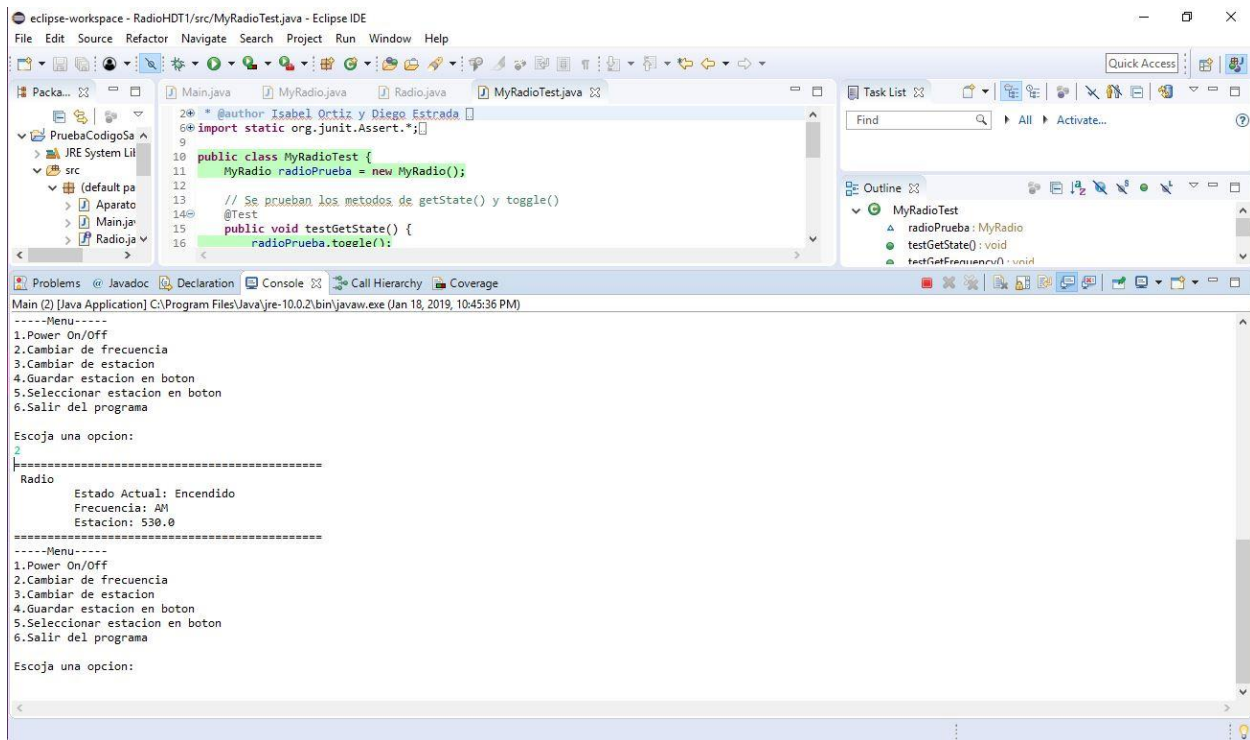
Opción 1



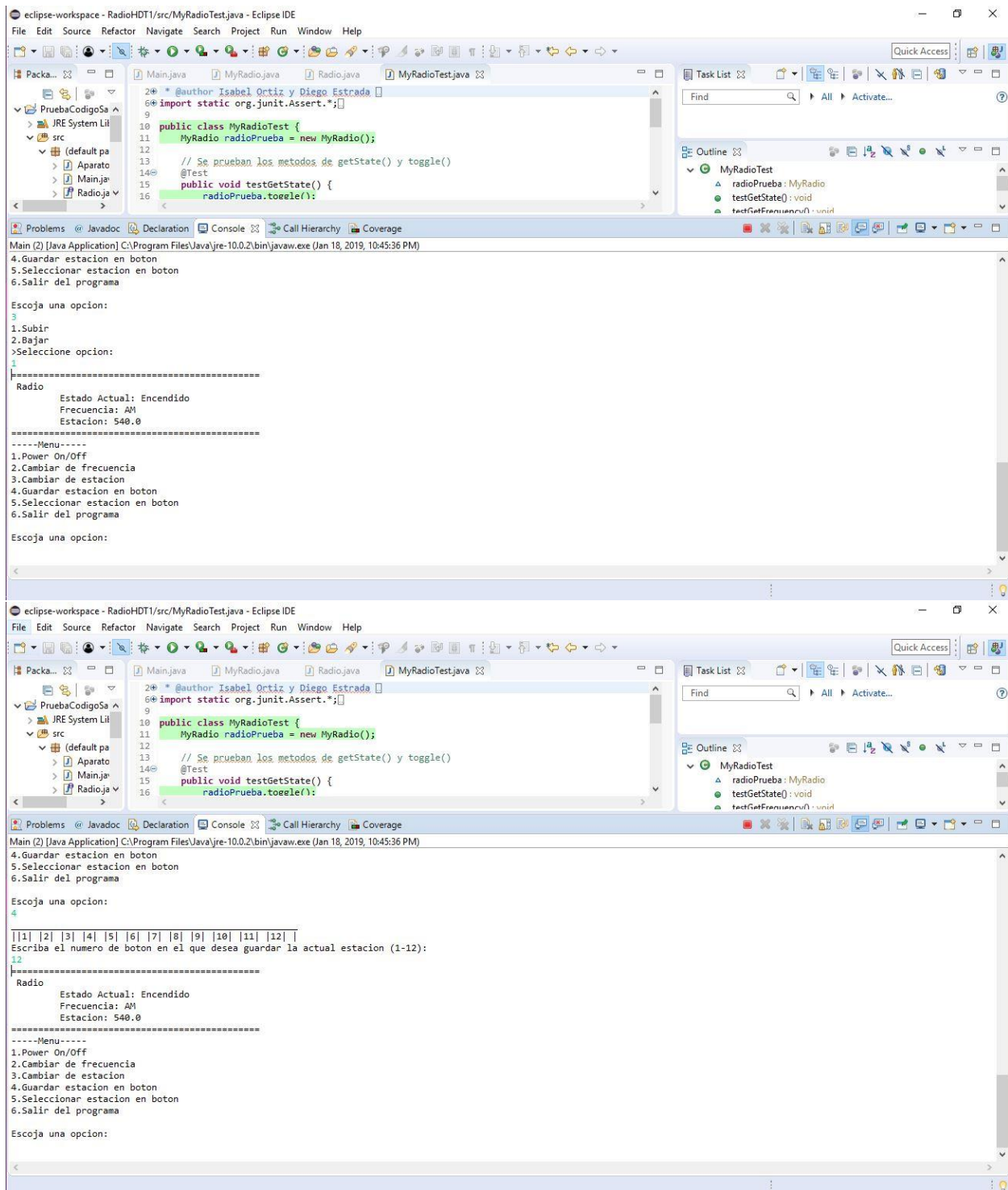
Opciones 1 y 2



Opciones 3 y 4



Opciones 3 y 4



Opciones 4 y 5

The screenshot shows the Eclipse IDE with the following components:

- Editor:** Displays the `MyRadioTest.java` file. The code includes a `MyRadio` class and a `MyRadioTest` class with a `testGetState()` method.
- Outline:** Shows the class hierarchy: `MyRadioTest` contains `radioPrueba: MyRadio`, `testGetState(): void`, and `testGetFrecuencia(): void`.
- Console:** Displays the output of the application. It shows the program running and the user selecting option 4.

```
4. Guardar estacion en boton
5. Seleccionar estacion en boton
6. Salir del programa

Escoja una opcion:
4

|1| 2| 3| 4| 5| 6| 7| 8| 9| 10| 11| 12|
Escriba el numero de boton en el que desea guardar la actual estacion (1-12):
10
-----
Radio
    Estado Actual: Encendido
    Frecuencia: AM
    Estacion: 540.0
-----
-----Menu-----
1. Power On/Off
2. Cambiar de frecuencia
3. Cambiar de estacion
4. Guardar estacion en boton
5. Seleccionar estacion en boton
6. Salir del programa

Escoja una opcion:
```

The screenshot shows the Eclipse IDE with the following components:

- Editor:** Displays the `MyRadioTest.java` file. The code includes a `MyRadio` class and a `MyRadioTest` class with a `testGetState()` method.
- Outline:** Shows the class hierarchy: `MyRadioTest` contains `radioPrueba: MyRadio`, `testGetState(): void`, and `testGetFrecuencia(): void`.
- Console:** Displays the output of the application. It shows the program running and the user selecting option 5.

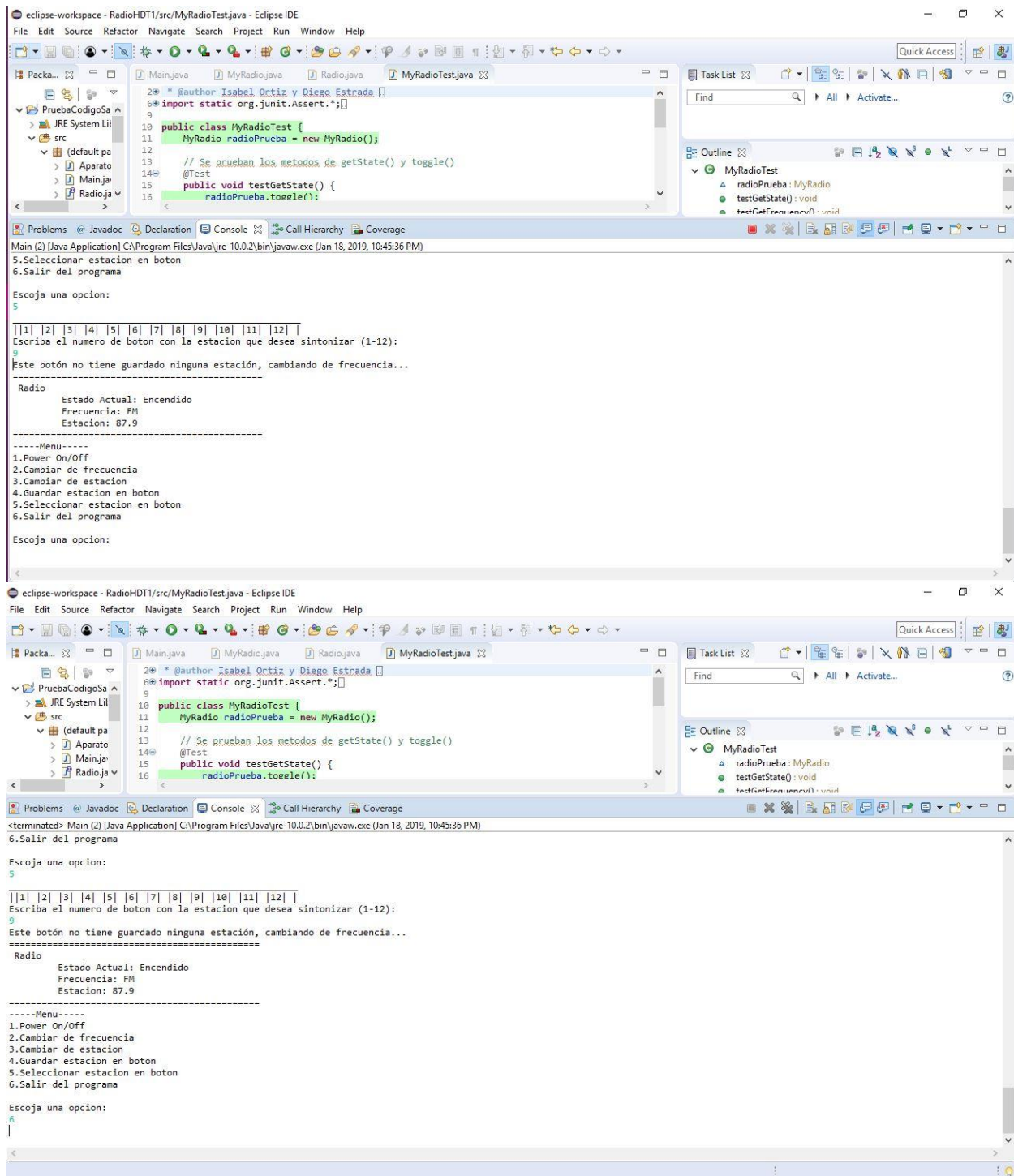
```
4. Guardar estacion en boton
5. Seleccionar estacion en boton
6. Salir del programa

Escoja una opcion:
5

|1| 2| 3| 4| 5| 6| 7| 8| 9| 10| 11| 12|
Escriba el numero de boton con la estacion que desea sintonizar (1-12):
10
-----
Radio
    Estado Actual: Encendido
    Frecuencia: AM
    Estacion: 540.0
-----
-----Menu-----
1. Power On/Off
2. Cambiar de frecuencia
3. Cambiar de estacion
4. Guardar estacion en boton
5. Seleccionar estacion en boton
6. Salir del programa

Escoja una opcion:
```


Opciones 5 y 6



Prueba con Código de Saul Contreras

The screenshot shows the Eclipse IDE with the file `AparatoRadial.java` open. The code defines a `Radio` interface and a `AparatoRadial` class that implements it. The class has private attributes for `estado`, `frequency`, `station`, `maxFM`, `minFM`, `maxAM`, and `minAM`. The console output shows the program running and displaying the current state of the radio: `Estado Actual: Encendido`, `Frecuencia: AM`, and `Estacion: 530.0`. A menu is displayed with options: 1. Power On/Off, 2. Cambiar de frecuencia, 3. Cambiar de estacion, 4. Guardar estacion en boton, 5. Seleccionar estacion en boton, and 6. Salir del programa. The user is prompted to choose an option.

```
10 /*
11 */
12 * Esta clase implementa la interfaz Radio.
13 */
14 public class AparatoRadial implements Radio {
15
16     /**
17      * El estado de la radio. ON/OFF
18      */
19     private boolean estado;
20
21     /**
22      * El tipo de frecuencia en el que se encuentra AM/FM
23      */
24     private boolean frequency;
25
26     /**
27      * La estacion en la que se encuentra
28      */
29     private double station;
30
31     /**
32      * Es la estación máxima que puede alcanzar el FM
33      */
34     private double maxFM;
35
36     /**
37      * Es la estación mínima que puede alcanzar el FM
38      */
39     private double minFM;
40
41     /**
42      * Es la estación máxima que puede alcanzar el AM
43      */
44     private double maxAM;
45
46     /**
47      * Es la estación mínima que puede alcanzar el AM
48      */
49 }
```

Console Output:

```
Main (2) [Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (Jan 18, 2019, 10:35:15 PM)
6.Salir del programa

Escoja una opcion:
2
Radio
Estado Actual: Encendido
Frecuencia: AM
Estacion: 530.0

-----Menu-----
1.Power On/Off
2.Cambiar de frecuencia
3.Cambiar de estacion
4.Guardar estacion en boton
5.Seleccionar estacion en boton
6.Salir del programa

Escoja una opcion:
```

The screenshot shows the Eclipse IDE with the file `AparatoRadial.java` open. The code is the same as in the previous screenshot. The console output shows the program running and displaying the current state of the radio: `Estado Actual: Encendido`, `Frecuencia: FM`, and `Estacion: 87.9`. A menu is displayed with options: 1. Power On/Off, 2. Cambiar de frecuencia, 3. Cambiar de estacion, 4. Guardar estacion en boton, 5. Seleccionar estacion en boton, and 6. Salir del programa. The user is prompted to choose an option. The user enters '5', and the program displays a list of 12 radio stations for selection. The user enters '12', and the program displays the current state of the radio: `Estado Actual: Encendido`, `Frecuencia: FM`, and `Estacion: 87.9`. A menu is displayed with options: 1. Power On/Off, 2. Cambiar de frecuencia, 3. Cambiar de estacion, 4. Guardar estacion en boton, 5. Seleccionar estacion en boton, and 6. Salir del programa. The user is prompted to choose an option.

```
10 /*
11 */
12 * Esta clase implementa la interfaz Radio.
13 */
14 public class AparatoRadial implements Radio {
15
16     /**
17      * El estado de la radio. ON/OFF
18      */
19     private boolean estado;
20
21     /**
22      * El tipo de frecuencia en el que se encuentra AM/FM
23      */
24     private boolean frequency;
25
26     /**
27      * La estacion en la que se encuentra
28      */
29     private double station;
30
31     /**
32      * Es la estación máxima que puede alcanzar el FM
33      */
34     private double maxFM;
35
36     /**
37      * Es la estación mínima que puede alcanzar el FM
38      */
39     private double minFM;
40
41     /**
42      * Es la estación máxima que puede alcanzar el AM
43      */
44     private double maxAM;
45
46     /**
47      * Es la estación mínima que puede alcanzar el AM
48      */
49 }
```

Console Output:

```
Main (2) [Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (Jan 18, 2019, 10:37:29 PM)
11 | 12 |
Escriba el numero de boton en el que desea guardar la actual estacion (1-12):
12
Radio
Estado Actual: Encendido
Frecuencia: FM
Estacion: 87.9

-----Menu-----
1.Power On/Off
2.Cambiar de frecuencia
3.Cambiar de estacion
4.Guardar estacion en boton
5.Seleccionar estacion en boton
6.Salir del programa

Escoja una opcion:
5
11 | 12 |
Escriba el numero de boton con la estacion que desea sintonizar (1-12):
12
Radio
Estado Actual: Encendido
Frecuencia: FM
Estacion: 87.9

-----Menu-----
1.Power On/Off
2.Cambiar de frecuencia
3.Cambiar de estacion
4.Guardar estacion en boton
5.Seleccionar estacion en boton
6.Salir del programa

Escoja una opcion:
```


eclipse-workspace - PruebaCodigoSaul/src/AparatoRadial.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Main.java AparatoRadial.java Radio.java

```
1 /*
2 AparatoRadial.java
3 java version 1.8.0_191
4 18/01/2018
5 Saul Contreras Godoy 18409
6 Radio Moja de trabajo 1
7 Algoritmos y estructura de datos
8 Clase aparato radial que implementa la Interfaz Radio
9 */
10
11 /**
12 * Esta clase implementa la interfaz Radio.
13 */
14 public class AparatoRadial implements Radio {
15
16     /**
17      * El estado de la radio. ON/OFF
18      */
19     private boolean estado;
20
21     /**
22      * El tipo de frecuencia en el que se encuentra AM/FM
23      */
24     private boolean frequency;
25
26     /**
27      * La estacion en la que se encuentra
28      */
29     private double station;
30
31     /**
32      * Es la estación máxima que puede alcanzar el FM
33      */
34     private double maxFM;
35
36     /**
37      * Es la estación mínima que puede alcanzar el AM
38      */
39     private double minFM;
40
41     /**
42      * Es la estación mínima que puede alcanzar el AM
43      */
44     private double maxAM;
45 }
```

Problems Javadoc Declaration Console Call Hierarchy Coverage

Main (2) [Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (Jan 18, 2019, 10:37:29 PM)

Radio

Estado Actual: Encendido
Frecuencia: FM
Estacion: 87.9

-----Menu-----
1.Power On/Off
2.Cambiar de frecuencia
3.Cambiar de estacion
4.Guardar estacion en boton
5.Seleccionar estacion en boton
6.Salir del programa

Escoja una opcion:
4

|1| 2| 3| 4| 5| 6| 7| 8| 9| 10| 11| 12| |
Escriba el numero de boton en el que desea guardar la actual estacion (1-12):
11

Radio

Estado Actual: Encendido
Frecuencia: FM
Estacion: 87.9

-----Menu-----
1.Power On/Off
2.Cambiar de frecuencia
3.Cambiar de estacion
4.Guardar estacion en boton
5.Seleccionar estacion en boton
6.Salir del programa

Escoja una opcion:
5

|1| 2| 3| 4| 5| 6| 7| 8| 9| 10| 11| 12| |
Escriba el numero de boton con la estacion que desea sintonizar (1-12):

eclipse-workspace - PruebaCodigoSaul/src/AparatoRadial.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Main.java AparatoRadial.java Radio.java

```
1 /*
2 AparatoRadial.java
3 java version 1.8.0_191
4 18/01/2018
5 Saul Contreras Godoy 18409
6 Radio Moja de trabajo 1
7 Algoritmos y estructura de datos
8 Clase aparato radial que implementa la Interfaz Radio
9 */
10
11 /**
12 * Esta clase implementa la interfaz Radio.
13 */
14 public class AparatoRadial implements Radio {
15
16     /**
17      * El estado de la radio. ON/OFF
18      */
19     private boolean estado;
20
21     /**
22      * El tipo de frecuencia en el que se encuentra AM/FM
23      */
24     private boolean frequency;
25
26     /**
27      * La estacion en la que se encuentra
28      */
29     private double station;
30
31     /**
32      * Es la estación máxima que puede alcanzar el FM
33      */
34     private double maxFM;
35
36     /**
37      * Es la estación mínima que puede alcanzar el FM
38      */
39     private double minFM;
40
41     /**
42      * Es la estación mínima que puede alcanzar el AM
43      */
44     private double maxAM;
45 }
```

Problems Javadoc Declaration Console Call Hierarchy Coverage

Main (2) [Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (Jan 18, 2019, 10:37:29 PM)

|1| 2| 3| 4| 5| 6| 7| 8| 9| 10| 11| 12| |
Escriba el numero de boton en el que desea guardar la actual estacion (1-12):
11

Radio

Estado Actual: Encendido
Frecuencia: FM
Estacion: 87.9

-----Menu-----
1.Power On/Off
2.Cambiar de frecuencia
3.Cambiar de estacion
4.Guardar estacion en boton
5.Seleccionar estacion en boton
6.Salir del programa

Escoja una opcion:
5

|1| 2| 3| 4| 5| 6| 7| 8| 9| 10| 11| 12| |
Escriba el numero de boton con la estacion que desea sintonizar (1-12):
6

Este boton no tiene guardado ninguna estación, cambiando de frecuencia...

Radio

Estado Actual: Encendido
Frecuencia: AM
Estacion: 530.0

-----Menu-----
1.Power On/Off
2.Cambiar de frecuencia
3.Cambiar de estacion
4.Guardar estacion en boton
5.Seleccionar estacion en boton
6.Salir del programa

Escoja una opcion:

eclipse-workspace - PruebaCodigoSaul/src/AparatoRadial.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Main.java AparatoRadial.java Radio.java

```
10 /*
11 AparatoRadial.java
12 java version 1.8.0_191
13 18/01/2018
14 Saul Contreras Godoy 18409
15 Radio Moja de trabajo 1
16 Algoritmos y estructura de datos
17 Clase aparato radial que implementa la Interfaz Radio
18 */
19
20 /**
21 * Esta clase implementa la interfaz Radio.
22 */
23 public class AparatoRadial implements Radio {
24
25     /**
26     * El estado de la radio. ON/OFF
27     */
28     private boolean estado;
29
30     /**
31     * El tipo de frecuencia en el que se encuentra AM/FM
32     */
33     private boolean frequency;
34
35     /**
36     * La estación en la que se encuentra
37     */
38     private double station;
39
40     /**
41     * Es la estación máxima que puede alcanzar el FM
42     */
43     private double maxFM;
44
45     /**
46     * Es la estación mínima que puede alcanzar el AM
47     */
48     private double minFM;
49
50     /**
51     * Es la estación máxima que puede alcanzar el AM
52     */
53     private double maxAM;
54
55     /**
56     * Es la estación mínima que puede alcanzar el AM
57     */
58 }
```

Problems Javadoc Declaration Console Call Hierarchy Coverage

Main (2) [Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (Jan 18, 2019, 10:39:58 PM)

```
6.Salir del programa
4
Escoja una opcion:
El radio esta apagado, enciendolo antes...
Radio
Estado Actual: Apagado
Frecuencia: ----
Estacion actual: ----
-----Menu-----
1.Power On/Off
2.Cambiar de frecuencia
3.Cambiar de estacion
4.Guardar estacion en boton
5.Seleccionar estacion en boton
6.Salir del programa
Escoja una opcion:
2
El radio esta apagado, enciendolo antes...
Radio
Estado Actual: Apagado
Frecuencia: ----
Estacion actual: ----
-----Menu-----
1.Power On/Off
2.Cambiar de frecuencia
3.Cambiar de estacion
4.Guardar estacion en boton
5.Seleccionar estacion en boton
6.Salir del programa
Escoja una opcion:
1
```

eclipse-workspace - PruebaCodigoSaul/src/AparatoRadial.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

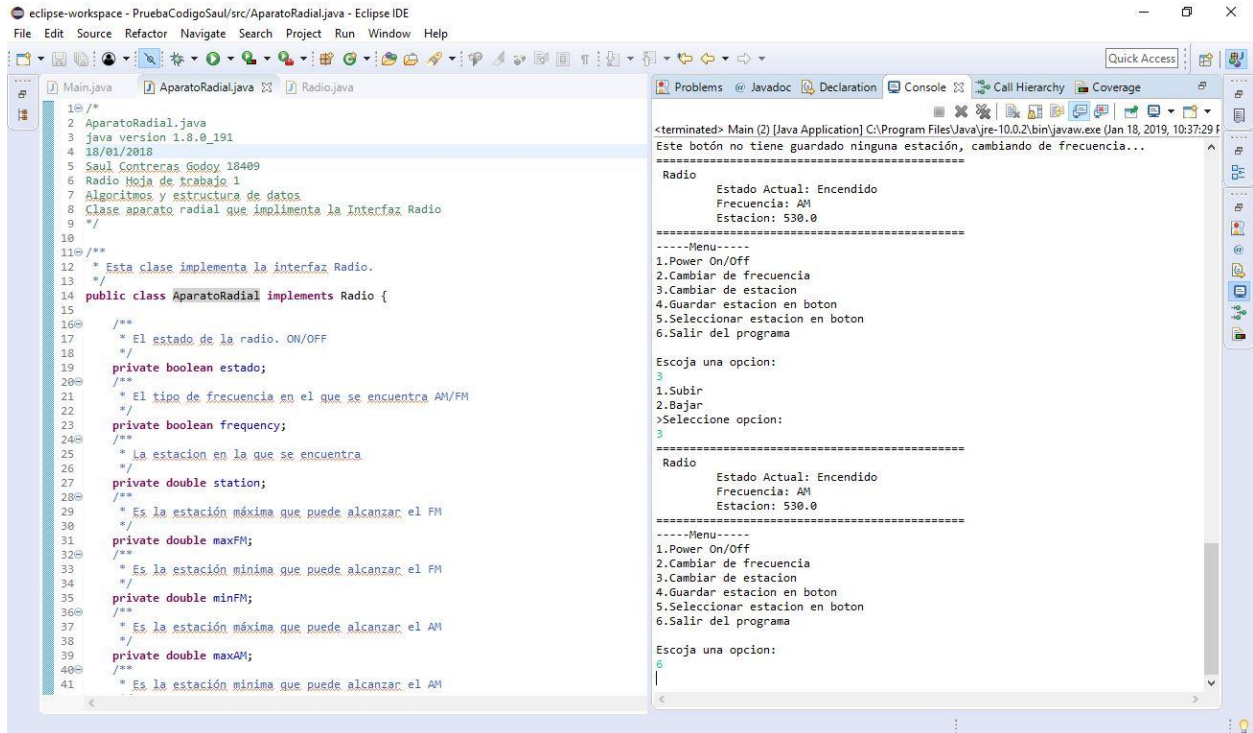
Main.java AparatoRadial.java Radio.java

```
10 /*
11 AparatoRadial.java
12 java version 1.8.0_191
13 18/01/2018
14 Saul Contreras Godoy 18409
15 Radio Moja de trabajo 1
16 Algoritmos y estructura de datos
17 Clase aparato radial que implementa la Interfaz Radio
18 */
19
20 /**
21 * Esta clase implementa la interfaz Radio.
22 */
23 public class AparatoRadial implements Radio {
24
25     /**
26     * El estado de la radio. ON/OFF
27     */
28     private boolean estado;
29
30     /**
31     * El tipo de frecuencia en el que se encuentra AM/FM
32     */
33     private boolean frequency;
34
35     /**
36     * La estación en la que se encuentra
37     */
38     private double station;
39
40     /**
41     * Es la estación máxima que puede alcanzar el FM
42     */
43     private double maxFM;
44
45     /**
46     * Es la estación mínima que puede alcanzar el FM
47     */
48     private double minFM;
49
50     /**
51     * Es la estación máxima que puede alcanzar el AM
52     */
53     private double maxAM;
54
55     /**
56     * Es la estación mínima que puede alcanzar el AM
57     */
58 }
```

Problems Javadoc Declaration Console Call Hierarchy Coverage

Main (2) [Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (Jan 18, 2019, 10:39:58 PM)

```
Encendiendo radio...
Radio
Estado Actual: Encendido
Frecuencia: FM
Estacion: 87.9
-----Menu-----
1.Power On/Off
2.Cambiar de frecuencia
3.Cambiar de estacion
4.Guardar estacion en boton
5.Seleccionar estacion en boton
6.Salir del programa
Escoja una opcion:
5
|1| 2| 3| 4| 5| 6| 7| 8| 9| 10| 11| 12|
Escriba el numero de boton con la estacion que desea sintonizar (1-12):
12
Este boton no tiene guardado ninguna estación, cambiando de frecuencia...
Radio
Estado Actual: Encendido
Frecuencia: AM
Estacion: 530.0
-----Menu-----
1.Power On/Off
2.Cambiar de frecuencia
3.Cambiar de estacion
4.Guardar estacion en boton
5.Seleccionar estacion en boton
6.Salir del programa
Escoja una opcion:
```



Pruebas con Junit – 3 pruebas

