Pruevas unitarias en Java:

¿Que son? Una librería desarrollada para poder probar el funcionamiento de las clases y métodos que componen nuestra aplicación, y asegurarnos de que se comportan como deben ante distintas situaciones de entrada.

¿Cómo fucionan?

función de algún valor de entrada se evalúa el valor de retorno esperado, si la clase cumple la especificación, entonces JUnit devolverá que el método de la clase pasó exitosamente la prueba En caso de que el valor esperado sea diferente al retornado por el método durante la ejecución, JUnit devolverá un fallo en el método correspondiente.

JUnit es también un medio para controlar las prueba de regresión, necesarias cuando una parte del código ha sido modificado y se desea ver que el nuevo código cumple los requisitos anteriores y que no se ha alterado su funcionalidad después de la nueva modificación.

Este framework se encuentra en el momento de escribir el post en la versión 4.5, con algunas mejoras:

■Incluye soporte para anotaciones (Java 5 Annotations).

■Permite timeouts en los test.

■Configurar excepciones esperadas.

■Ordenación, priorización, categorización y filtrado de tests.

■Más tipos de aserciones (ej: assertEquals(Object[], Object[]).

■Se elimina la distinción entre errores (errors) y fallos (failures).

Ejemplo:

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import static org.junit.Assert.\*;

import codigo.Suma;

public class TestSuma {

private Suma suma = new Suma();

@Before

public void setUpClass() throws Exception {

suma.setA(4);

suma.setB(5);

}

@Test

public void testSuma(){

assertEquals(9,suma.sumar());

}

@After

public void tearDownClass() throws Exception {

}

}