ENTORNO DE DESARROLLO

Practica

Enero-Febrero 2019

Klever Uyana Barahona -Diseñador gráfico

Indice

PRUEBAS JUnit

¿Requisitos del trabajo a realizar?

[DISEÑO Y REALIZACIÓN DE PRUEBAS]

-Suponiendo que una empresa requiere de nuestros servicios como desarrolladores de un proyectos, para un fin determina.
-El proyecto de realizara de forma modular, a la hora de la repartición del trabajo se a decido que cada elemento del equipo realice 2 módulos (Procesos).
Para garantizar la corrección del desarrollo,se propone establecer la planificación de las pruebas.
Por lo qué hay que probar el software, seguir un orden concreto en la realización de pruebas.

Requisitos.

1.- Hay que programar al menos una clase Java. No se requiere método main
2.- Hay que programar al menos un método en la clase Java
3.- Hay que calcular la complejidad ciclomática del método que se programa
4.- Hay que indicar los casos de prueba
5.- Hay que hacer un plan de prueba, con resultados esperados y resultados obtenidos
6.-Hay que programar con Junit, indicando la asociación con el caso de prueba:
Al menos 2 test cases
Al menos 1 test suite
Se debe controlar que se pueda producir excepción en algún método

METODOS A REALIZAR

Método que calcule la suma de un conjunto de números negativos almacenados en memoria. Deberá de dar excepción si algún número es positivo.
Método que calcule el número primo n. Debe devolver excepción si se le pide un n > 10;

ANEXO I

La estructura de la documentación deberá contener al menos los siguientes puntos:
Análisis y Diseño
Definición del sistema
Definición del alcance del sistema
Establecimiento de requisitos
Catálogo de requisitos
Construcción
Complejidad ciclomática de McCabe
Conjunto de caminos básicos
Plan de pruebas
Casos de prueba con resultado esperado
Código fuente junit de cada test case (Caso de pruebas) y test suite

Análisis y Diseño
Método 9

Método que calcule la suma de un conjunto de números negativos almacenados en memoria.
Deberá de dar excepción si algún número es positivo.

Para la realización de este proceso debemos tener en cuenta los requisitos que debe cumplir.
Entre estos se encuentran.

(Requisitos establecidos)

-El método debe poder sumar números de un numero indefinido de numero.
-Los números sumados solo deben ser negativos.
-En el caso de que el usuario introduzca un numero positivo el método lanzara un mensaje.(Exception)

Para poder entender con mas claridad el Método, es preferible verlo desde Eclipse, es posible que al momento de importarlo ocurran error de reconocimiento en versiones anteriores, dado para poder realizar esta actividad allá tenido que recurrir a la versión mas actual.
Lo he creado en una clase llamada Ejercicio9MyMath

```java
package com.islas.filipinas.entornos.trabajo9;

import com.islas.filipinas.entornos.trabajos.DiesException;
/**
 * Classe que contiene el metodo de suma de numeros negativos
 * @author klever Uyana Barahona
 *
 */
public class Ejercicio9MyMath {
    /** Comenzamos lo que debe ralizar el modulo.
     * En este caso nos piden Método que calcule la suma de un conjunto de
     *  números negativos almacenados en memoria.
     *  Deberá de dar excepción si algún número es positivo.
     * @throws NueveException */
    public static double PI = Math.PI;
    /**
     * Metodo suma de los numero negativos
     * @param numeros - dados por el usuario
     * @return la suma de los numero negativos
     * @throws NueveException
     */
    //Para este Proceso se ha utilizado Array para Guardar la informacion.
    public static int sumaArrayNegativos(int[] numeros) throws NueveException {
        // TODO Auto-generated method stub
         int s=0;
        for(int i=0; i< numeros.length;i++) {
            if(numeros[i]>0)
                throw new NueveException("No se pueden numeros positivos.");
            else
            s=s+numeros[i];
        }
        return s;
    }
}
```

Propuesta de código.

MAIN.

Aunque no es imprescindible a la hora de realizar pruebas el método main, dado que se realizara con JUnit , he decido crearlo de todas formas, para poder así verificar su funcionamiento antes de hacer pruebas unitarias.

```
7  public class Main9Math {
8      /**
9       * Este es el metodo principal del ejercicio 9
10      * @param args
11      */
12     public static void main(String[] args) {
13         // TODO Auto-generated method stub
14         System.out.println("Introduce numeros negativos para sumarlos");
15         System.out.println("No introduzcas numeros positivos porfavor");
16         int[] numeros = {-1,-2,-3,-4,-5,};
17
18         int s;
19         try {
20             s = Ejercicio9MyMath.sumaArrayNegativos(numeros);
21             System.out.println("La Suma es"+ s);
22         } catch (NueveException e) {
23         System.out.println(e.getMessage()+" Que te no se puede¡¡¡¡");
24         }
25
26     }
27
```

Problems @ Javadoc Declaration Console Outline

```
<terminated> Main9Math [Java Application] C:\Program Files\Java\jre1.8.0_192\bin\javaw.exe (21 feb. 2019 1:33:38)
Introduce numeros negativos para sumarlos
No introduzcas numeros positivos porfavor
La Suma es-15
```

En este Primer RUN a la hora de definir los valores, se les dio valores acordes a lo que pedía el método.
Que en este caso son números negativos //-1,-2,-3,-4,-5
El resultado esperado es -15

```
1  package com.islas.filipinas.entornos.trabajo9;
2  /**
3   * Clase prinsipal del sistema ejercicio 9
4   * @author Klever
5   *
6   */
7  public class Main9Math {
8      /**
9       * Este es el metodo principal del ejercicio 9
10      * @param args
11      */
12     public static void main(String[] args) {
13         // TODO Auto-generated method stub
14         System.out.println("Introduce numeros negativos para sumarlos");
15         System.out.println("No introduzcas numeros positivos porfavor");
16         int[] numeros = {-1,-2,-3,-4,-5,5};
17
18         int s;
19         try {
20             s = Ejercicio9MyMath.sumaArrayNegativos(numeros);
21             System.out.println("La Suma es"+ s);
22         } catch (NueveException e) {
23         System.out.println(e.getMessage()+" Que te no se puede¡¡¡¡");
24         }
25
26     }
```

Problems @ Javadoc Declaration Console Outline

```
<terminated> Main9Math [Java Application] C:\Program Files\Java\jre1.8.0_192\bin\javaw.exe (21 feb. 2019 1:45:34)
Introduce numeros negativos para sumarlos
No introduzcas numeros positivos porfavor
No se pueden numeros positivos. Que te no se puede¡¡¡¡
```

En el siguiente RUN se a introducido un 6
Por lo que la Exception sera posible visualizarla gracias a try{.....}cath
En este caso se espera el Mensaje de la exception.

Exception

Para poder crear una exception, se a tenido que crear una clase aparte, en mi Proyecto lo de definido de la siguiente forma

```java
//Para este Proceso se ha utilizado Array para Guardar la imformacion.
public static int sumaArrayNegativos(int[] numeros) throws NueveException {
    // TODO Auto-generated method stub
     int s=0;
    for(int i=0; i< numeros.length;i++) {
        if(numeros[i]>0)
            throw new NueveException("No se pueden numeros positivos.");
```

En la clase del metodo

```java
package com.islas.filipinas.entornos.trabajo9;
/**
 * En esta clase se define la clase Exception del ejecicio 9
 * @author klever
 *
 */
public class NueveException extends Exception {
    public NueveException(String m) {
        super(m);
    }
}
```

Clase Exception

Catálogo de requisitos

Lo mas básico del método se deben de cumplir .

Sumar números negativos
Guardar los números sumados
Exception (si hay alguno positivo)

En resumen este método solo deja sumar un conjunto de numero negativos.

PRUEBAS JUnit

Complejidad ciclomática de McCabe

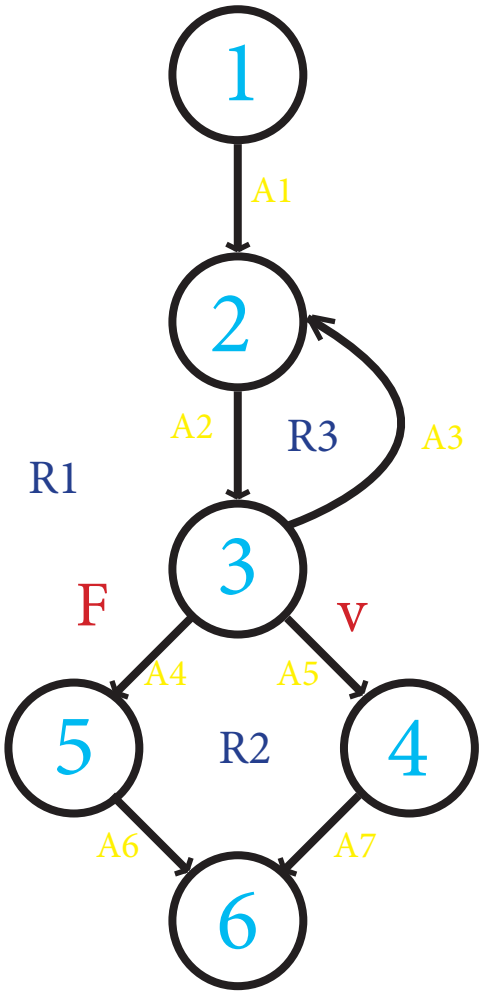
Este es el apartado donde mas tiempo le di, dado que, quería conseguirlo de una forma que se vio en clase . Utilizando un software en especifico que me ayude en esta labor. Ademas de Hacerlo de forma normal. En un principio lo estaba intentando con Visual Studio pero no encontraba la manera ni el pluing necesario. Busque en muchos foros leí muchos post pero nada me servia.
Hasta que por allí en una perfil de github di con mi respuesta, se trata de un programa que ya tiene muchos años llamado R esource S tandard M etrics de Msquare Technologies, genera un archivo html similar a javadoc
http://msquaredtechnologies.com/RSM-Wizard.html

Resultados del Modulo 9

```
------------------------------------------------------------------------

Función : com.islas.filipinas.entornos.trabajo9.Ejercicio9MyMath.sumaArr
Parámetros
          ayNegativos : (int [] numeros)
   Complejidad Ciclomática Vg Detalle
    Base de funciones: 1
    Bucles para / foreach: 1
    Condicional si / else if: 1
Complejidad    Parámetros 1        Retorno 1       Ciclo Vg 3        Total         5
 LOC 10        eLOC 8         lLOC 5         Comentario 12        Líneas        11

------------------------------------------------- ----------------------

                      ~~ Total File Summary ~~

 LOC 15        eLOC 12        lLOC 8         Comentario 17        Líneas        33
------------------------------------------------- ----------------------

                    ~~ Resumen Funcional de Archivo ~~

 Cuenta de Función de Archivo ....:           1
 Función Total LOC .....:          10   Función Total Pts LOC:         0.3
 Función Total eLOC ....:           8   Función Total Pts eLOC:        0.2
 Función Total lLOC ....:           5   Pts de la función total lLOC:         0.2
 Parámetros de la función total:           1   Retorno de la función total.:           1
 Complejidad total del ciclo:           3   Complejo de la función total .:           5
       ------ ----- ----- ------ ------ -----
Función máx. LOC ......:          10   Función promedio LOC ..:       10.00
 Función Max eLOC .....:           8   Función promedio eLOC.:        8.00
 Función Max lLOC .....:           5   Función promedio lLOC.:        5.00
       ------ ----- ----- ------ ------ -----
Parámetros de función máx .:           1   parámetros de función prom.:        1.00
 devoluciones de función máx .: Devoluciones de función avg.           1   : complejo de interfaz máx .        1.00
 . :           2   Avg Interface Complex. :        2,00
 Max ciclomática Complex .:           3   Promedio ciclomática Complex .:        3,00
 Max total Complejidad ..:           5   Promedio de la Complejidad ..:        5,00
------------------------------------------------------------------------
Fin del archivo: E: \ trabajo de Junit \ proyecto \ KleverUyana_Junit \ src \ com \ isla
            s \ filipinas \ entornos \ trabajo9 \ Ejercicio9MyMath.java
```

Complejidad ciclomática de McCabe
Modulo 9

En un principio la complejidad ciclomática que yo había realizado tenia esta forma.
No sabia si realmente lo tenia bien .



Aristas=7
Regiones=3
Nodos=6

1

V(G)	Aristas-Nodos+2
V(G)	7-6+2=3

2

V(G)	Nodos predicado+1
V(G)	2+1=3

4

V(G)	Regiones
V(G)	3

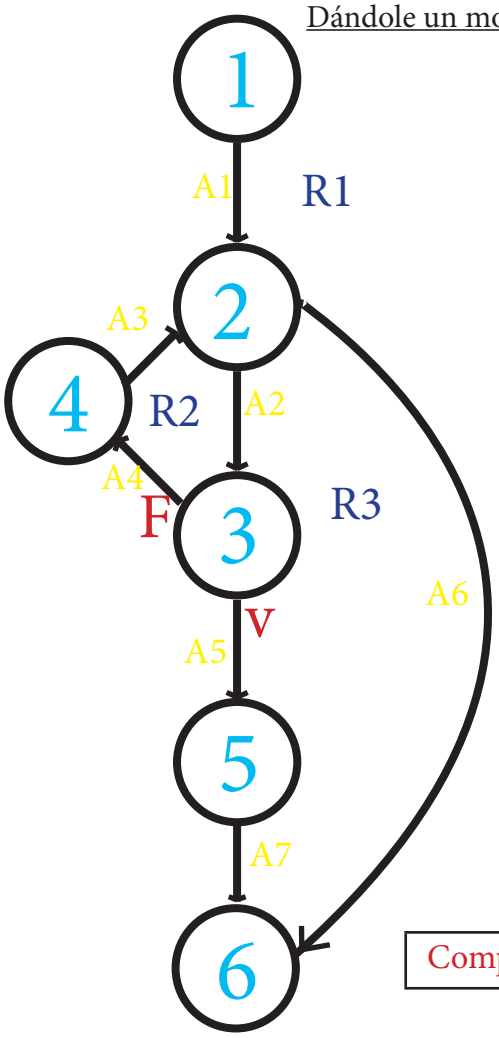
Pruebas de caja Blaca - Caminos Básicos

Camino	Entrada	Salida
1-2-3-5-6	b=F	Suma
1-2-3-4-6	b=V	Exception

Complejidad ciclomática = 3	Programa Sencillo sin mucho riesgo.

Complejidad ciclomática de McCabe
Modulo 9 Con ayuda

Dado que no tenia la suficiente experiencia para realizar este ejercicio en aquel momento, decido pedir ayuda a una compañera de clase (Maria E.) que tenia el mismo modulo que yo.
Al momento de discutirlo entre nosotros, lo teníamos muy similar , pero había cosas que no nos llegaba a convencer, como si un FOR es o no un Nodo Predicado, ya que según lo pone en libro, un nodo predicado es aquel de que salen dos aristas, y en un FOR sale una y llega una para cerra el bucle, entre otros.
Dándole un montón de vueltas llegamos a esta solución.



Aristas=7
Regiones=3
Nodos=6

1

V(G)	Aristas-Nodos+2
V(G)	7-6+2=3

2

V(G)	Nodos predicado+1
V(G)	2+1=3

4

V(G)	Regiones
V(G)	3

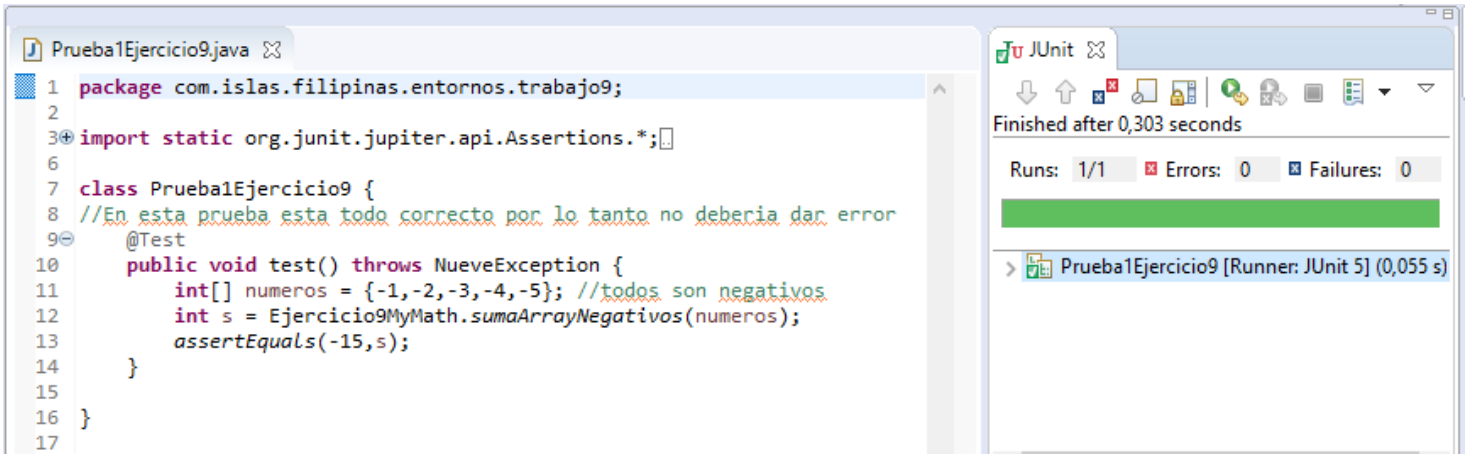
Complejidad ciclomática = 3	Programa Sencillo sin mucho riesgo.

Pruebas de caja Blaca - Caminos Básicos

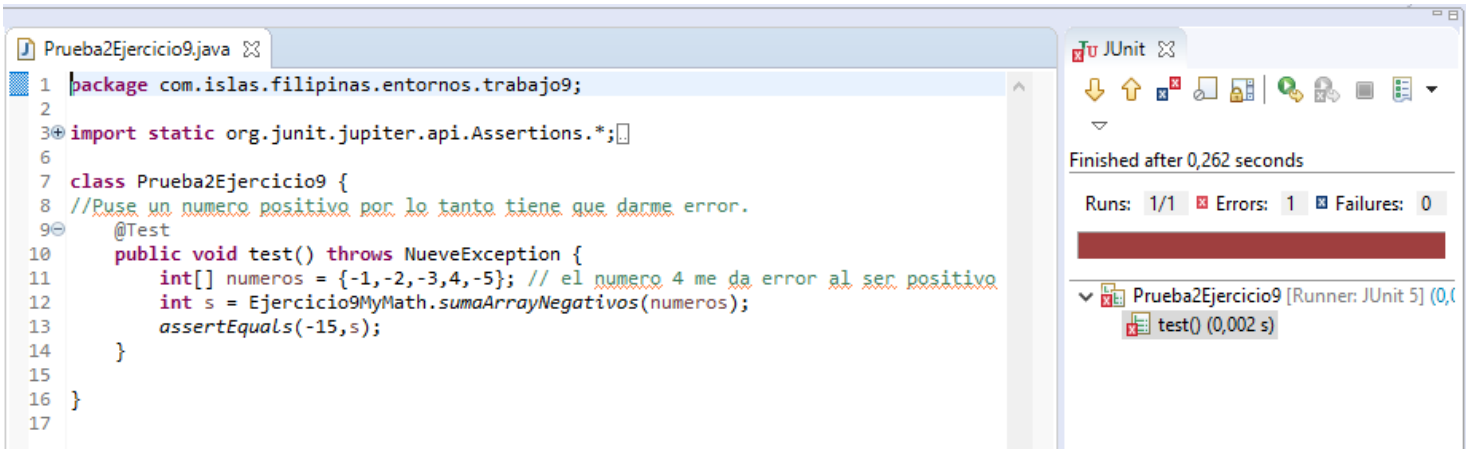
Camino	Entrada	Salida
1-2-3-4-2-6	a=F	Suma
1-2-3-5-6	b=V	Exception
1-2-3-4-6	?	En este camino lo que hace es que si a inicio de FOR (i) le damos un valor de 5, al ser mayor que lo que hemos puesto no va entrar en el FOR.

Pruebas con JUnit.

Se me complico la cosa lo que en principio era una tarea fácil resulto ser una odisea.



En mi primera prueba no tuve gran dificultad Introduje un comparación de resultados y listo ,
Suma de los datos predefinidos(-1,-2,-3-,-4,-5) = Respuesta esperada (-15)
Todo correcto
Lo que me dio guerra como mas tarde se vera es la parte de las Exception y a la hora de hacer el Suit.



En prueba 2 introduje un error, para que fallara la respuesta por lo tanto ya esperaba que ocurriera ya que si no convocas a la Exception. class es lo normal
Introduje un numero positivo (4)

Pruebas con JUnit.

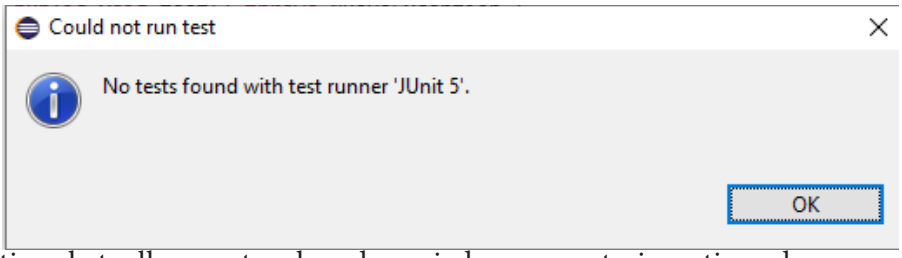
Es en este punto donde se complico todo.

*Prueba3Ejercicio9.java Prueba2Ejercicio9.java Prueba1Ejercicio9.java

```java
package com.islas.filipinas.entornos.trabajo9;

import static org.junit.jupiter.api.Assertions.*;

class Prueba3Ejercicio9 {
// al poner expected=NueveException.class, la prueba deberia dar positiva.
    //al parecer no lo detecta por que lo estoy haciendo con Junit5 y el expected en del Junit4
    //probare haciendolo en el Junit4 pero no se si puede funcionar
    @Test(expected=NueveException.class)
    public void test() throws NueveException {
        int[] numeros = {-1,-2,-3,4,-5}; // el numero 4 me da error al ser positivo
        int s = Ejercicio9MyMath.sumaArrayNegativos(numeros);

```

Tarde demasiado tiempo intentando resolver esto, ademas lo tenia en los dos proyectos.

```java
package com.islas.filipinas.entornos.trabajo9;

import static org.junit.Assert.*;

class Prueba4Ejercicio9 {
    // Lo he echo en Junit4 y al parecer si lo detecta pero al darle run no me va
    // desconosco el porque
        @Test(expected=NueveException.class)
        public void test() throws NueveException {
            int[] numeros = {-1,-2,-3,4,-5}; // el numero 4 me da error al ser positivo
            int s = Ejercicio9MyMath.sumaArrayNegativos(numeros);
        }
        //he estado buscando en foros y al paracer tengo que cambier el tipo de run
        //Run Configuration
}
```

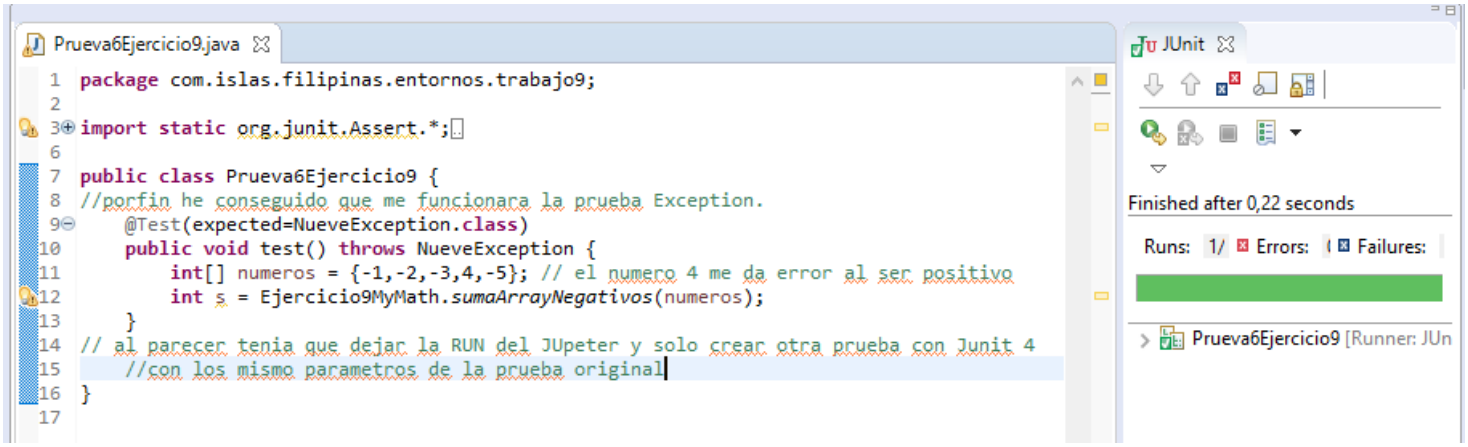


Estaba a punto de tiran la toalla pero tras leer demasiados comentarios y tipos de respuestas ,de como declarar una Expection en Junit5 no intente hacer. Pero la cosa no salio bien

```java
package com.islas.filipinas.entornos.trabajo9;

import static org.junit.jupiter.api.Assertions.*;

class Prueba5Ejercicio9 {
// he intentado hacerlo con el Junit 5 las Exception pero no he podido.
    @Test
    void testNueveException() {
    Thowable exception = assetThrows(NueveException.class,() ->{
    throw new NueveExpection("dhdu");
    });

```

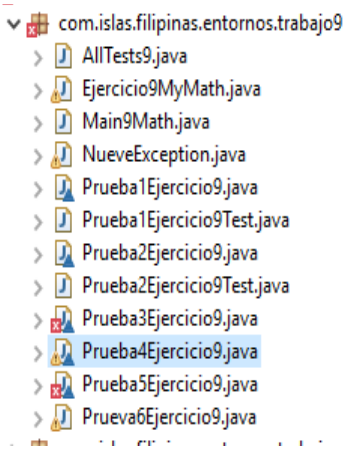
Pruebas con JUnit.

Al final lo pude realizar a tiempo
Al parecer tenia un error de reconocimiento de características de declaración entre versiones (Junit4-Junit5) ademas el tipo de RUN Junit4 RUN Jumpeter.

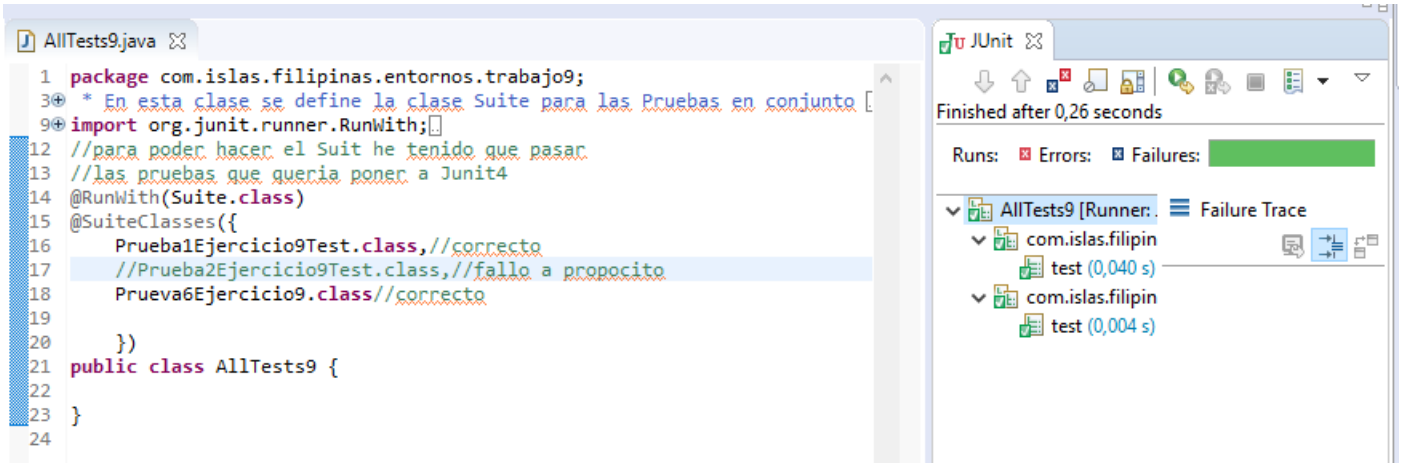


Pruebas con SUITE.

Dado que el Suit solo lo realiza Junit4, y las pruebas(1,2) la había hecho en Junit5 tuve que volver a realizarlas

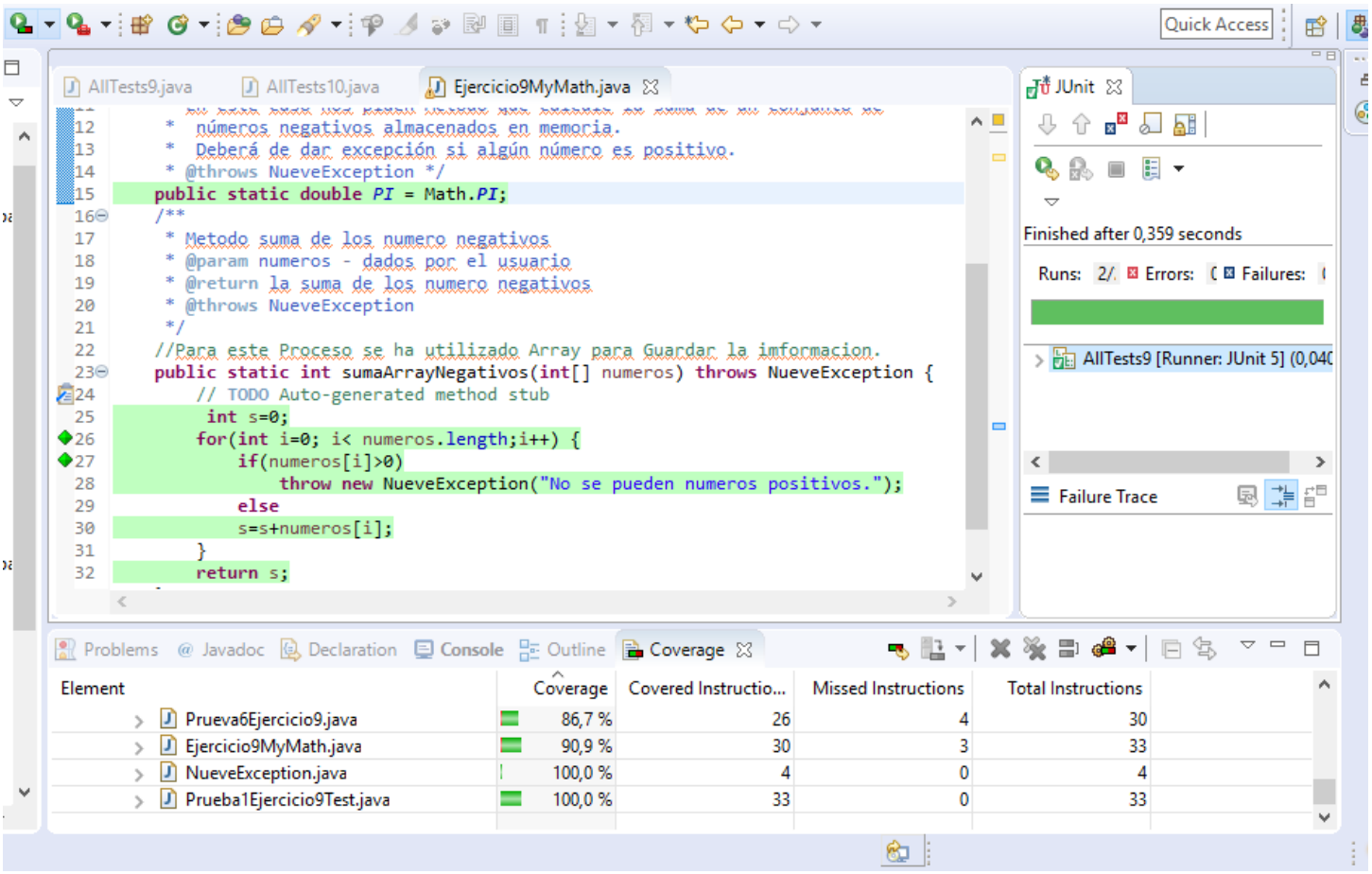


Como se había esperado, las pruebas conjuntas estaban bien ejecutadas.



CAVERAGE

TODO BIEN, SIN PROBLEMA.



PRUEBAS JUnit

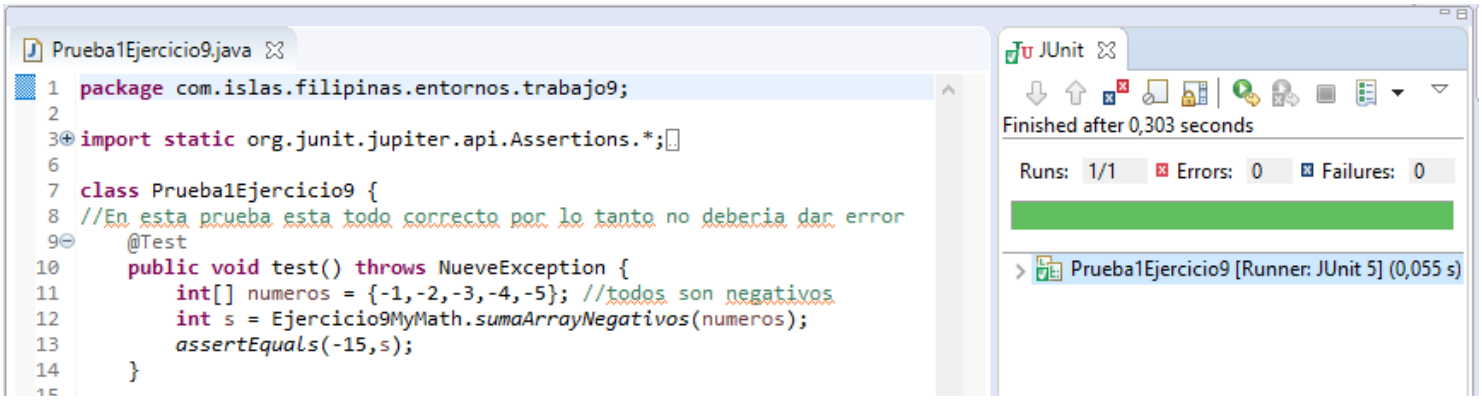
Plan de pruebas

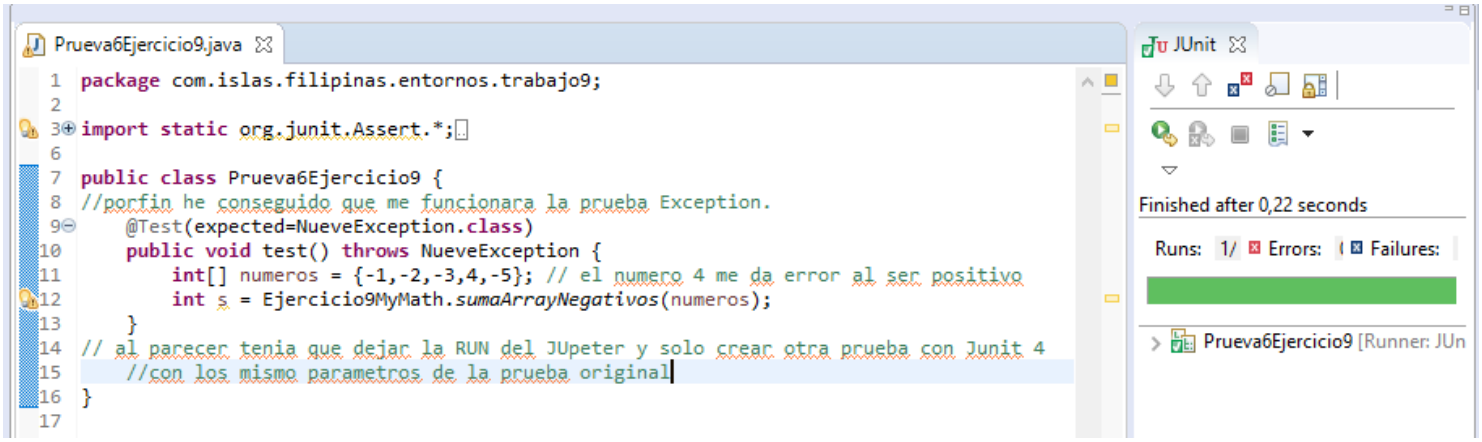
Para poder realizar el plan de pruebas tuve que tomar como referencia los caminos de pruebas de caja Blanca o caminos básicos.

Pruebas de caja Blaca - Caminos Básicos

Camino	Entrada	Salida
1-2-3-4-2-6	a=F	Suma
1-2-3-5-6	b=V	Exception
1-2-3-4-6	?	En este camino lo que hace es que si a inicio de FOR (i) le damos un valor de 5, al ser mayor que lo que hemos puesto no va entrar en el FOR.

NªPrueba	Nombre de Prueba	Datos dados	Resultados Esperados	Resultados Optenidos
1ª	Prueba1Ejercicio9Test	-1,-2,-3,-4,-5	-15	-15
2ª	Prueva6Ejercicio9	-1,-2,-3,4,-5	Exception	Exception





Reflexiones

Al parecer la parte de pruebas es una de las mas importantes a la hora de llevar un proyecto a cabo, por eso hay que tener conocimiento sobre como realizarlo y el porque de las mismas.

Me ha venido bien tener fallos dado que de esa manera me involucro mas a la hora de sacar el trabajo a tiempo , así como de aprender a tener un plan B para ciertas ocasiones.

En conclucion me a gustado realizar este trabajo así como el de GIT Y Github(GITLAB)

Habia pensado generar un Java.doc para este trabajo
pero ya veré como hacerlo.