

Entrega PRO 3 – UD5 – Entornos de Desarrollo

Ejercicio 1:

Código de EmpleadoBRBrutoTest.java :

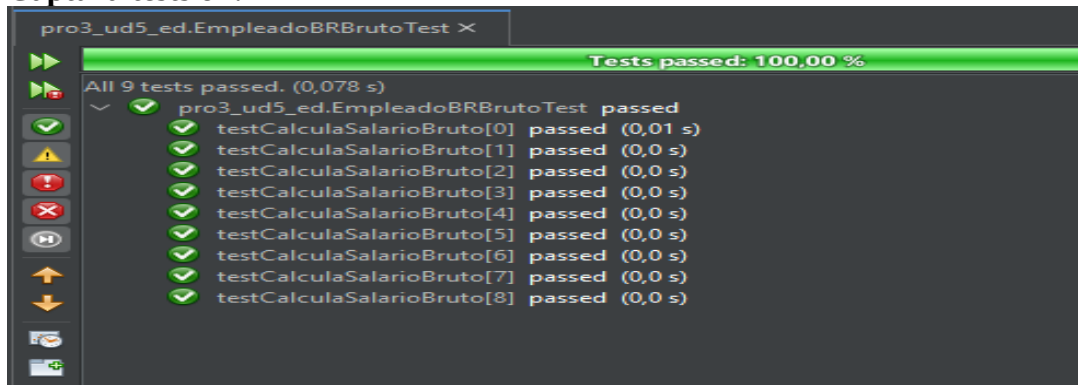
```
//package, imports...
@RunWith(Parameterized.class)
public class EmpleadoBRBrutoTest {
    private TipoEmpleado tipo;
    private float ventasMes;
    private float horasExtra;
    private Object resultadoEsperado; //Es de tipo Object porque puede ser un float o una Exception

    public EmpleadoBRBrutoTest(TipoEmpleado tipo, float ventasMes, float horasExtra, Object resultadoEsperado) {
        this.tipo = tipo;
        this.ventasMes = ventasMes;
        this.horasExtra = horasExtra;
        this.resultadoEsperado = resultadoEsperado;
    }

    @Parameters
    public static Collection<Object[]> parametrosBruto() {
        return Arrays.asList(new Object[][]{
            {TipoEmpleado.VENDEDOR, 2000, 8, 1360}, {TipoEmpleado.VENDEDOR, 1500, 3, 1260},
            {TipoEmpleado.VENDEDOR, 1499.99f, 0, 1100}, {TipoEmpleado.ENCARGADO, 1250, 8, 1760},
            {TipoEmpleado.ENCARGADO, 1000, 0, 1600}, {TipoEmpleado.ENCARGADO, 999.99f, 3, 1560},
            {TipoEmpleado.ENCARGADO, 500, 0, 1500}, {TipoEmpleado.ENCARGADO, 0, 8, 1660},{null, 1500, 8,
            BRException.class}});
    }

    @Test
    public void testCalculaSalarioBruto() throws Exception {
        EmpleadoBR instance = new EmpleadoBR();
        Object expResult = this.resultadoEsperado;
        if (expResult.equals(Float.class)) {
            float result = instance.calculaSalarioBruto(tipo, ventasMes, horasExtra);
            assertEquals((float) expResult, result, 0);
        } else {
            try {
                float result = instance.calculaSalarioBruto(tipo, ventasMes, horasExtra);
            } catch (Exception e) {
                //Comprobamos que realmente devuelve el tipo de excepción esperada
                assertEquals(expResult, e.getClass());
            }
        }
    }
}
```

Captura tests ok:



Ejercicio 2:

Código EmpleadoBRNetoTest.java :

```
//package, imports...
```

```
@RunWith(Parameterized.class)
public class EmpleadoBRNetoTest {
    private float salarioBruto;
    private float netoEsperado;

    @Parameters
    public static Collection<Object[]> parametrosNeto() {
        return Arrays.asList(new Object[][]{
            {2000, 1640}, {1500, 1230}, {1499.99f, 1259.9916f}, {1250, 1050},
            {1000, 840}, {999.99f, 999.99f}, {500, 500},{0, 0}
        });
    }

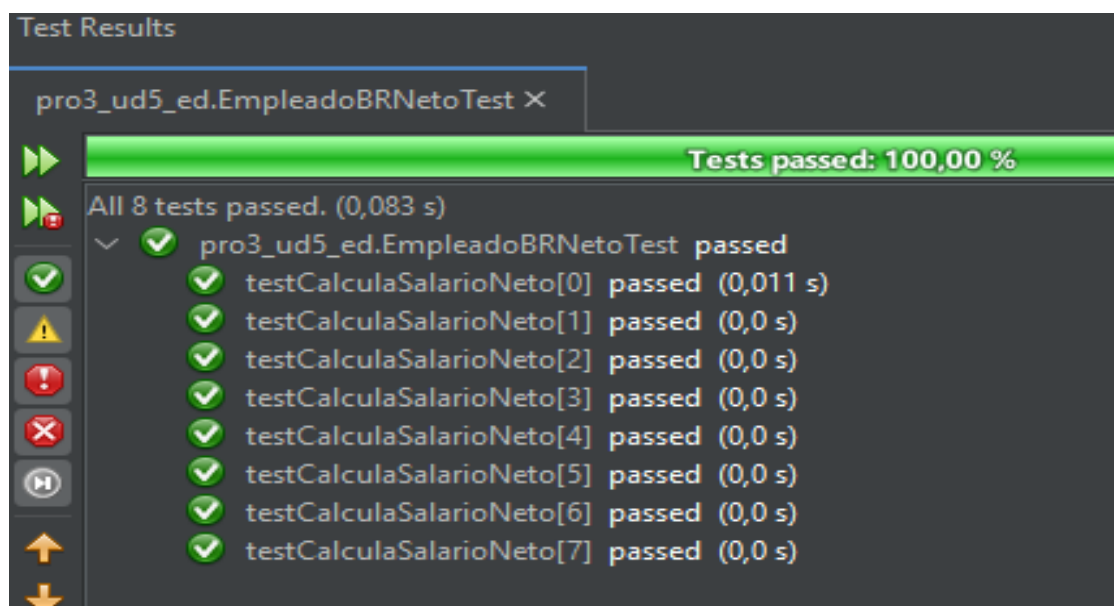
    public EmpleadoBRNetoTest(float salarioBruto, float netoEsperado) {
        this.salarioBruto = salarioBruto;
        this.netoEsperado = netoEsperado;
    }

    @Test
    public void testCalculaSalarioNeto() throws Exception {

        EmpleadoBR instance = new EmpleadoBR();
        float expResult = this.netoEsperado;
        float result = instance.calculaSalarioNeto(salarioBruto);

        assertEquals(expResult, result, 0.001f);
    }
}
```

Captura tests ok:



Ejercicio 3:

Código EmpleadoBRTestSuite.java :

```
//package, imports...
```

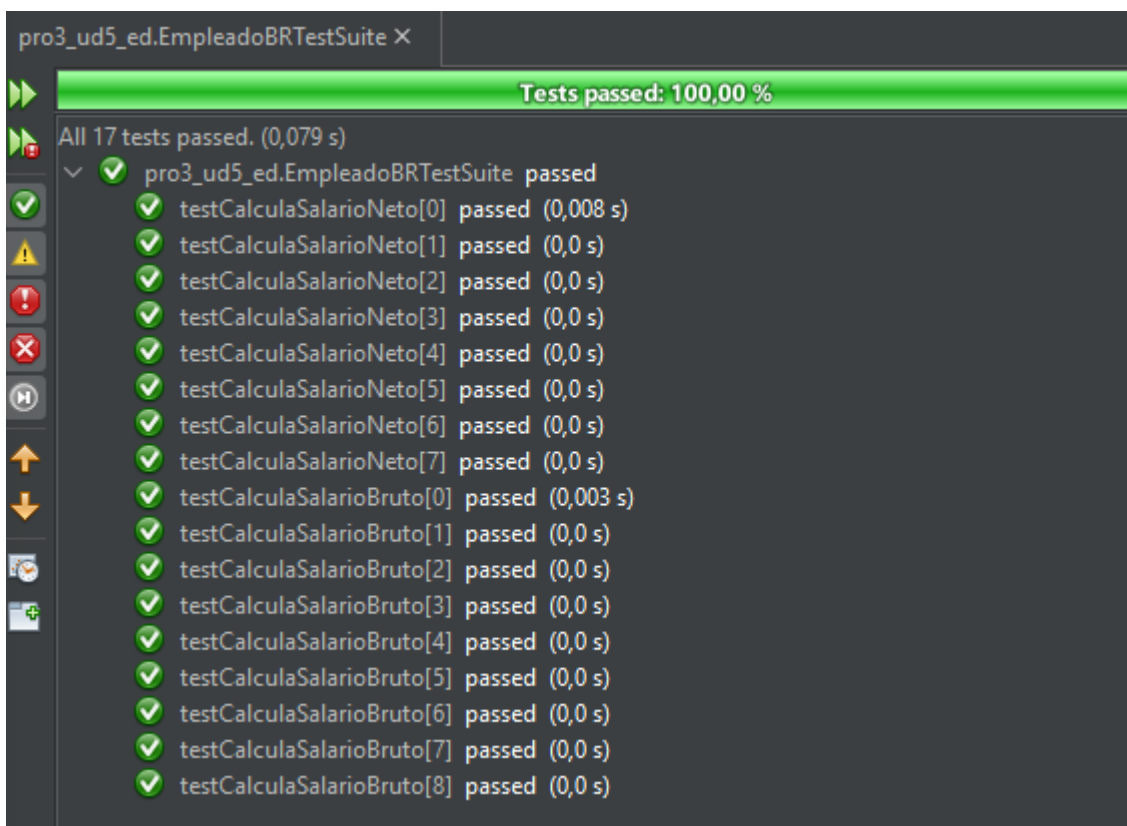
```
@RunWith(Suite.class)
```

```
@Suite.SuiteClasses({pro3_ud5_ed.EmpleadoBRNetoTest.class,  
pro3_ud5_ed.EmpleadoBRBrutoTest.class})
```

```
public class EmpleadoBRTestSuite {
```

```
}
```

Captura tests ok:



Link tarea en GitHub: https://github.com/Kadeathdeluz/PRO3_UD5_ED