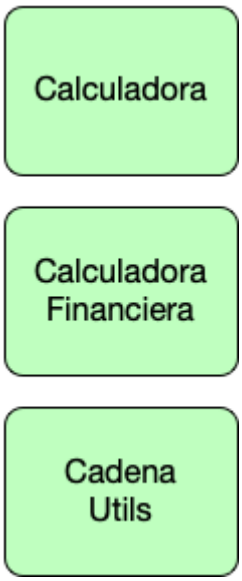
¿Qué es un JUnit Test Suite? . Un JUnit Test Suite es un conjunto de pruebas unitarias de JUnit agrupadas de una forma especifica para facilitar nuestro trabajo cotidiano . Esto en principio puede costar entenderlo un poco ya que las pruebas unitarias suelen estar perfectamente organizadas en grupos de paquetes. Vamos a ver unos ejemplos sencillos con tres clases.



Estas clases disponen de una serie de métodos sencillos para realizar calculos :

```
public class Calculadora {

  public static int sumar(int a, int b) {

    return a + b;
  }

  public static int restar(int a, int b) {

    return a - b;
  }
```

```
  public static int division(int a, int b) {

    return a / b;
  }

  public static int producto(int a, int b) {

    return a * b;
  }
}
```

```
package com.arquitecturajava.utilidades;

public class CalculadoraFinanciera {

  public static double media(int ...numeros) {
    double total=0;
    for ( int numero :numeros) {
      total+=numero;
    }
    return total/numeros.length;
  }
}
```

```
package com.arquitecturajava.utilidades;

public class CadenaUtils {

  public  static String sumaCadenas( String delimitador, String...
cadenas) {

    String total = "";
```

```
    for (String cadena : cadenas) {

      total += cadena+ delimitador;

    }
    return total;
  }
}
```

Como podemos ver cada clase tiene un pequeño grupo de métodos sobre los que podemos realizar una serie de pruebas unitarias.

```
package com.arquitecturajava.matematicas.test;



import static org.junit.Assert.assertEquals;

import org.junit.Test;

import com.arquitecturajava.utilidades.Calculadora;
import com.arquitecturajava.utilidades.CalculadoraFinanciera;

public class CalculadoraTest {

  @Test
  public void sumaTest() {
    assertEquals(4, Calculadora.sumar(2, 2));
  }
  @Test
  public void restaTest() {
    assertEquals(0, Calculadora.restar(2, 2));
```

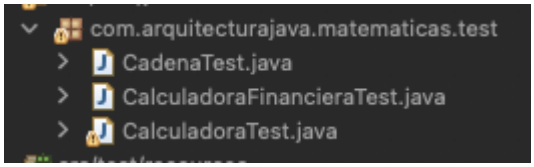
```
  }
  @Test
  public void productoTest() {
    assertEquals(0, Calculadora.restar(2, 2));
  }
  @Test
  public void divisionTest() {
    assertEquals(0, Calculadora.restar(2, 2));
  }

}
```

```
package com.arquitecturajava.matematicas.test;

import static org.junit.Assert.*;

import org.junit.Test;

import com.arquitecturajava.utilidades.CalculadoraFinanciera;

public class CalculadoraFinancieraTest {

  @Test
  public void mediaTest() {
    assertEquals(2, CalculadoraFinanciera.media(2, 2),0);
  }

}
```

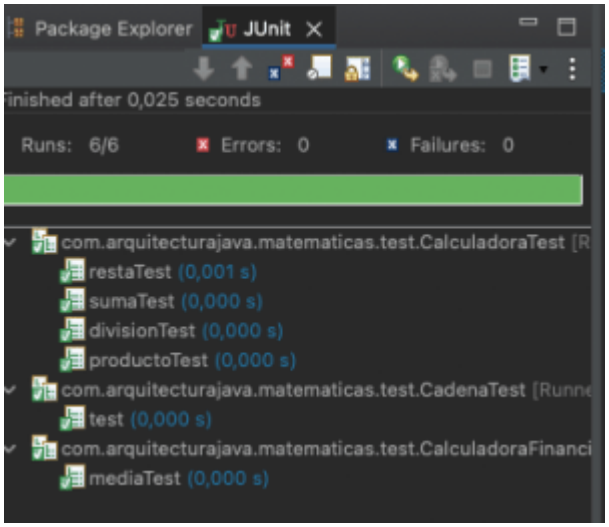
```
package com.arquitecturajava.matematicas.test;
```

```
import static org.junit.Assert.*;

import org.junit.Test;

import com.arquitecturajava.utilidades.CadenaUtils;

public class CadenaTest {

  @Test
  public void test() {
    assertEquals ("hola,hola,",CadenaUtils.sumaCadenas(","
,"hola","hola"));
  }

}
```

Estas clases nos permiten ejecutar las diferentes pruebas unitarias y se encuentran ubicadas en un package concreto “test”:



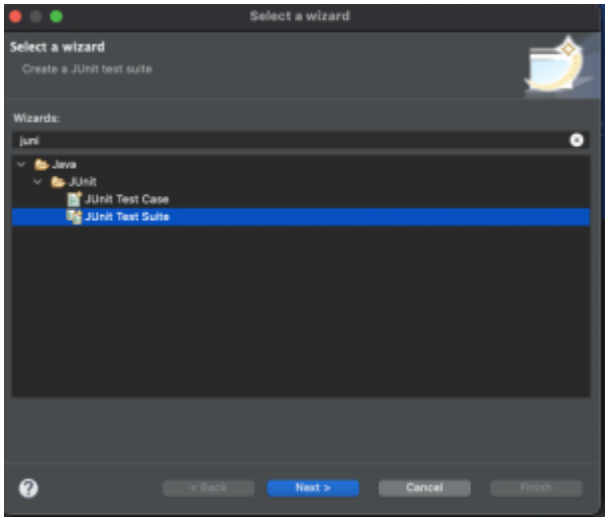
Esto nos permite ejecutar las pruebas unitarias por cada clase o ejecutar todas si pedimos a eclipse que nos ejecute todas las pruebas unitarias del Package:



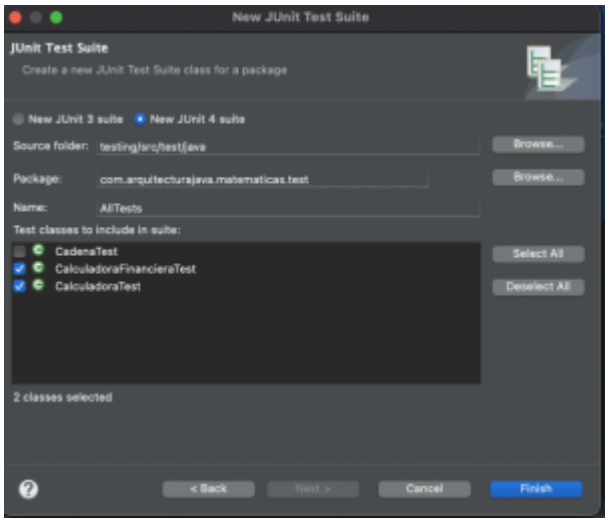
Sin embargo hay situaciones en las que nosotros podemos querer ejecutar un conjunto de pruebas “determinadas” y no todas . Por ejemplo en este caso nos puede interesar querer ejecutar las pruebas unitarias que están ligadas a las Calculadoras. Con la configuración que tenemos no nos es posible ya que o ejecutamos las de una clase o las de todo el Package.

JUnit Test Suite y agrupamiento

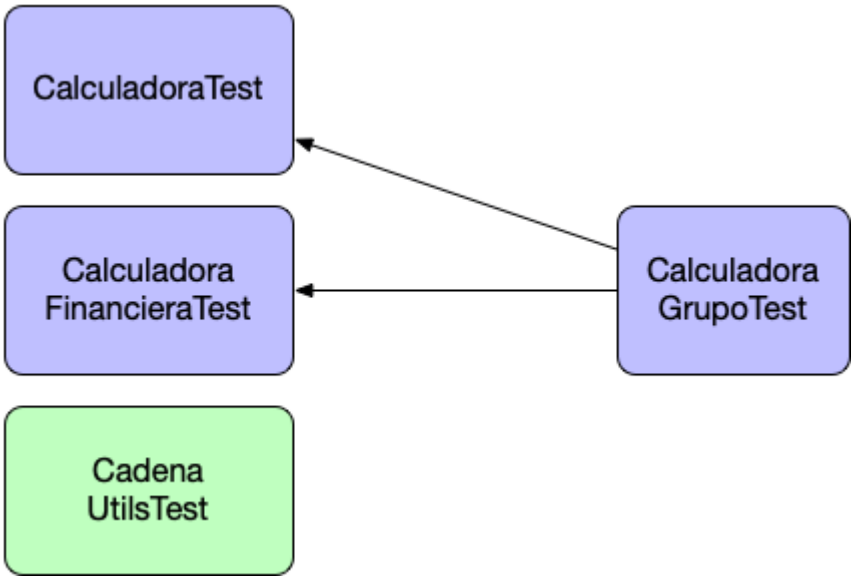
Si queremos poder agrupar los test de la forma que a nosotros nos encaje de forma más natural para su posterior ejecución podemos construir un JUnit Test Suite con Eclipse.



Este asistente nos permitirá elegir las clases que queremos agrupar la ejecutar las pruebas unitarias:



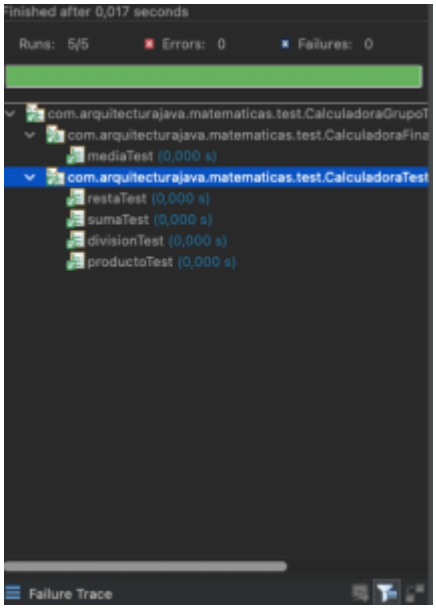
Una vez finalizado el asistente nos generará una nueva clase a la cual podemos asignar el nombre que deseemos y contendrá ese grupo de Test.



En este caso he decidido denominarle CalculadoraGrupoTest.

```
package com.arquitecturajava.matematicas.test;

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

@RunWith(Suite.class)
@SuiteClasses({ CalculadoraFinancieraTest.class, CalculadoraTest.class
})
public class CalculadoraGrupoTest {

}
```

Como podemos ver se genera una nueva clase que se encarga de agrupar los test de las Calculadoras



Conclusión

El uso de JUnit Test Suite siempre nos ayuda a organizar de forma más solida los grupos de test.

Otros Artículos relacionados

1. @Before vs @BeforeClass en JUnit
2. Spring Testing y el manejo de JUnit
3. Spring Boot Load Data y testing
4. Junit5