

PRACTICA 1

Link de github: https://github.com/MaelGM/Pruebas_JUnit.git

```
package es.iesmz;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.assertEquals;

public class EmpleadoBRTest {

    @Test
    void testCalculaSalarioNeto1() { assertEquals( expected: 1640, EmpleadoBR.calculaSalarioNeto( salarioBruto: 2000)); }
    @Test
    void testCalculaSalarioNeto2() { assertEquals( expected: 1230, EmpleadoBR.calculaSalarioNeto( salarioBruto: 1500)); }
    @Test
    void testCalculaSalarioNeto3() { assertEquals( expected: 1259.9916f, EmpleadoBR.calculaSalarioNeto( salarioBruto: 1499.99f)); }
    @Test
    void testCalculaSalarioNeto4() { assertEquals( expected: 1050, EmpleadoBR.calculaSalarioNeto( salarioBruto: 1250)); }
    @Test
    void testCalculaSalarioNeto5() { assertEquals( expected: 840, EmpleadoBR.calculaSalarioNeto( salarioBruto: 1000)); }
    @Test
    void testCalculaSalarioNeto6() { assertEquals( expected: 999.99f, EmpleadoBR.calculaSalarioNeto( salarioBruto: 999.99f)); }
    @Test
    void testCalculaSalarioNeto7() { assertEquals( expected: 500, EmpleadoBR.calculaSalarioNeto( salarioBruto: 500)); }
    @Test
    void testCalculaSalarioNeto8() { assertEquals( expected: 0, EmpleadoBR.calculaSalarioNeto( salarioBruto: 0)); }
    @Test
    void testCalculaSalarioNeto9() { assertEquals( expected: -1, EmpleadoBR.calculaSalarioNeto( salarioBruto: -1)); }

    @Test
    void testCalculaSalarioBruto1(){
        assertEquals( expected: 1360, EmpleadoBR.calculaSalarioBruto(TipoEmpleado.venedor, ventasMes: 2000, horasExtra: 8));
    }
    @Test
    void testCalculaSalarioBruto2(){
        assertEquals( expected: 1260, EmpleadoBR.calculaSalarioBruto(TipoEmpleado.venedor, ventasMes: 1500, horasExtra: 3));
    }
    @Test
    void testCalculaSalarioBruto3(){
        assertEquals( expected: 1100, EmpleadoBR.calculaSalarioBruto(TipoEmpleado.venedor, ventasMes: 1499.99f, horasExtra: 0));
    }
    @Test
    void testCalculaSalarioBruto4(){
        assertEquals( expected: 1760, EmpleadoBR.calculaSalarioBruto(TipoEmpleado.encarregat, ventasMes: 1250, horasExtra: 8));
    }
}
```

```
@Test
void testCalculaSalarioBruto5(){
    assertEquals( expected: 1600, EmpleadoBR.calculaSalarioBruto(TipoEmpleado.encarregat, ventasMes: 1000, horasExtra: 0));
}
@Test
void testCalculaSalarioBruto6(){
    assertEquals( expected: 1560, EmpleadoBR.calculaSalarioBruto(TipoEmpleado.encarregat, ventasMes: 999.99f, horasExtra: 3));
}
@Test
void testCalculaSalarioBruto7(){
    assertEquals( expected: 1500, EmpleadoBR.calculaSalarioBruto(TipoEmpleado.encarregat, ventasMes: 500, horasExtra: 0));
}
@Test
void testCalculaSalarioBruto8(){
    assertEquals( expected: 1660, EmpleadoBR.calculaSalarioBruto(TipoEmpleado.encarregat, ventasMes: 0, horasExtra: 8));
}
@Test
void testCalculaSalarioBruto9() { assertEquals( expected: -1, EmpleadoBR.calculaSalarioBruto(TipoEmpleado.venedor, ventasMes: -1, horasExtra: 8)); }
@Test
void testCalculaSalarioBruto10(){
    assertEquals( expected: -1, EmpleadoBR.calculaSalarioBruto(TipoEmpleado.venedor, ventasMes: 1500, horasExtra: -1));
}
@Test
void testCalculaSalarioBruto11() { assertEquals( expected: -1, EmpleadoBR.calculaSalarioBruto( tipus: null, ventasMes: 1500, horasExtra: 8)); }
}
```