



ESCUELA SUPERIOR  
PÓLITECNICA DEL LITORAL

# **WORKSHOP ABOUT EMPIRICAL SOFTWARE TESTING**

## **AUTORES:**

- AGUILAR MORA OSWALDO
- BERMUDEZ MOREIRA KAREN
- BERNAL MOREIRA GUILLERMO
- ORTIZ HOLGUIN EDUARDO
- WONG PAVON HUGO

**MATERIA:** ING. SOFTWARE II

**TUTOR:** DR. MERA CARLOS

**ENTREGA:** 2020/06/11

## 1. Resumen

El presente documento contiene el reporte técnico correspondiente al primer taller grupal de nombre **"WORKSHOP ABOUT EMPIRICAL SOFTWARE TESTING"** del **GRUPO#4** perteneciente al curso de **INGENIERIA DE SOFTWARE II** del **PAO I**.

El reporte contiene una descripción del taller, el pseudocódigo, el planteamiento de las pruebas, la implementación en JAVA y el resultado de las pruebas.

## 2. Descripción

Se presenta la problemática de triángulo junto a un pseudocódigo que da solución a la identificación del tipo de triángulo, o si los valores ingresados no forman dicha figura, con sus respectivas validación, se nos pide en este taller implementar el pseudocódigo, diseñar los casos de pruebas que se consideren necesarios y finalmente ejecutar dichos casos de prueba.

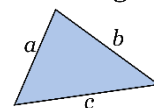
### 2.1. Problema del triángulo

Es uno de los problemas más usados para la enseñanza de **pruebas de software**, consiste en un programa que debe tomar como entrada tres valores numéricos enteros que corresponden a los lados de un triángulo, debe evaluar las condiciones necesarias y finalmente determinar si los valores ingresados corresponden a un triángulo **escaleno**, a uno **equilátero**, uno **isósceles** o en última instancia los valores no forman un triángulo.

Para determinarlo se tienen las siguientes condiciones:

- Los valores ingresados deben estar en el rango de [1,200]
- Los valores deben cumplir con la **desigualdad triangular**.
- La igualdad de los lados determina si son equiláteros, escalenos o isósceles.

Desigualdad  
del triángulo



$$\begin{aligned}a + b &> c \\b + c &> a \\c + a &> b\end{aligned}$$

Gráfico 1.- Desigualdad

### 2.2. Implementación

Se pide realizarla implementación usando como lenguaje **JAVA** adicionalmente emplear alguna herramienta para realizar los test y la herramienta de desarrollo colaborativo **Git**.

### 3. Pseudocódigo

```
Program triangle'
Dim a, b, c As Integer
Dim c1, c2, c3, IsATriangle As Boolean

'Step 1: Get Input
Do
    Output("Enter 3 integers which are sides of a triangle")
    Input(a, b, c)
    c1 = (1 ≤ a) AND (a ≤ 200)
    c2 = (1 ≤ b) AND (b ≤ 200)
    c3 = (1 ≤ c) AND (c ≤ 200)
    If NOT(c1)
        Then Output("Value of a is not in the range of permitted values")
    EndIf
    If NOT(c2)
        Then Output("Value of b is not in the range of permitted values")
    EndIf
    If NOT(c3)
        ThenOutput("Value of c is not in the range of permitted values")
    EndIf
Until c1 AND c2 AND c3
Output("Side A is",a)
Output("Side B is",b)
Output("Side C is",c)

'Step 2: Is A Triangle?
If (a < b + c) AND (b < a + c) AND (c < a + b)
    Then IsATriangle = True
    Else IsATriangle = False
EndIf

'Step 3: Determine Triangle Type
If IsATriangle
    Then If (a = b) AND (b = c)
        Then Output ("Equilateral")
    Else If (a ≠ b) AND (a ≠ c) AND (b ≠ c)
        Then Output ("Scalene")
    Else Output ("Isosceles")
    EndIf
EndIf
Else Output("Not a Triangle")
EndIf
End triangle
```

*Gráfico 2.- Pseudocódigo Problemática Triangulo*

### 4. Diseño de Casos de Pruebas

Se ha decidido dividir las pruebas en las siguientes categorías:

- Pruebas de Tipo de Datos

- Pruebas de Rango de Valores
- Pruebas de Resultados

#### 4.1. Pruebas de Tipo de Datos

En estas pruebas se evaluará la **robustez** del programa, es decir si no se cae debido a los valores ingresados y si notifica al usuario del error. Se probarán con valores no enteros ya sean de punto flotante o cadenas de texto.

#### 4.2. Pruebas de Rango de Valores

En estas pruebas se evaluará la correcta validación de las variables, si se encuentran en el rango designado entre **1 a 200**, si muestran los mensajes correctos.

#### 4.3. Pruebas de Resultados

En estas pruebas se evaluará los resultados finales luego de pasar las validaciones previas, se usarán valores correctos y se verificarán los resultados entre los 4 casos posibles (**Equilátero, Escaleno, Isósceles, No-Triángulo**).

#### 4.4. Detalle de Casos de pruebas planteados

# Caso	Comentario	a	b	c	Resultado
<b>Pruebas de Tipo de Datos</b>					
<b>1</b>	String	30	b	2	<b>Error 1</b>
<b>2</b>	String	sk	1	Sd	<b>Error 1</b>
<b>3</b>	Decimales	21	5	3,4	<b>Error 1</b>
<b>4</b>	Decimales	5,4	2,1	199,2	<b>Error 1</b>
<b>5</b>	Mezcla	4	7,5	abc	<b>Error 1</b>
<b>Pruebas de Rango de Valores</b>					
<b>6</b>	Excedente a	274	12	97	<b>Error 2</b>
<b>7</b>	Excedente b	76	599	3	<b>Error 2</b>
<b>8</b>	Excedente c	55	55	201	<b>Error 2</b>
<b>9</b>	Insuficiente a	0	2	2	<b>Error 2</b>
<b>10</b>	Insuficiente b	12	-45	90	<b>Error 2</b>

<b>11</b>	Insuficiente c	124	125	-1	<b>Error 2</b>
<b>12</b>	Mezcla 1	156	500	-12	<b>Error 2</b>
<b>13</b>	Mezcla 2	0	-999	999	<b>Error 2</b>
<b>Pruebas de Resultados</b>					
<b>14</b>	Equilátero 1	5	5	5	Equilátero
<b>15</b>	Equilátero 2	200	200	200	Equilátero
<b>16</b>	Escaleno 1	5	3	7	Escaleno
<b>17</b>	Escaleno 2	70	120	170	Escaleno
<b>18</b>	Isósceles 1	6	3	6	Isósceles
<b>19</b>	Isósceles 2	132	132	140	Isósceles
<b>20</b>	No triangulo 1	6	13	6	No triangulo
<b>21</b>	No triangulo 2	40	24	199	No triangulo
<b>22</b>	No triangulo 3	1	2	3	No triangulo

## 5. Código Fuente

### 5.1. Repositorio

Se uso **Github** como herramienta de colaboración para el desarrollo del taller, el vínculo es el siguiente:

- [https://github.com/kbermude/Taller1\\_Software](https://github.com/kbermude/Taller1_Software)

### 5.2. Consideraciones de Desarrollo

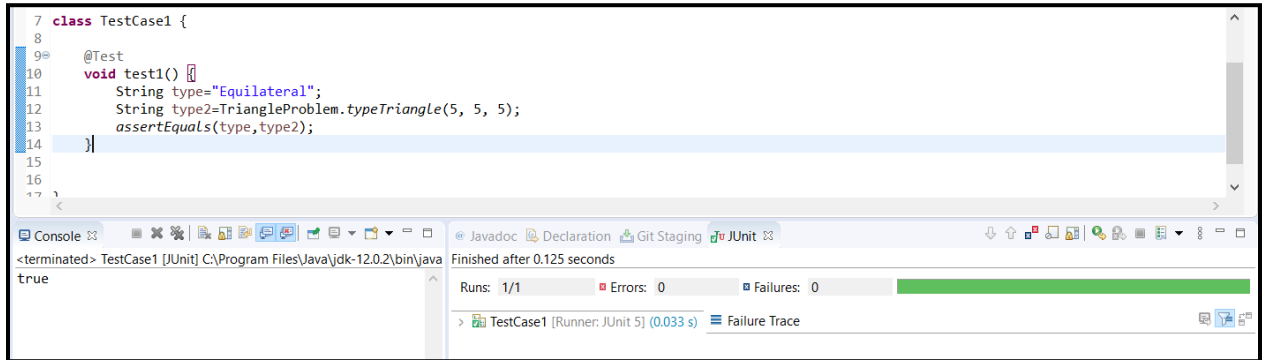
Para el desarrollo de la actividad y su implementación se agregaron los siguientes puntos:

- **Eclipse** se usó como IDE de Java para esta actividad.
- La herramienta **JUnit** se usó para realizar las pruebas.

## 6. Testeo y Resultados

# Caso	Comentario	a	b	c	Resultado Teórico	Resultado de prueba
<b>Pruebas de Tipo de Datos</b>						
<b>1</b>	String	30	b	2	<b>Error 1</b>	<b>Caída</b>
<b>2</b>	String	sk	1	Sd	<b>Error 1</b>	<b>Caída</b>
<b>3</b>	Decimales	21	5	3,4	<b>Error 1</b>	<b>Caída</b>
<b>4</b>	Decimales	5,4	2,1	199,2	<b>Error 1</b>	<b>Caída</b>
<b>5</b>	Mezcla	4	7,5	abc	<b>Error 1</b>	<b>Caída</b>
<b>Pruebas de Rango de Valores</b>						
<b>6</b>	Excedente a	274	12	97	<b>Error 2</b>	<b>Error 2</b>
<b>7</b>	Excedente b	76	599	3	<b>Error 2</b>	<b>Error 2</b>
<b>8</b>	Excedente c	55	55	201	<b>Error 2</b>	<b>Error 2</b>
<b>9</b>	Insuficiente a	0	2	2	<b>Error 2</b>	<b>Error 2</b>
<b>10</b>	Insuficiente b	12	-45	90	<b>Error 2</b>	<b>Error 2</b>
<b>11</b>	Insuficiente c	124	125	-1	<b>Error 2</b>	<b>Error 2</b>
<b>12</b>	Mezcla 1	156	500	-12	<b>Error 2</b>	<b>Error 2</b>
<b>13</b>	Mezcla 2	0	-999	999	<b>Error 2</b>	<b>Error 2</b>
<b>Pruebas de Resultados</b>						
<b>14</b>	Equilátero 1	5	5	5	Equilátero	Equilátero
<b>15</b>	Equilátero 2	200	200	200	Equilátero	Equilátero
<b>16</b>	Escaleno 1	5	3	7	Escaleno	Escaleno
<b>17</b>	Escaleno 2	70	120	170	Escaleno	Escaleno
<b>18</b>	Isósceles 1	6	3	6	Isósceles	Isósceles
<b>19</b>	Isósceles 2	132	132	140	Isósceles	Isósceles
<b>20</b>	No triangulo 1	6	13	6	No triangulo	No triangulo
<b>21</b>	No triangulo 2	40	24	199	No triangulo	No triangulo
<b>22</b>	No triangulo 3	1	2	3	No triangulo	No triangulo

## 7. Evidencia



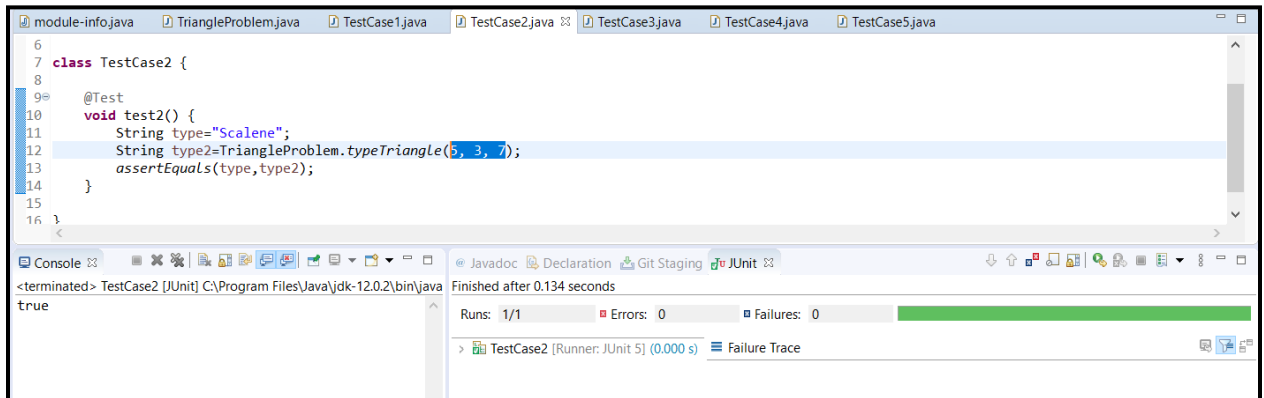
The screenshot shows an IDE window with the following code in `TestCase1.java`:

```
7 class TestCase1 {  
8  
9     @Test  
10    void test1() {  
11        String type="Equilateral";  
12        String type2=TriangleProblem.typeTriangle(5, 5, 5);  
13        assertEquals(type,type2);  
14    }  
15  
16  
17 }
```

The JUnit test results are displayed below the code editor:

- Console: `<terminated> TestCase1 [JUnit] C:\Program Files\Java\jdk-12.0.2\bin\java`  
`true`
- JUnit: Finished after 0.125 seconds  
Runs: 1/1 Errors: 0 Failures: 0  
> TestCase1 [Runner: JUnit 5] (0.033 s) Failure Trace

Gráfico 3.- Test #14 (5,5,5)



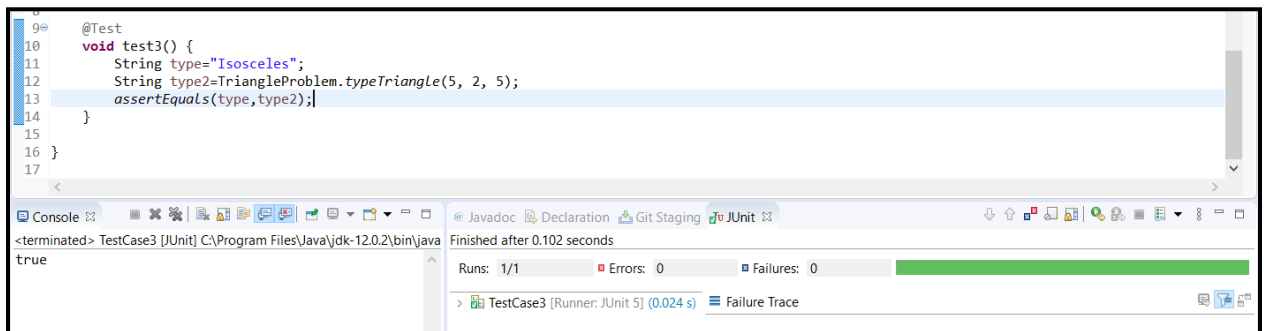
The screenshot shows an IDE window with the following code in `TestCase2.java`:

```
6  
7 class TestCase2 {  
8  
9     @Test  
10    void test2() {  
11        String type="Scalene";  
12        String type2=TriangleProblem.typeTriangle(5, 3, 7);  
13        assertEquals(type,type2);  
14    }  
15  
16 }
```

The JUnit test results are displayed below the code editor:

- Console: `<terminated> TestCase2 [JUnit] C:\Program Files\Java\jdk-12.0.2\bin\java`  
`true`
- JUnit: Finished after 0.134 seconds  
Runs: 1/1 Errors: 0 Failures: 0  
> TestCase2 [Runner: JUnit 5] (0.000 s) Failure Trace

Gráfico 4.- Test #16 (5,3,7)



The screenshot shows an IDE window with the following code in `TestCase3.java`:

```
9  
10    @Test  
11    void test3() {  
12        String type="Isosceles";  
13        String type2=TriangleProblem.typeTriangle(5, 2, 5);  
14        assertEquals(type,type2);  
15    }  
16  
17 }
```

The JUnit test results are displayed below the code editor:

- Console: `<terminated> TestCase3 [JUnit] C:\Program Files\Java\jdk-12.0.2\bin\java`  
`true`
- JUnit: Finished after 0.102 seconds  
Runs: 1/1 Errors: 0 Failures: 0  
> TestCase3 [Runner: JUnit 5] (0.024 s) Failure Trace

Gráfico 5.- Test #18 (5,2,5)

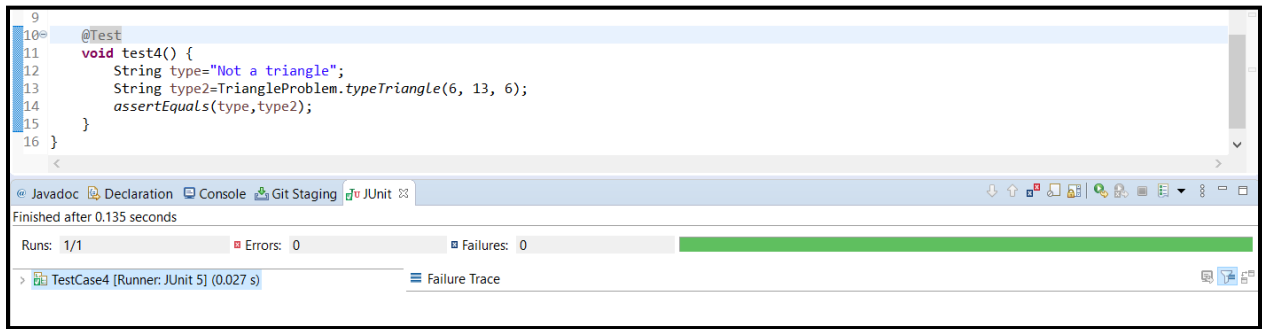


Gráfico 6.- Test #20 (6,13,6)

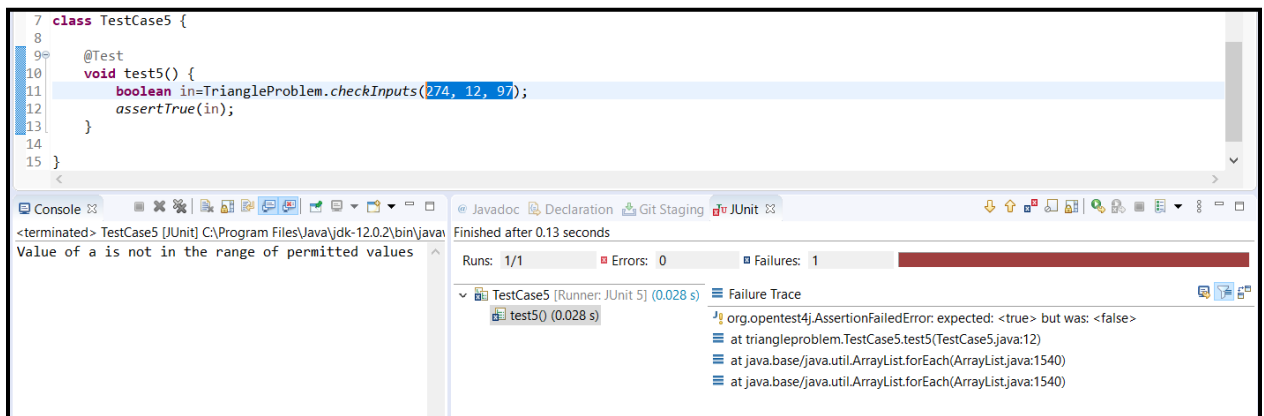


Gráfico 7.- Test #6 (274,12,97)

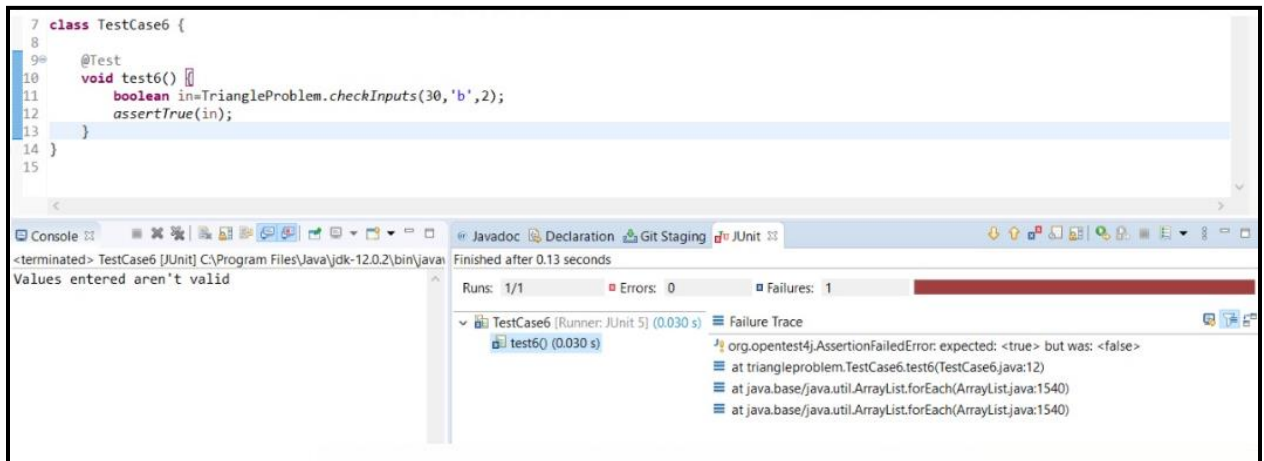


Gráfico 8.- Test #1 (30,'b',2)



## 8. Conclusiones

- El programa pasó **17/22** pruebas.
- Las **5 fallas** son a pruebas del primer grupo "**Data type Tests**", las entradas del programa están definidas a ser enteros, por lo que al recibir algo que no es entero ocasiona una falla.
- El programa pasa perfectamente las pruebas de rango, valida correctamente si los valores exceden o son inferiores al rango preestablecido.
- El programa pasa perfectamente las pruebas de resultado, da resultados correctos si la figura es un triángulo equilátero, escaleno o isósceles.
- El programa pasa perfectamente las pruebas de resultado, muestra el mensaje correcto en caso de que los valores no formen un triángulo.
- **JUnit** es un herramienta muy útil para las pruebas de software.