

Empleado

```
public class EmpleadoBR { 2 usages

    static float calculaSalarioBruto(TipoEmpleado tipus, float ventasMes, float horasExtra) { 1 usage
        float base = 0;
        float extra;
        if (tipus == TipoEmpleado.venedor){
            base=1000;
        } else if (tipus == TipoEmpleado.encarregat){
            base=1500;
        }

        if (ventasMes>=1500){
            base=base+200;
        } else if (ventasMes>=1000) {
            base=base+100;
        }
        extra=horasExtra*20;

        if (tipus == null || ventasMes<0 || horasExtra<0){
            return -1;
        } else {
            return base+extra;
        }
    }
}
```

```
static float calculaSalarioNeto(float salarioBruto){ 1 usage
    float retencion=0;
    if (salarioBruto>1500){
        retencion=18;
    } else if (salarioBruto>1000) {
        retencion=16;
    }

    if (salarioBruto<0){
        return -1;
    } else {
        return salarioBruto*(1-(retencion/100));
    }
}
```

Test

```
public class EmpleadoBRTTest {

    @BeforeEach
    public void BE() {
        System.out.println("EXECUTE PROVA");
    }

    @Test
    public void testCalculaSalarioBruto1() {
        float resultadoReal = EmpleadoBR.calculaSalarioBruto(
            TipoEmpleado.venedor, ventasMes: 2000.0f, horasExtra: 8.0f);
        float resultadoEsperado = 1360.0f;
        assertEquals(resultadoEsperado, resultadoReal, delta: 0.01);
    }

    @Test
    public void testCalculaSalarioBruto2() {
        float resultadoReal = EmpleadoBR.calculaSalarioBruto(
            TipoEmpleado.venedor, ventasMes: 1500.0f, horasExtra: 3.0f);
        float resultadoEsperado = 1260.0f;
        assertEquals(resultadoEsperado, resultadoReal, delta: 0.01);
    }

    @Test
    public void testCalculaSalarioBruto3() {
        float resultadoReal = EmpleadoBR.calculaSalarioBruto(
            TipoEmpleado.venedor, ventasMes: 1499.99f, horasExtra: 0.0f);
        float resultadoEsperado = 1100.0f;
        assertEquals(resultadoEsperado, resultadoReal, delta: 0.01);
    }
}
```

```
@Test
public void testCalculaSalarioBruto4() {
    float resultadoReal = EmpleadoBR.calculaSalarioBruto(
        TipoEmpleado.encarregat, ventasMes: 1250.0f, horasExtra: 8.0f);
    float resultadoEsperado = 1760.0f;
    assertEquals(resultadoEsperado, resultadoReal, delta: 0.01);
}

@Test
public void testCalculaSalarioBruto5() {
    float resultadoReal = EmpleadoBR.calculaSalarioBruto(
        TipoEmpleado.encarregat, ventasMes: 1000.0f, horasExtra: 0.0f);
    float resultadoEsperado = 1600.0f;
    assertEquals(resultadoEsperado, resultadoReal, delta: 0.01);
}

@Test
public void testCalculaSalarioBruto6() {
    float resultadoReal = EmpleadoBR.calculaSalarioBruto(
        TipoEmpleado.encarregat, ventasMes: 999.99f, horasExtra: 3.0f);
    float resultadoEsperado = 1560.0f;
    assertEquals(resultadoEsperado, resultadoReal, delta: 0.01);
}

@Test
public void testCalculaSalarioBruto7() {
    float resultadoReal = EmpleadoBR.calculaSalarioBruto(
        TipoEmpleado.encarregat, ventasMes: 500.0f, horasExtra: 0.0f);
    float resultadoEsperado = 1500.0f;
    assertEquals(resultadoEsperado, resultadoReal, delta: 0.01);
}
```

```

@Test
public void testCalculaSalarioBruto8() {
    float resultadoReal = EmpleadoBR.calculaSalarioBruto(
        TipoEmpleado.encarregat, ventasMes: 0.0f, horasExtra: 8.0f);
    float resultadoEsperado = 1660.0f;
    assertEquals(resultadoEsperado, resultadoReal, delta: 0.01);
}

@Test
public void testCalculaSalarioBruto9() {
    float resultadoReal = EmpleadoBR.calculaSalarioBruto(
        TipoEmpleado.venedor, ventasMes: -1.0f, horasExtra: 8.0f);
    float resultadoEsperado = -1.0f;
    assertEquals(resultadoEsperado, resultadoReal, delta: 0.01);
}

@Test
public void testCalculaSalarioBruto10() {
    float resultadoReal = EmpleadoBR.calculaSalarioBruto(
        TipoEmpleado.venedor, ventasMes: 1500.0f, horasExtra: -1.0f);
    float resultadoEsperado = -1.0f;
    assertEquals(resultadoEsperado, resultadoReal, delta: 0.01);
}

@Test
public void testCalculaSalarioBruto11() {
    float resultadoReal = EmpleadoBR.calculaSalarioBruto(
        tipus: null, ventasMes: 1500.0f, horasExtra: 8.0f);
    float resultadoEsperado = -1.0f;
    assertEquals(resultadoEsperado, resultadoReal, delta: 0.01);
}

```

```

@Test
public void testCalculaSalarioNeto1() {
    float resultadoReal = EmpleadoBR.calculaSalarioNeto( salarioBruto: 2000.0f);
    float resultadoEsperado = 1640.0f;
    assertEquals(resultadoEsperado, resultadoReal, delta: 0.01);
}

@Test
public void testCalculaSalarioNeto2() {
    float resultadoReal = EmpleadoBR.calculaSalarioNeto( salarioBruto: 1500.0f);
    float resultadoEsperado = 1230.0f;
    assertEquals(resultadoEsperado, resultadoReal, delta: 0.01);
}

@Test
public void testCalculaSalarioNeto3() {
    float resultadoReal = EmpleadoBR.calculaSalarioNeto( salarioBruto: 1499.99f);
    float resultadoEsperado = 1259.9916f;
    assertEquals(resultadoEsperado, resultadoReal, delta: 0.01);
}

@Test
public void testCalculaSalarioNeto4() {
    float resultadoReal = EmpleadoBR.calculaSalarioNeto( salarioBruto: 1250.0f);
    float resultadoEsperado = 1050.0f;
    assertEquals(resultadoEsperado, resultadoReal, delta: 0.01);
}

@Test
public void testCalculaSalarioNeto5() {
    float resultadoReal = EmpleadoBR.calculaSalarioNeto( salarioBruto: 1000.0f);
    float resultadoEsperado = 840.0f;
    assertEquals(resultadoEsperado, resultadoReal, delta: 0.01);
}

```

```
@Test
public void testCalculaSalarioNeto6() {
    float resultadoReal = EmpleadoBR.calculaSalarioNeto( salarioBruto: 999.99f);
    float resultadoEsperado = 999.99f;
    assertEquals(resultadoEsperado, resultadoReal, delta: 0.01);
}

@Test
public void testCalculaSalarioNeto7() {
    float resultadoReal = EmpleadoBR.calculaSalarioNeto( salarioBruto: 500.0f);
    float resultadoEsperado = 500.0f;
    assertEquals(resultadoEsperado, resultadoReal, delta: 0.01);
}

@Test
public void testCalculaSalarioNeto8() {
    float resultadoReal = EmpleadoBR.calculaSalarioNeto( salarioBruto: 0.0f);
    float resultadoEsperado = 0.0f;
    assertEquals(resultadoEsperado, resultadoReal, delta: 0.01);
}

@Test
public void testCalculaSalarioNeto9() {
    float resultadoReal = EmpleadoBR.calculaSalarioNeto( salarioBruto: -1.0f);
    float resultadoEsperado = -1.0f;
    assertEquals(resultadoEsperado, resultadoReal, delta: 0.01);
}
```