

```

package practica4junit;

import static org.junit.jupiter.api.Assertions.*;

import org.junit.jupiter.api.Test;

class VueloTest {

    @Test
    void VueloTest1() {
        Vuelo prueba = new Vuelo(956, "Ryanair", "Tenerife", "Maldivas",
200, false);
        int nuevovalortotal = 1000;
        prueba.ampliarcapacidad(nuevovalortotal);
        assertEquals(nuevovalortotal, prueba.getasientostotales());
    }
    /*Como la capacidad no puede ser mayor a 800, la prueba resulta
fallida*/

    @Test
    void VueloTest2() {
        Vuelo prueba = new Vuelo(956, "Ryanair", "Tenerife", "Maldivas",
800, false);
        int nuevovalortotal = 700;
        prueba.ampliarcapacidad(nuevovalortotal);
        assertEquals(nuevovalortotal, prueba.getasientostotales());
    }
    /*Como nuevovalortotal es menor que asientostotales , la prueba
resulta fallida*/

    @Test
    void VueloTest3() {
        Vuelo prueba = new Vuelo(956, "Ryanair", "Tenerife", "Maldivas",
600, false);
        int nuevovalortotal = 700;
        prueba.ampliarcapacidad(nuevovalortotal);
        assertEquals(nuevovalortotal, prueba.getasientostotales());
    }
    /*Como nuevovalortotal es mayor que asientostotales , la prueba
resulta correcta*/

    //-----
    -----

    @Test

```

```

void VueloTest4() {
    Vuelo prueba = new Vuelo(956, "Ryanair", "Tenerife", "Maldivas",
600, false);
    boolean nescala = false;
    prueba.modificarescala(nescala);
    assertEquals(nescala, prueba.getescala());
}
/*No pasar nada, ya que nescala y escala tienen el mismo valor.*/

@Test
void VueloTest5() {
    Vuelo prueba = new Vuelo(956, "Ryanair", "Tenerife", "Maldivas",
600, true);
    boolean nescala = false;
    prueba.modificarescala(nescala);
    assertEquals(nescala, prueba.getescala());
}
/*Como nescala y escala tienen valores distintos, nescala
sobreescribir el valor de escala*/

//-----
-----

@Test
void VueloTest6() {
    Vuelo prueba = new Vuelo(956, "Ryanair", "Tenerife", "Maldivas",
600, true);
    int asientosReservados = 3;
    prueba.reservarasiento(asientosReservados);
    prueba.setasientosdisponibles(425);
    int asientosEsperados = prueba.getasientosdisponibles() -
asientosReservados;
    assertEquals(asientosEsperados, prueba.getasientosdisponibles());
}
/*Como el número de asientos reservados es 0, nos saltar un mensaje
para que introduzcamos
un valor válido*/

@Test
void VueloTest7() {
    Vuelo prueba = new Vuelo(956, "Ryanair", "Tenerife", "Maldivas",
600, true);
    int asientosReservados = 500;
    prueba.reservarasiento(asientosReservados);

```

```

        prueba.setasientosdisponibles(425);
        int asientosEsperados = prueba.getasientosdisponibles() -
asientosReservados;
        assertEquals(asientosEsperados, prueba.getasientosdisponibles());
    }

    /*Como el número de asientos reservados es mayor que el número de
asientos disponibles,
    * el programa nos avisar de que no hay asientos suficientes.*/

    @Test
    void VueloTest8() {
        Vuelo prueba = new Vuelo(956, "Ryanair", "Tenerife", "Maldivas",
600, true);
        int asientosReservados = 3;
        prueba.reservarasiento(asientosReservados);
        prueba.setasientosdisponibles(425);
        int asientosEsperados = prueba.getasientosdisponibles() -
asientosReservados;
        assertEquals(asientosEsperados, prueba.getasientosdisponibles());
    }

    /*Como el número de asientos reservados es menor al número de
asientos totales,
    * la petición se ejecutar correctamente.*/

    //-----
    -----

    @Test
    void VueloTest9() {
        Vuelo prueba = new Vuelo(956, "Ryanair", "Tenerife", "Maldivas",
600, true);
        int cantidad = 0;
        prueba.devolverreserva(cantidad);
        prueba.setasientosdisponibles(425);
        int asientosEsperados = prueba.getasientosdisponibles() +
cantidad;
        assertEquals(asientosEsperados, prueba.getasientosdisponibles());
    }

    /*Como cantidad es igual a 0, el programa nos avisar que debemos
    * devolver al menos un asiento.*/

    @Test
    void VueloTest10() {

```

```

        Vuelo prueba = new Vuelo(956, "Ryanair", "Tenerife", "Maldivas",
600, true);
        int cantidad = 300;
        prueba.devolverreserva(cantidad);
        prueba.setasientosdisponibles(425);
        int asientosEsperados = prueba.getasientosdisponibles() +
cantidad;
        assertEquals(asientosEsperados, prueba.getasientosdisponibles());
    }
    /*Como asientosEsperados es mayor a los asientos totales, el programa
nos avisar💎
    * que la cantidad supera el n💎mero de asientos totales.*/

    @Test
    void VueloTest24() {
        Vuelo prueba = new Vuelo(956, "Ryanair", "Tenerife", "Maldivas",
600, true);
        int cantidad = 10;
        prueba.devolverreserva(cantidad);
        prueba.setasientosdisponibles(425);
        int asientosEsperados = prueba.getasientosdisponibles() +
cantidad;
        assertEquals(asientosEsperados, prueba.getasientosdisponibles());
    }
    /*Como asientosEsperados es menor a los asientos totales, el programa
sobreescribir💎 el valor de asientosdisponibles*/

    //-----
    -----

    @Test
    void VueloTest11() {
        int nidvuelo = 967;
        Vuelo prueba = new Vuelo(nidvuelo, "Ryanair", "Tenerife",
"Maldivas", 600, true);
        assertEquals(nidvuelo, prueba.getId());
    }

    @Test
    void VueloTest12() {
        String naerolinea = "Ryanair";
        Vuelo prueba = new Vuelo(967, naerolinea, "Tenerife", "Maldivas",
600, true);
        assertEquals(naerolinea, prueba.getaerolinea());
    }

```

```

    }

    @Test
    void VueloTest13() {
        String norigen = "Tenerife";
        Vuelo prueba = new Vuelo(967, "Ryanair", norigen, "Maldivas",
600, true);
        assertEquals(norigen, prueba.getorigen());
    }

    @Test
    void VueloTest14() {
        String ndestino = "Maldivas";
        Vuelo prueba = new Vuelo(967, "Ryanair", "Tenerife", ndestino,
600, true);
        assertEquals(ndestino, prueba.getdestino());
    }

    @Test
    void VueloTest15() {
        int nasientostotales = 600;
        Vuelo prueba = new Vuelo(967, "Ryanair", "Tenerife", "Maldivas",
nasientostotales, true);
        assertEquals(nasientostotales, prueba.getasientostotales());
    }

    @Test
    void VueloTest16() {
        Vuelo prueba = new Vuelo(967, "Ryanair", "Tenerife", "Maldivas",
600, true);
        int nasientosdisponibles = 200;
        prueba.setasientosdisponibles(nasientosdisponibles);
        assertEquals(nasientosdisponibles,
prueba.getasientosdisponibles());
    }

    @Test
    void VueloTest17() {
        boolean nescala = false;
        Vuelo prueba = new Vuelo(967, "Ryanair", "Tenerife", "Maldivas",
600, nescala);
        assertEquals(nescala, prueba.getescala());
    }

    @Test
    void VueloTest18() {

```

```

        Vuelo prueba = new Vuelo(967, "Ryanair", "Tenerife", "Maldivas",
600, true);
        int nidvuelo = 541;
        prueba.settid(nidvuelo);
        assertEquals(nidvuelo, prueba.gettid());
    }

    @Test
    void VueloTest19() {
        Vuelo prueba = new Vuelo(967, "Ryanair", "Tenerife", "Maldivas",
600, true);
        String naerolinea = "Iberia";
        prueba.setaerolinea(naerolinea);
        assertEquals(naerolinea, prueba.getaerolinea());
    }
    //ATENCIÓN, NO ME PATROCINA NINGUNA AEROLÍNEA

    @Test
    void VueloTest20() {
        Vuelo prueba = new Vuelo(967, "Ryanair", "Tenerife", "Maldivas",
600, true);
        String norigen = "Lisboa";
        prueba.setorigen(norigen);
        assertEquals(norigen, prueba.getorigen());
    }

    @Test
    void VueloTest21() {
        Vuelo prueba = new Vuelo(967, "Ryanair", "Tenerife", "Maldivas",
600, true);
        String ndestino = "Múnich";
        prueba.setdestino(ndestino);
        assertEquals(ndestino, prueba.getdestino());
    }

    @Test
    void VueloTest22() {
        Vuelo prueba = new Vuelo(967, "Ryanair", "Tenerife", "Maldivas",
600, true);
        int nasientostotales = 800;
        prueba.setasientostotales(nasientostotales);
        assertEquals(nasientostotales, prueba.getasientostotales());
    }

    @Test

```

```
void VueloTest23() {  
    Vuelo prueba = new Vuelo(967, "Ryanair", "Tenerife", "Maldivas",  
600, true);  
    int nasientosdisponibles = 400;  
    prueba.setasientosdisponibles(nasientosdisponibles);  
    assertEquals(nasientosdisponibles,  
prueba.getasientosdisponibles());  
}  
}
```