

Este es l'aspecte final que ens quedara al nostre projecte un cop creades les classes i paquets

```
package fachada;
public class Suma {
    private int a;
    private int b;

    public Suma(int a, int b) {
        this.a = a;
        this.b = b;
    }

    /**
     * Suma a + b
     * @return a+b
     */
    public int suma() {
        return a + b;
    }

    /**
     * Resta a - b
     * @return a-b
     */
    public int resta() {
        return a - b;
    }
}
```

Aquesta d'aquí es la 1era classe suma

```
package fachada;

public class Multiplicacion {
    private int a;
    private int b;

    public Multiplicacion(int a, int b) {
        this.a = a;
        this.b = b;
    }

    public int multiplicar() {
        return a*b;
    }
}
```

```
package fachada2;

public class Division {
    private int a;
    private int b;

    public Division(int a, int b) {
        this.a = a;
        this.b = b;
    }

    public int divide() {
        return a/b;
    }
}
```

Fem el mateix per a la classe Multiplicacio i divisio sempre en l'ordre que ja tenim al principi

```

package fachadatest;

import static org.junit.Assert.*;
import fachada.*;
import org.junit.After;
import org.junit.Before;
import org.junit.Test;

public class SumaTest {
    public Suma suma;

    @Before
    public void antesDelTest() {
        /**
         * El metodo precedido por la etiqueta @Before
         * es para indicar a JUnit que debe ejecutarlo
         * antes de ejecutar los Tests que figuran en
         * esta clase.
         */
        this.suma = new Suma(3, 5);
    }

    @After
    public void despuesDelTest() {
        /**
         * La etiqueta @After es la antitesis de @Before.
         * Simplemente este metodo se ejecutara despues de
         * ejecutar todos los tests de esta clase.
         */
        // en este caso no hago nada, solo esta de ejemplo
    }

    @Test
    public void testSuma() {
        /**
         * Marcais el metodo con la etiqueta @Test y es
         * importante que el nombre del metodo comience
         * siempre por test.
         */
        int resultado = this.suma.suma();
        // con esto verificamos que el resultado es el esperado
        assertTrue(resultado == 8);
    }

    @Test
    public void testResta() {
        int resultado = this.suma.resta();
        assertTrue(resultado == -2);
    }
}

```

Aquesta es la JUnit Test de Suma

```

package fachadatest;

import static org.junit.Assert.*;

public class MultiplicacionTest {
    /**
     * Lo mismo de antes pero esta vez sin
     * before y after.
     */

    @Test
    public void testMultiplicar() {
        Multiplicacion multiplicacion = new Multiplicacion(3, 5);
        int resultado = multiplicacion.multiplicar();
        assertTrue(resultado == 15);
    }
}

```

```

package fachadatest2;

import static org.junit.Assert.*;

public class DivisionTest {
    @Test
    public void testDivide() {
        Division division = new Division(4, 2);
        int resultado = division.divide();
        assertTrue(resultado == 2);
    }
}

```

Fem el mateix JUnit test per a Multiplicacion i Division

```

package fachadatest;

import org.junit.runner.RunWith;
import org.junit.runners.Suite;
import org.junit.runners.Suite.SuiteClasses;

@RunWith(Suite.class)
@SuiteClasses({SumaTest.class, MultiplicacionTest.class})
public class FachadaTest {

}

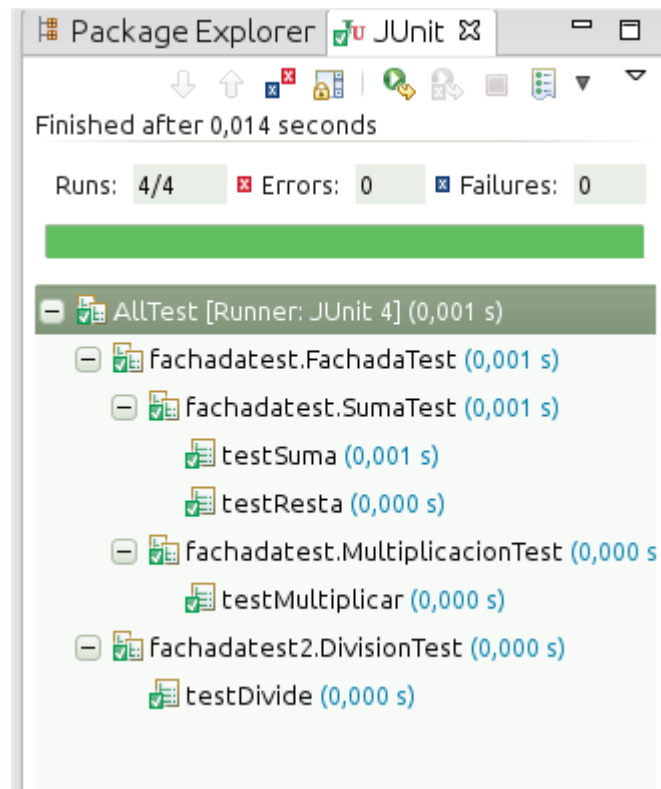
```

Aquesta es una classe que selecciona els metodes que voldrem executar de dins del paquet on son aquestes 2 ja que division esta a un altre paquet

```
import org.junit.runner.RunWith;

@RunWith(Suite.class)
@SuiteClasses({FachadaTest.class, DivisionTest.class})
public class AllTest {}
```

Per fer-los anar tots el que fem es crear un classe fora del paquet i ens crea un default package i dins posem el mateix ficat la classe fachadatest que es la que hem fet anteriorment i la DivisionTest que es la que ens fa falta



Hi aquest es el resultat de fer els tests.