Universidad interamericana de Panamá

Facultad de Ingeniería

Materia :

Programación II

Tema:

Prueba unitaria

Profesor:

Abdiel Martínez

Estudiante:

González Jeshua

[](http://2.bp.blogspot.com/-E9K6tZcSyZs/UbWAjTroucI/AAAAAAAAAzw/rhC51_lQD2U/s1600/junit.gif)  
Una **prueba unitaria** según Wikipedia es “una forma de probar el correcto funcionamiento de un módulo de código. Esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado.”. Para algunos, este tipo de pruebas es una pérdida de tiempo o una inutilidad. Hay que decir que crear las pruebas supone un **tiempo adicional** al que dedicamos a programar la aplicación en sí. Debemos valorar si merece la pena o no, según las dimensiones y la complejidad de nuestro proyecto. Lo cierto es que si las hacemos, dependiendo del caso, podemos ahorrarnos mucho tiempo, ya que habremos **automatizado**las pruebas e instantáneamente sabremos si nuestra aplicación cumple con los resultados correctos. Mejoraremos la calidad del software, asegurándonos de que aunque hagamos cambios en nuestro código, cumpliremos con los requisitos esperados. En un futuro podremos modificar, ampliar o eliminar código y las pruebas unitarias seguirán sirviéndonos para realizar las comprobaciones pertinentes.  
  
  
Para empezar, vamos a ver el ejemplo más simple posible, para que podamos centrarnos en la comprensión de lo que son las pruebas unitarias y de la herramienta para llevarlas a cabo: **JUnit**. El IDE que utilizaremos será Eclipse.

**1.**Empezaremos creando la clase que queremos probar. Tendrá el siguiente código:

public class Suma {

    private int num1;

   private int num2;

   public Suma(int n1, int n2) {

       num1 = n1;

       num2 = n2;

   }

   public int sumar() {

       int resultado = num1 + num2;

       return resultado;

   }

}

Como vemos, es una clase muy sencilla que dispone de un método que permite sumar dos números.   
  
**2.** Lo siguiente es crear la clase que nos servirá para probar la clase Suma. Queremos saber si la suma se hace correctamente en tres casos: sumando dos números positivos, sumando dos números negativos y sumando un número positivo y un número negativo. El código será el siguiente: 

public class SumaTest {

   @Test

   public void sumaPositivos() {

      System.out.println("Sumando dos números positivos ...");

       Suma S = new Suma(2, 3);

       assertTrue(S.sumar() == 5);

   }

   @Test

   public void sumaNegativos() {

       System.out.println("Sumando dos números negativos ...");

       Suma S = new Suma(-2, -3);

       assertTrue(S.sumar() == -5);

   }

   @Test

   public void sumaPositivoNegativo() {

       System.out.println("Sumando un número positivo y un número negativo ...");

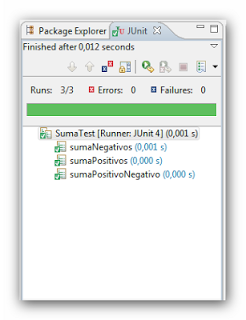
       Suma S = new Suma(2, -3);

       assertTrue(S.sumar() == -1);

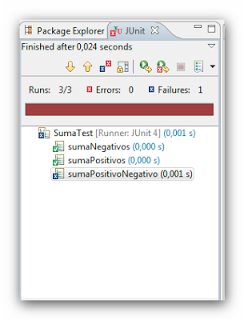
   }

}

**3.** Activamos la vista “JUnit” en Eclipse. Para ello hay que pulsar en: “Window”; “Show View”; “Other...”; “Java”; “JUnit”.   
  
**4.**Y ejecutamos el proyecto como test JUnit, pulsando sobre el proyecto con el botón derecho, luego pulsamos “Run as” y finalmente “JUnit Test”.   
Ahora podremos ver los resultados. Si todo hay ido bien, en la vista JUnit nos aparecerá ésto:

[](http://1.bp.blogspot.com/-V1cZL610nFg/UbA-YqYgwJI/AAAAAAAAAzY/3F64bxIoPnI/s1600/JUnit1.png)

Si ha habido algún fallo nos aparecerá lo siguiente:

[](http://4.bp.blogspot.com/-RUityjkazdk/UbA-cUQVe4I/AAAAAAAAAzg/IL9udB06Wq0/s1600/JUnit2.png)  
  
Como vemos, en el primer caso el test ha sido completado sin fallos. Sin embargo, en el segundo ejemplo (he modificado el código a conciencia para que falle) se ha encontrado un error en “sumaPositivoNegativo”.   
  
Estamos ante un ejemplo poco práctico, por su simplicidad. Pero lo importante es comprender el concepto de pruebas unitarias y aprender a manejar JUnit dentro de Eclipse.