Una Prueba Unitaria, es una forma de comprobar que nuestro código, hace lo que se supone debe hacer; es decir, se asegura que muestro código no presente fallos, errores, o cálculos inesperados, y que siempre retorne el valor correcto. Cada lenguaje de programación cuenta con sus propias herramientas para realizar estas pruebas, en el caso de java, tenemos a **JUnit***(creado por Erich Gamma y Kent Beck)*.

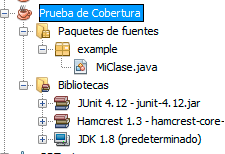
En este post realizaremos un test a un método.

Necesitamos:

* Netbeans 8.x y Java 8
* Librería JUnit 4.x *(Viene con Netbeans)*
* Librería Hamcrest 1.3 *(Viene con Netbeans)*

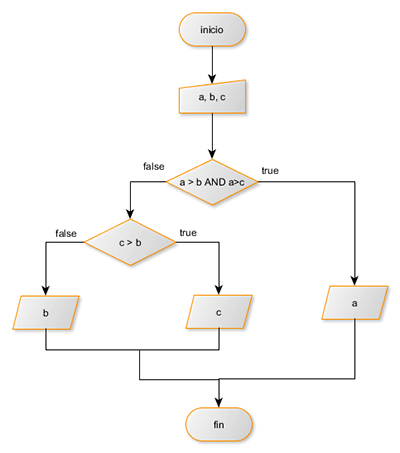
**Paso 1: El proyecto**

Creamos un proyecto que llamaremos ***«Prueba de Cobertura»***, agregamos un paquete «*example*» y las librerías arriba mencionadas. Para finalizar, creamos una clase «*Miclase.java*«.

[](https://www.jc-mouse.net/wp-content/uploads/2017/10/cobertura_java.gif)

**Paso 2. El código**

El algoritmo corresponde al problema de detectar el número mayor dados tres números.

[](https://www.jc-mouse.net/wp-content/uploads/2017/10/number_one_two_three.gif)



Pasamos del algoritmo a código y tenemos:

package example;

/\*\*

\* @see https://www.jc-mouse.net/

\* @author mouse

\*/

public class MiClase {

public int numero\_mayor(int a, int b, int c) {

if (a > b && a > c) {

return a;

} else if (c > b) {

return c;

} else {

return b;

}

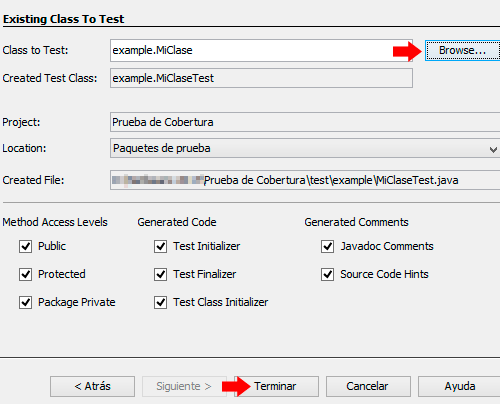
}

}

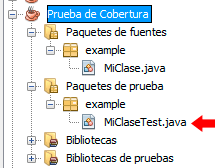
**Paso 3: Paquete de Prueba**

Ya que tenemos nuestra clase de prueba lista, procedemos a crear las *clases de prueba*. Clic derecho sobre nuestro proyecto -> New -> Other… En Categoría seleccionamos «***Unit Tests***» y en File Types, seleccionamos «***Test for Existing Class***» y presionamos el botón *siguiente*.

A continuación, presionando el botón «*browse*…» buscamos y seleccionamos la clase que testearemos, «*Miclase*«, dejamos todas las opciones tal cual están y finalizamos presionando el botón *«Terminar»*.

[](https://www.jc-mouse.net/wp-content/uploads/2017/10/test_unit_class.gif)

Se creara otra clase en la sección de «**Paquete de prueba**» como se ve a continuación

[](https://www.jc-mouse.net/wp-content/uploads/2017/10/paquete-de-prueba.gif)

**Paso 5: JUnit**

Antes de realizar la prueba con *JUnit*, debemos estudiar un poco el código generado. En la parte inferior de la clase *MiClaseTest* encontramos el siguiente método:

1 @Test

2 public void testNumero\_mayor() {

3 System.out.println("numero\_mayor");

4 int a = 0;

5 int b = 0;

6 int c = 0;

7 MiClase instance = new MiClase();

8 int expResult = 0;

9 int result = instance.numero\_mayor(a, b, c);

10 assertEquals(expResult, result);

11 // TODO review the generated test code and remove the default call to fail.

12 fail("The test case is a prototype.");

13 }

donde:

1) @Test: Los métodos marcados con esta anotación, le indican a JUnit que es código que queremos que se ejecute. Son estos métodos donde se implementan el código de pruebas.

4,5,6) Las variables del método que probaremos

7) Instancia a nuestra de prueba

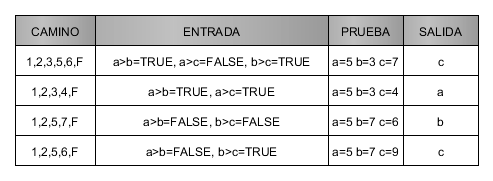
8) El resultado que esperamos obtener si la prueba tiene éxito

9) la llamada al método de prueba, el resultado se almacena en otra variable

10) assertEquals: Es uno de varios métodos con los que cuenta JUnit, este método en particular compara si dos objetos son iguales, de no ser así, lanzará una excepción y la prueba se detiene.

12) fail: Este método hace que la prueba termine con fallo

Como vimos en el paso anterior este algoritmo consta de 4 caminos posibles, eso quiere decir que debemos implementar 4 métodos de prueba.

[](https://www.jc-mouse.net/wp-content/uploads/2017/10/grafo-de-flujo-kawaii.gif)



Partiendo como base el método de prueba arriba mencionado, creamos 4 métodos con sus respectivos valores de entrada y valores esperados, es decir:

@Test

public void testNumero\_mayor\_caso1() {

int a = 5;

int b = 3;

int c = 7;

MiClase instance = new MiClase();

int expResult = 7;

int result = instance.numero\_mayor(a, b, c);

assertEquals(expResult, result);

}

@Test

public void testNumero\_mayor\_caso2() {

int a = 5;

int b = 3;

int c = 4;

MiClase instance = new MiClase();

int expResult = 5;

int result = instance.numero\_mayor(a, b, c);

assertEquals(expResult, result);

}

@Test

public void testNumero\_mayor\_caso3() {

int a = 5;

int b = 7;

int c = 6;

MiClase instance = new MiClase();

int expResult = 7;

int result = instance.numero\_mayor(a, b, c);

assertEquals(expResult, result);

}

@Test

public void testNumero\_mayor\_caso4() {

int a = 5;

int b = 7;

int c = 9;

MiClase instance = new MiClase();

int expResult = 9;

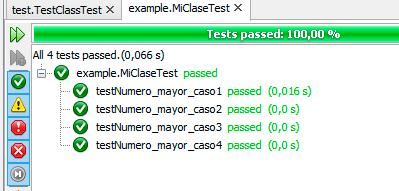
int result = instance.numero\_mayor(a, b, c);

assertEquals(expResult, result);

}

Para ejecutar el test con JUnit, clic derecho sobre la clase *MiClaseTest.java* -> **test file**

Observamos que pasamos las 4 pruebas con éxito.

[](https://www.jc-mouse.net/wp-content/uploads/2017/10/junit-test-java-example.gif)