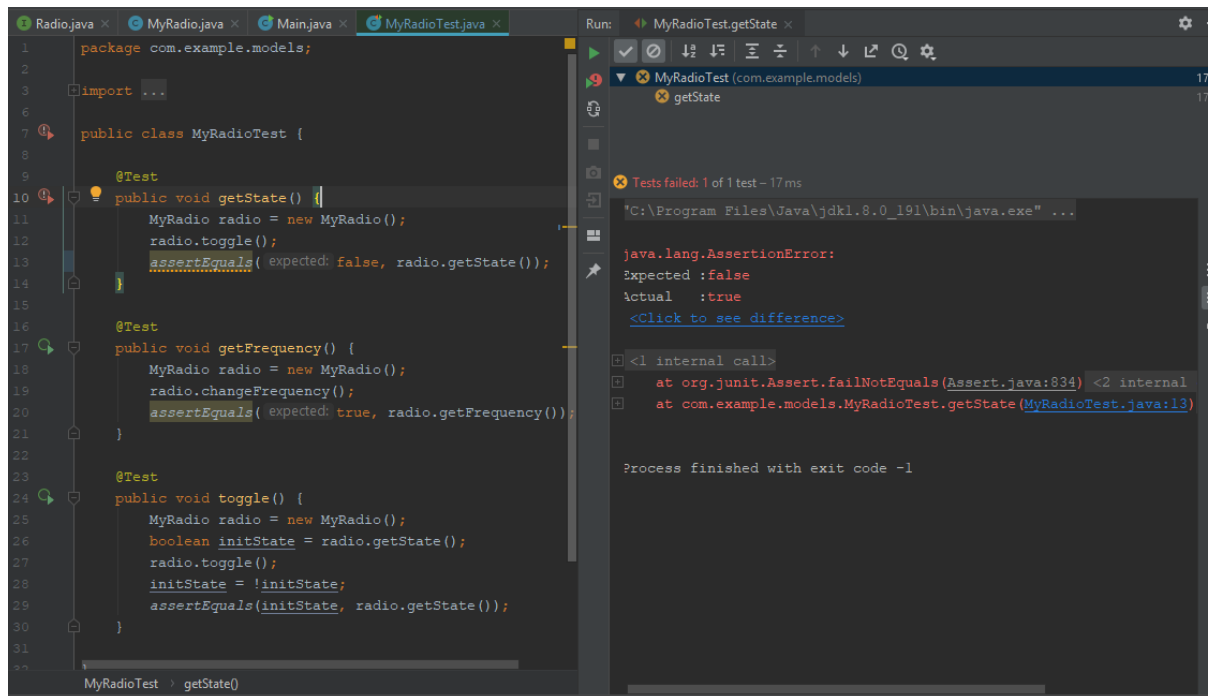


Capturas de Pantalla

1.1 Capturas de pantalla de pruebas unitarias con Junit:

1.1.1 Prueba Unitaria para método getState() fallida



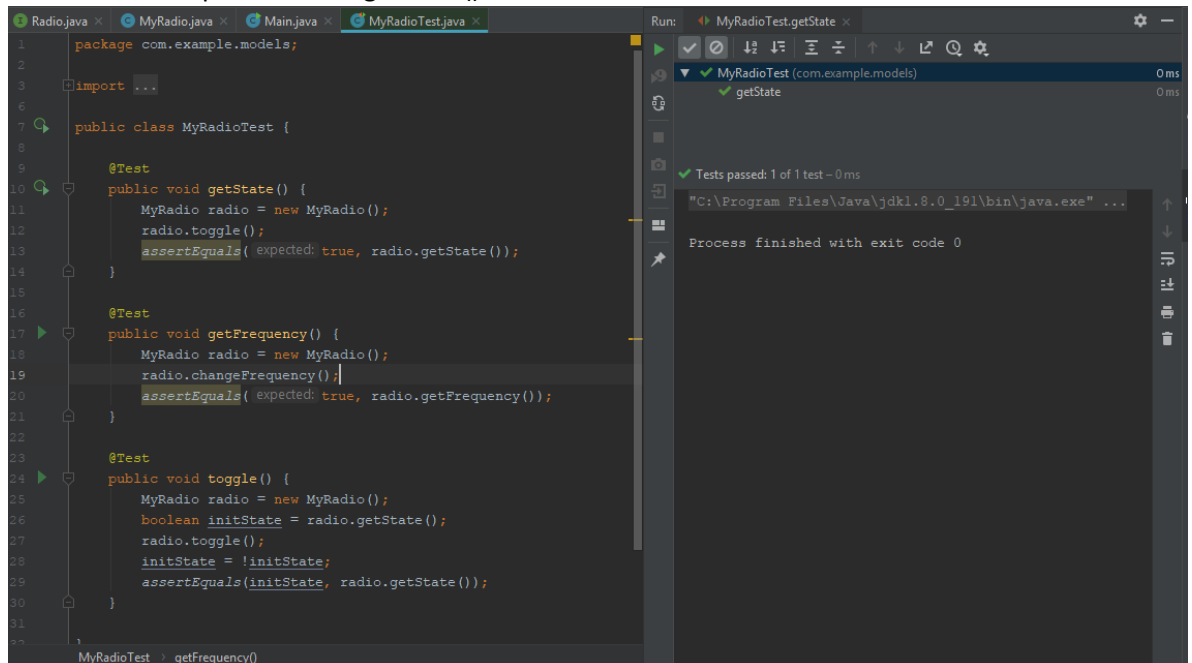
The screenshot shows an IDE with the following code in `MyRadioTest.java`:

```
1 package com.example.models;
2
3 import ...
4
5
6
7 public class MyRadioTest {
8
9
10 @Test
11 public void getState() {
12     MyRadio radio = new MyRadio();
13     radio.toggle();
14     assertEquals("expected: false, radio.getState()");
15 }
16
17 @Test
18 public void getFrequency() {
19     MyRadio radio = new MyRadio();
20     radio.changeFrequency();
21     assertEquals("expected: true, radio.getFrequency()");
22 }
23
24 @Test
25 public void toggle() {
26     MyRadio radio = new MyRadio();
27     boolean initState = radio.getState();
28     radio.toggle();
29     initState = !initState;
30     assertEquals("initState, radio.getState()");
31 }
32 }
```

The Run window shows the test results for `MyRadioTest.getState()`:

```
Run: MyRadioTest.getState
MyRadioTest (com.example.models)
getState
Tests failed: 1 of 1 test - 17ms
"C:\Program Files\Java\jdk1.8.0_191\bin\java.exe" ...
java.lang.AssertionError:
Expected :false
Actual   :true
<Click to see difference>
<1 internal call>
at org.junit.Assert.failNotEquals(Assert.java:834) <2 internal
at com.example.models.MyRadioTest.getState(MyRadioTest.java:13)
Process finished with exit code -1
```

1.1.2 Prueba Unitaria para método getState() exitosa



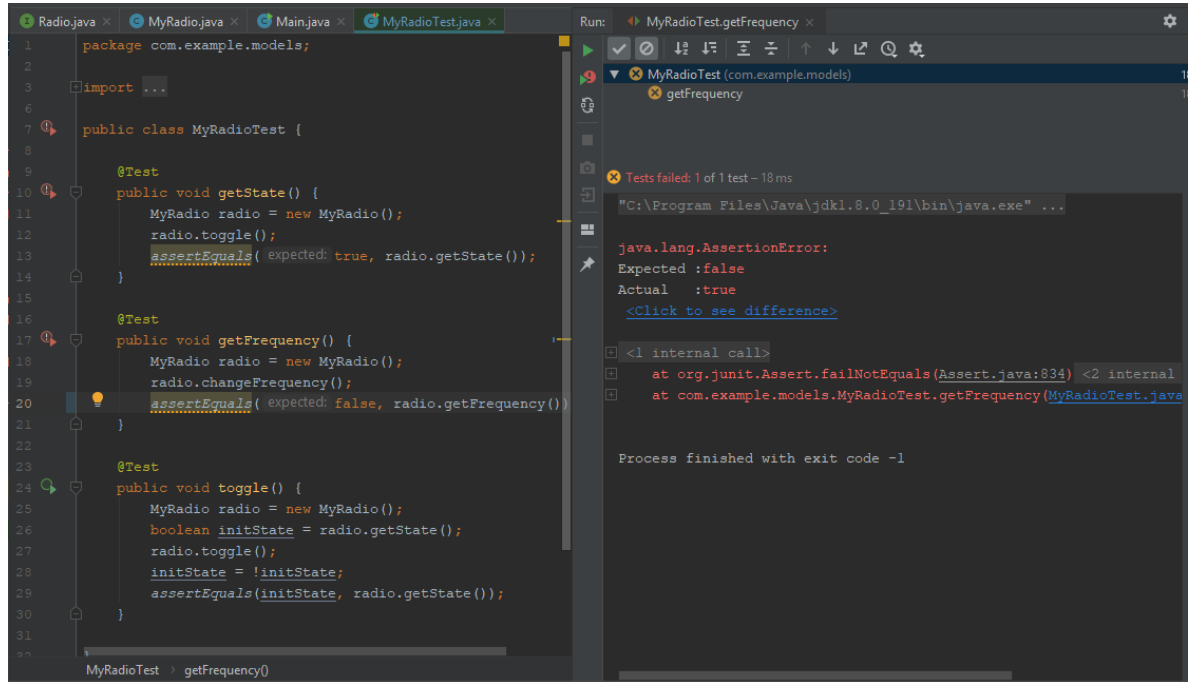
The screenshot shows the same IDE with the following code in `MyRadioTest.java`:

```
1 package com.example.models;
2
3 import ...
4
5
6
7 public class MyRadioTest {
8
9
10 @Test
11 public void getState() {
12     MyRadio radio = new MyRadio();
13     radio.toggle();
14     assertEquals("expected: true, radio.getState()");
15 }
16
17 @Test
18 public void getFrequency() {
19     MyRadio radio = new MyRadio();
20     radio.changeFrequency();
21     assertEquals("expected: true, radio.getFrequency()");
22 }
23
24 @Test
25 public void toggle() {
26     MyRadio radio = new MyRadio();
27     boolean initState = radio.getState();
28     radio.toggle();
29     initState = !initState;
30     assertEquals("initState, radio.getState()");
31 }
32 }
```

The Run window shows the test results for `MyRadioTest.getState()`:

```
Run: MyRadioTest.getState
MyRadioTest (com.example.models)
0ms
getState
0ms
Tests passed: 1 of 1 test - 0ms
"C:\Program Files\Java\jdk1.8.0_191\bin\java.exe" ...
Process finished with exit code 0
```

1.1.3 Prueba Unitaria para método getFrequency() fallida



```
1 package com.example.models;
2
3 import ...
4
5
6
7 public class MyRadioTest {
8
9     @Test
10    public void getState() {
11        MyRadio radio = new MyRadio();
12        radio.toggle();
13        assertEquals( expected: true, radio.getState());
14    }
15
16    @Test
17    public void getFrequency() {
18        MyRadio radio = new MyRadio();
19        radio.changeFrequency();
20        assertEquals( expected: false, radio.getFrequency());
21    }
22
23    @Test
24    public void toggle() {
25        MyRadio radio = new MyRadio();
26        boolean initState = radio.getState();
27        radio.toggle();
28        initState = !initState;
29        assertEquals( initState, radio.getState());
30    }
31
32 }
```

Run: MyRadioTest.getFrequency

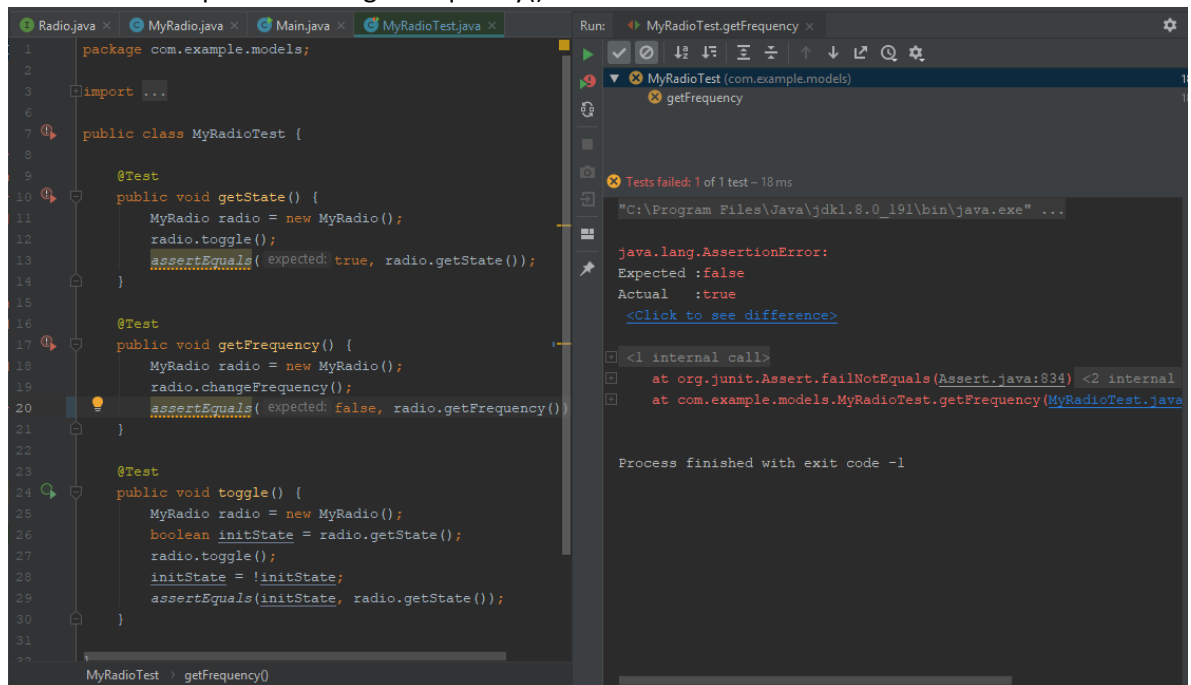
Tests failed: 1 of 1 test - 18 ms

java.lang.AssertionError:
Expected :false
Actual :true
<Click to see difference>

<1 internal call>
at org.junit.Assert.failNotEquals(Assert.java:834) <2 internal
at com.example.models.MyRadioTest.getFrequency(MyRadioTest.java

Process finished with exit code -1

1.1.4 Prueba Unitaria para método getFrequency() exitosa



```
1 package com.example.models;
2
3 import ...
4
5
6
7 public class MyRadioTest {
8
9     @Test
10    public void getState() {
11        MyRadio radio = new MyRadio();
12        radio.toggle();
13        assertEquals( expected: true, radio.getState());
14    }
15
16    @Test
17    public void getFrequency() {
18        MyRadio radio = new MyRadio();
19        radio.changeFrequency();
20        assertEquals( expected: false, radio.getFrequency());
21    }
22
23    @Test
24    public void toggle() {
25        MyRadio radio = new MyRadio();
26        boolean initState = radio.getState();
27        radio.toggle();
28        initState = !initState;
29        assertEquals( initState, radio.getState());
30    }
31
32 }
```

Run: MyRadioTest.getFrequency

Tests failed: 1 of 1 test - 18 ms

java.lang.AssertionError:
Expected :false
Actual :true
<Click to see difference>

<1 internal call>
at org.junit.Assert.failNotEquals(Assert.java:834) <2 internal
at com.example.models.MyRadioTest.getFrequency(MyRadioTest.java

Process finished with exit code -1

1.1.5 Prueba Unitaria para método toggle() fallida



```
public class MyRadioTest {  
    @Test  
    public void getState() {  
        MyRadio radio = new MyRadio();  
        radio.toggle();  
        assertEquals( expected: true, radio.getState());  
    }  
  
    @Test  
    public void getFrequency() {  
        MyRadio radio = new MyRadio();  
        radio.changeFrequency();  
        assertEquals( expected: true, radio.getFrequency());  
    }  
  
    @Test  
    public void toggle() {  
        MyRadio radio = new MyRadio();  
        boolean initState = radio.getState();  
        radio.toggle();  
        assertEquals( initState, radio.getState());  
    }  
}
```

Run: MyRadioTest.toggle
MyRadioTest (com.example.models)
toggle
Tests failed: 1 of 1 test - 33 ms
java.lang.AssertionError:
Expected :false
Actual :true
Process finished with exit code -1

1.1.6 Prueba Unitaria para método toggle() exitosa

```
import ...  
  
public class MyRadioTest {  
    @Test  
    public void getState() {  
        MyRadio radio = new MyRadio();  
        radio.toggle();  
        assertEquals( expected: true, radio.getState());  
    }  
  
    @Test  
    public void getFrequency() {  
        MyRadio radio = new MyRadio();  
        radio.changeFrequency();  
        assertEquals( expected: true, radio.getFrequency());  
    }  
  
    @Test  
    public void toggle() {  
        MyRadio radio = new MyRadio();  
        boolean initState = radio.getState();  
        radio.toggle();  
        initState = !initState;  
        assertEquals( initState, radio.getState());  
    }  
}
```

Run: MyRadioTest.toggle
MyRadioTest (com.example.models)
toggle
Tests passed: 1 of 1 test - 13 ms
Process finished with exit code 0

1.2 Capturas de Pantalla de programa funcionando con otra clase que implemente la Interfaz Radio

1.2.1 Grupo 15 conformado por Luis Quezada y André Toledo



```
Run: Main x Main x
"C:\Program Files\Java\jdk1.8.0_171\bin\java.exe" ...

-----
Radio: (OFF)
  Frecuencia: FM
  Emisora actual: 530.0

-----

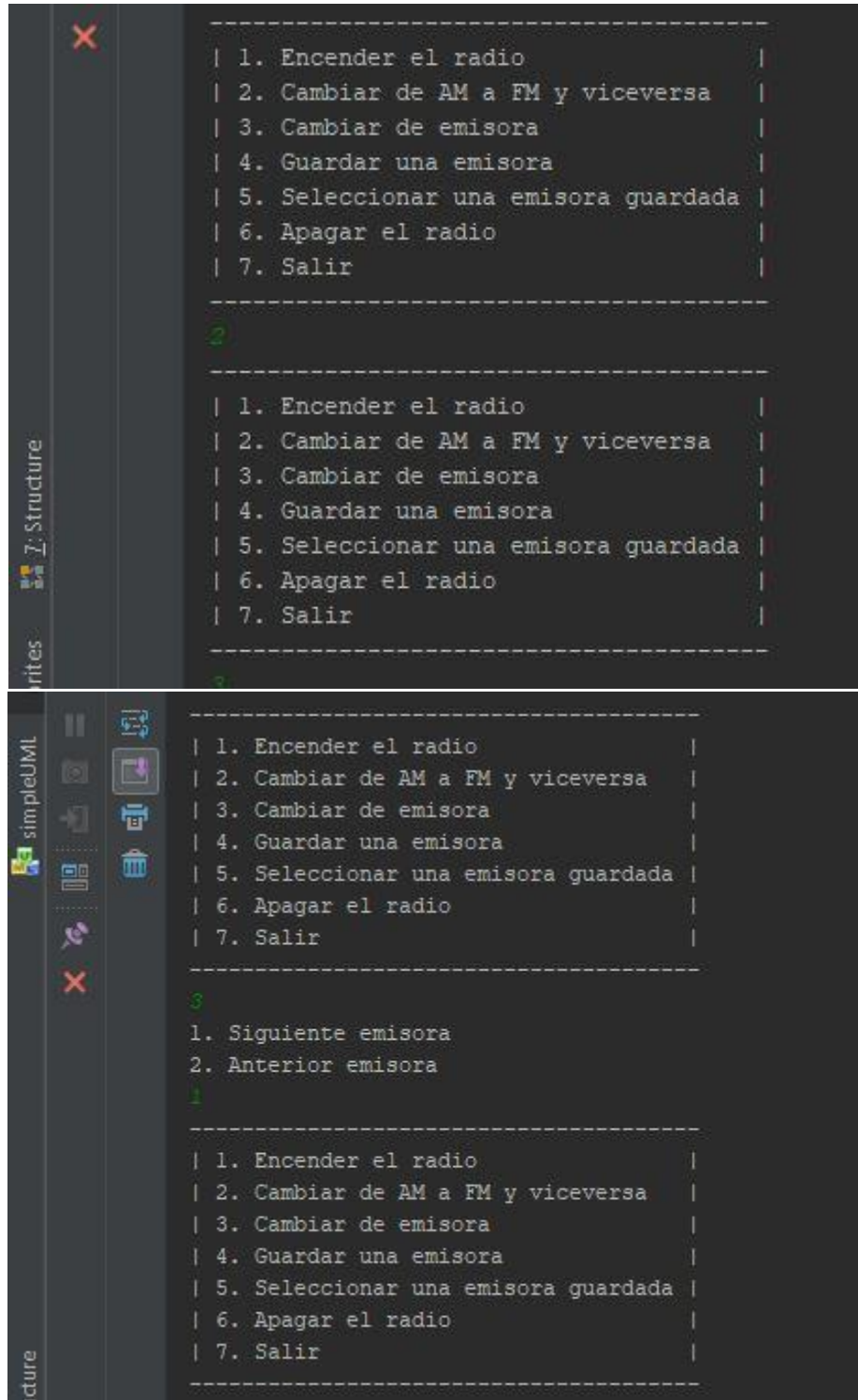
Menu:
  1. Encender Radio

Ingrese su selección: 1

-----
Radio: (ON)
  Frecuencia: FM
  Emisora actual: 530.0

-----

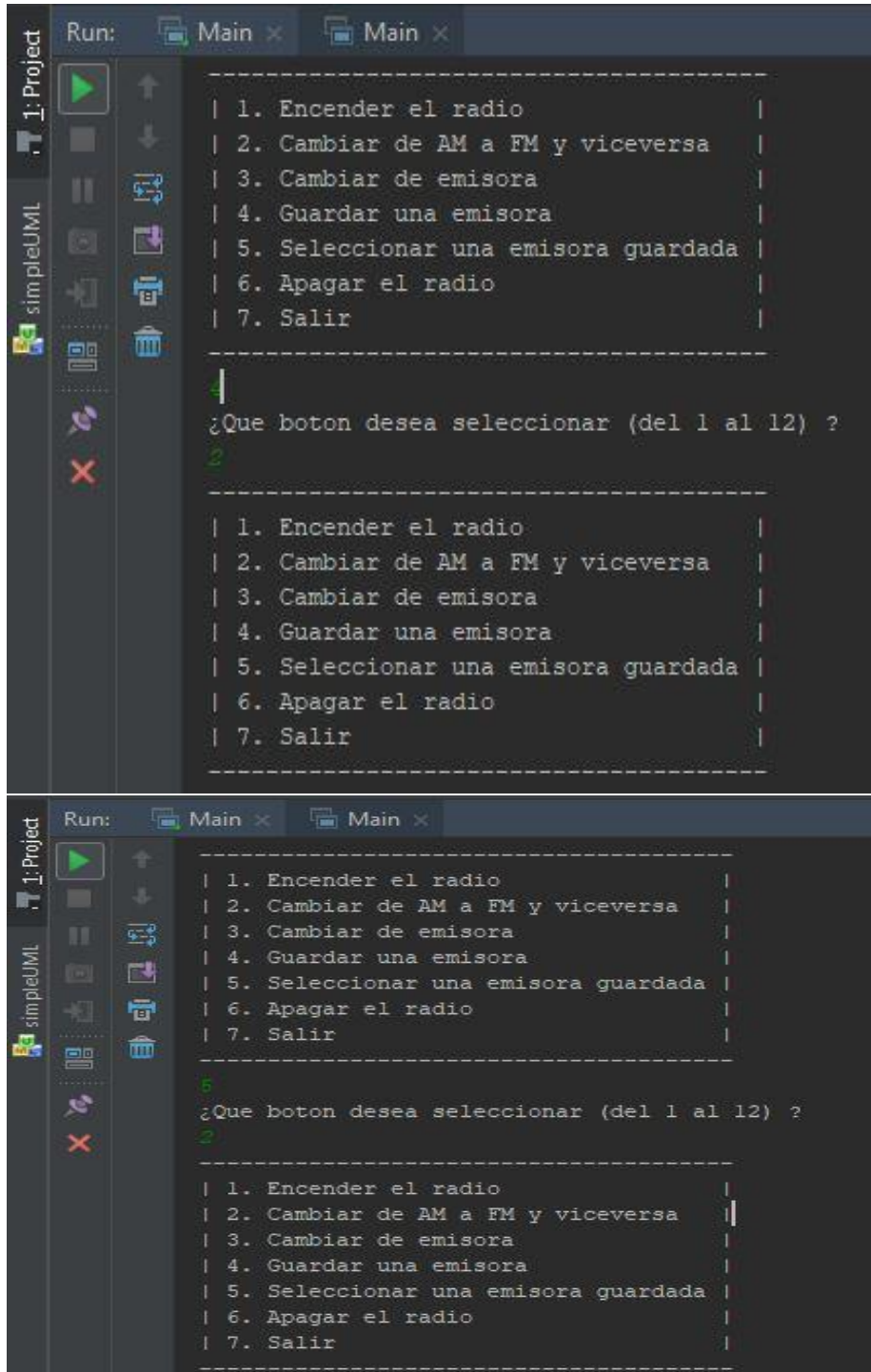
Menu:
  1. Encender Radio
  2. Cambiar Frecuencia (AM/FM).
  3. Cambiar de estación.
  4. Guardar emisora Actual
  5. Sintonizar una emisora de un botón
  6. Apagar el Radio
```



```

-----
| 1. Encender el radio |
| 2. Cambiar de AM a FM y viceversa |
| 3. Cambiar de emisora |
| 4. Guardar una emisora |
| 5. Seleccionar una emisora guardada |
| 6. Apagar el radio |
| 7. Salir |
-----
2
-----
| 1. Encender el radio |
| 2. Cambiar de AM a FM y viceversa |
| 3. Cambiar de emisora |
| 4. Guardar una emisora |
| 5. Seleccionar una emisora guardada |
| 6. Apagar el radio |
| 7. Salir |
-----
3
-----
1. Siguiete emisora
2. Anterior emisora
1
-----
| 1. Encender el radio |
| 2. Cambiar de AM a FM y viceversa |
| 3. Cambiar de emisora |
| 4. Guardar una emisora |
| 5. Seleccionar una emisora guardada |
| 6. Apagar el radio |
| 7. Salir |
-----

```



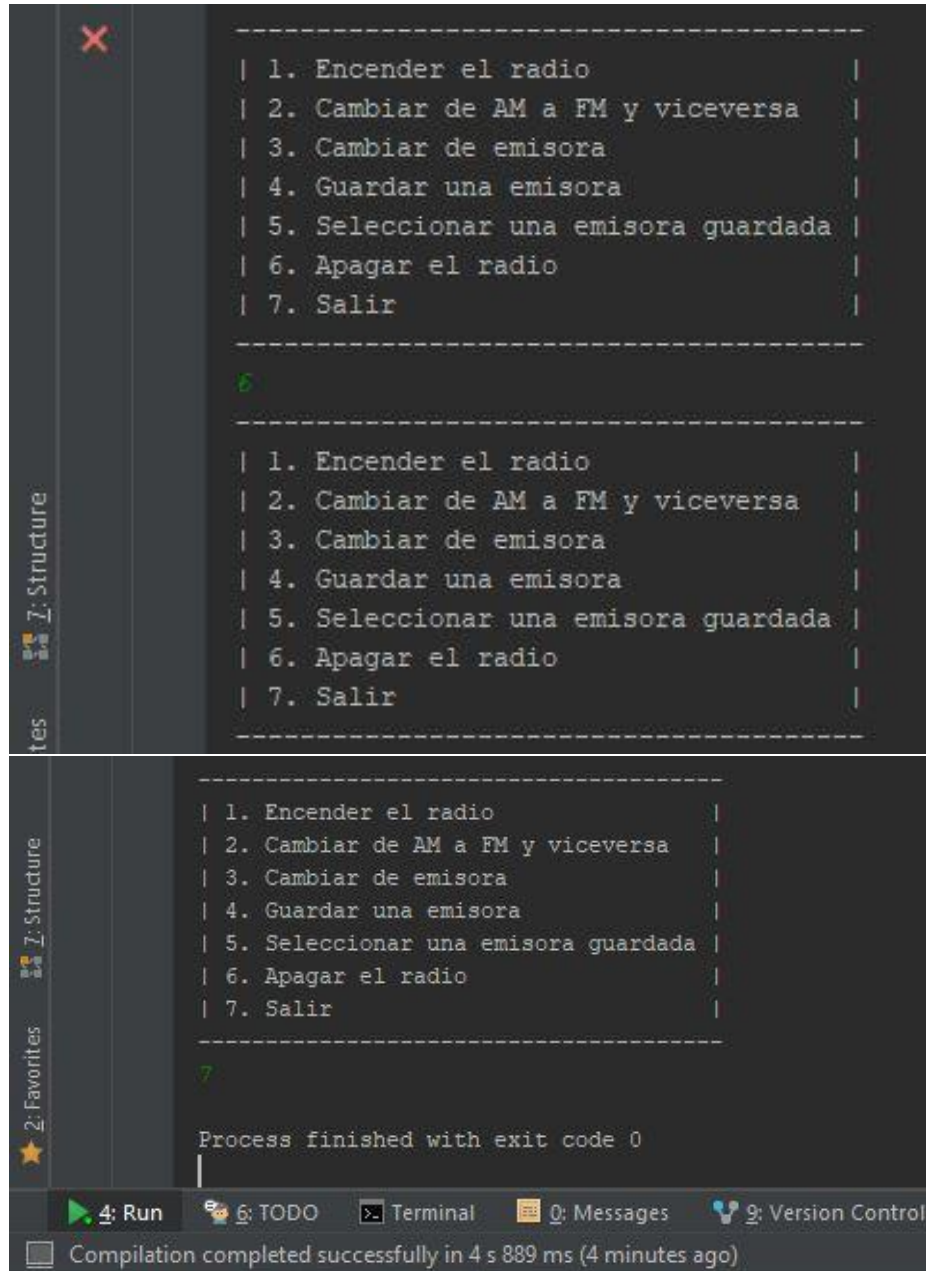
```
Run: Main x Main x
1. Encender el radio
2. Cambiar de AM a FM y viceversa
3. Cambiar de emisora
4. Guardar una emisora
5. Seleccionar una emisora guardada
6. Apagar el radio
7. Salir

4
¿Que boton desea seleccionar (del 1 al 12) ?
2

1. Encender el radio
2. Cambiar de AM a FM y viceversa
3. Cambiar de emisora
4. Guardar una emisora
5. Seleccionar una emisora guardada
6. Apagar el radio
7. Salir

6
¿Que boton desea seleccionar (del 1 al 12) ?
2

1. Encender el radio
2. Cambiar de AM a FM y viceversa
3. Cambiar de emisora
4. Guardar una emisora
5. Seleccionar una emisora guardada
6. Apagar el radio
7. Salir
```

The screenshot shows an IDE with a dark theme. The main editor area displays a menu for a radio application. The menu is enclosed in a box with a red 'X' in the top-left corner. The menu items are:

- 1. Encender el radio
- 2. Cambiar de AM a FM y viceversa
- 3. Cambiar de emisora
- 4. Guardar una emisora
- 5. Seleccionar una emisora guardada
- 6. Apagar el radio
- 7. Salir

The menu is displayed twice, once with a green '6' and once with a green '7'. The IDE interface includes a sidebar with '1: Structure' and '2: Favorites' tabs. The bottom status bar shows '4: Run', '6: TODO', 'Terminal', '0: Messages', and '9: Version Control'. A message at the bottom states: 'Compilation completed successfully in 4 s 889 ms (4 minutes ago)'.

1.2.1 Grupo 6 conformado por Diana de León y Fátima Albeño



```
Run: Main x Main x
Frecuencia: FM
Estacion actual: 87.9
=====
Menu:
1. Apagar
2. Cambiar Frecuencia
3. Cambiar estacion
4. Guardar estacion
5. Elegir estacion guardada
6. Salir
=====
>>> Ingrese el numero de opcion que desea hacer:
3
>>> Ingrese '0' para retroceder o '1' para avanzar de estacion:
1
=====
RADIO
Estado: ON
Frecuencia: FM
Estacion actual: 88.100000000000001
=====

Run: Main x Main x
Menu:
1. Apagar
2. Cambiar Frecuencia
3. Cambiar estacion
4. Guardar estacion
5. Elegir estacion guardada
6. Salir
=====
>>> Ingrese el numero de opcion que desea hacer:
4
>>> Ingrese el numero de boton al cual desea guarda la estacion actual: <1 a 12>
2
=====
RADIO
Estado: ON
Frecuencia: FM
Estacion actual: 88.100000000000001
=====
```




```
Project Files | MyRadio.java | Main.java | Main | main()
Run: Main | Main
simpleUML
Z-Structure

-----
Menu:
1. Apagar
2. Cambiar Frecuencia
3. Cambiar estacion
4. Guardar estacion
5. Elegir estacion guardada
6. Salir
-----

>>> Ingrese el numero de opcion que desea hacer:
6
>>> Ingrese el numero del boton que quiere elegir: <1 a 12>
2

=====
RADIO
Estado: ON
Frecuencia: FM
Estacion actual: 88.10000000000001
=====
-----
```

```
=====
RADIO
Estado: ON
Frecuencia: FM
Estacion actual: 88.10000000000001
=====

-----
Menu:
1. Apagar
2. Cambiar Frecuencia
3. Cambiar estacion
4. Guardar estacion
5. Elegir estacion guardada
6. Salir
-----

>>> Ingrese el numero de opcion que desea hacer:
6
>>> Saliendo...

Process finished with exit code 0
```



```
=====
RADIO
Estado: OFF
Frecuencia: --
Estacion actual: --
=====

-----
Menu:
1. Prender
6. Salir
-----

>>> Ingrese el numero de opcion que desea hacer:
1

=====
RADIO
Estado: ON
Frecuencia: AM
Estacion actual: 530.0
=====

-----
Menu:
1. Apagar
2. Cambiar Frecuencia
3. Cambiar estacion
4. Guardar estacion
5. Elegir estacion guardada
6. Salir
-----

>>> Ingrese el numero de opcion que desea hacer:
2

=====
RADIO
Estado: ON
Frecuencia: FM
Estacion actual: 87.9
=====
```