

Pruebas y Despliegue de Software

Práctica-1

Triángulo

Hernán Iglesias Ramos

Índice

INTRODUCCIÓN 2

DISEÑO DE PRUEBAS 2

SITUACIONES DE PRUEBA..... 2

CASOS DE PRUEBA 5

REPORTE DE FALLOS 8

RESUMEN EJECUTIVO 11


ANEXO 11

INTRODUCCIÓN

En el contexto de las pruebas, se busca observar la ejecución de una aplicación o sistema para emitir un veredicto sobre la misma, para encontrar posibles fallos y obtener información para evaluar la calidad del producto, para ello se usan las siguientes técnicas: Especificación y Diseño, de la que se obtienen casos de prueba, que son un conjunto de entradas, condiciones de ejecución y resultados esperados para un objetivo particular, la Ejecución del software bajo prueba, y por último la Evaluación y Reporting, donde determinamos los fallos.

El propósito de la presente práctica reside en la identificación y subsanación de posibles deficiencias en la aplicación, mediante la creación de casos de prueba que aborden diversos escenarios. Nos enfocaremos específicamente en la elaboración de casos de prueba destinados a una aplicación diseñada para determinar el tipo de triángulo, a partir de la introducción de tres campos que representan las longitudes numéricas de sus lados, clasificándolos en equiláteros, isósceles, escalenos o inválidos.

Introduzca los valores de los lados del triángulo o 'Cancelar' para salir. ×

	UO: <input type="text" value="UO278462"/>	Primer lado: <input type="text"/>	Segundo lado: <input type="text"/>	Tercer lado: <input type="text"/>
<div><input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/></div>				

1. Interfaz de la aplicación

DISEÑO DE PRUEBAS

SITUACIONES DE PRUEBA

A continuación, enumeramos diversas situaciones de prueba que consideramos pertinentes para evaluar posibles discrepancias entre el comportamiento esperado del software y el observado:

A) Topología:

1. Equilátero (*todos sus lados son iguales*)
 1. Cumpliendo la desigualdad triangular
 2. Sin cumplir la desigualdad triangular (I)
2. Escaleno (*todos sus lados son distintos*)
 1. Cumpliendo la desigualdad triangular
 2. Sin cumplir la desigualdad triangular (I)
3. Isósceles (*dos de sus lados tiene el mismo valor*)
 1. Lados 1 y 2 cumpliendo la desigualdad triangular
 2. Lados 1 y 3 cumpliendo la desigualdad triangular
 3. Lados 2 y 3 cumpliendo la desigualdad triangular
 4. Lados 1 y 2 sin cumplir la desigualdad triangular (I)
 5. Lados 1 y 3 sin cumplir la desigualdad triangular (I)

6. Lados 2 y 3 sin cumplir la desigualdad triangular (I)
4. Inválido
 1. Inserción de valores igual a 0
 1. Lado 1 igual a 0 (I)
 2. Lado 2 igual a 0 (I)
 3. Lado 3 igual a 0 (I)
 4. Todos los lados iguales a 0 (I)
 2. Cadenas vacías (I)
- B) Inserción de caracteres:
 5. Alfabéticos: (*letras, cadenas*)
 1. Lado 1 es una letra/cadena (I)
 2. Lado 2 es una letra/cadena (I)
 3. Lado 3 es una letra/cadena (I)
 4. Todos los lados son letras/cadenas (I)
 6. Numéricos:
 1. Negativos:
 1. Lado 1 es negativo (I)
 2. Lado 2 es negativo (I)
 3. Lado 3 negativo (I)
 4. Todos los lados son negativos (I)
 2. Decimales:
 1. Separados por punto:
 1. Cumpliendo la desigualdad triangular:
 1. Lado 1
 2. Lado 2
 3. Lado 3
 4. Todos los lados
 2. Sin cumplir la desigualdad triangular:
 1. Lado 1 (I)
 2. Lado 2 (I)
 3. Lado 3 (I)
 4. Todos los lados (I)
 2. Separados por coma:
 1. Lado 1 (I)
 2. Lado 2 (I)
 3. Lado 3 (I)
 4. Todos los lados (I)
 3. Valores grandes
 1. Cumpliendo la desigualdad triangular
 2. Sin cumplir la desigualdad triangular (I)
 4. Valores pequeños
 1. Cumpliendo la desigualdad triangular

2. Sin cumplir la desigualdad triangular (I)

7. Especiales:

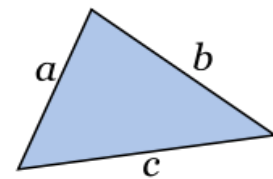
1. Lado 1 tiene un valor especial (I)
2. Lado 2 tiene un valor especial (I)
3. Lado 3 tiene un valor especial (I)
4. Todos los lados tienen un valor especial (I)

C) Desigualdad triangular: (Es un teorema de geometría euclidiana que establece: "En todo triángulo la suma de las longitudes de dos lados cualesquiera es siempre mayor a la longitud del lado restante)

8. Desigualdad:

1. Lado 1 + lado 2 mayor que lado 3
2. Lado 1 + lado 3 mayor que lado 2
3. Lado 2 + lado 3 mayor que lado 1
4. Lado 1 + lado 2 igual a lado 3 (I)
5. Lado 1 + lado 3 igual a lado 2 (I)
6. Lado 2 + lado 3 igual a lado 1 (I)
7. Lado 1 + lado 2 menor que lado 3 (I)
8. Lado 1 + lado 3 menor que lado 2 (I)
9. Lado 2 + lado 3 menor que lado 1 (I)

Desigualdad del triángulo



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

2. Desigualdad triangular

D) Redondeo: (El redondeo es un proceso matemático mediante el cual se ajusta un número a un valor más cercano y manejable, usualmente con menos dígitos significativos o decimales)

9. Redondeo:

1. Cumpliendo la desigualdad triangular
 1. Lado 1 con posible valor a redondear
 2. Lado 2 con posible valor a redondear
 3. Lado 3 con posible valor a redondear
 4. Todos los lados con posible valor a redondear
2. Sin cumplir la desigualdad triangular
 1. Lado 1 con posible valor a redondear (I)
 2. Lado 2 con posible valor a redondear (I)
 3. Lado 3 con posible valor a redondear (I)

E) Truncamiento: (El truncamiento es un proceso matemático que consiste en eliminar los dígitos excedentes de un número, sin redondear)

10. Truncamiento:

1. Cumpliendo la desigualdad triangular
 1. Lado 1 con posible valor a truncar
 2. Lado 2 con posible valor a truncar
 3. Lado 3 con posible valor a truncar
 4. Todos los lados con posible valor a truncar

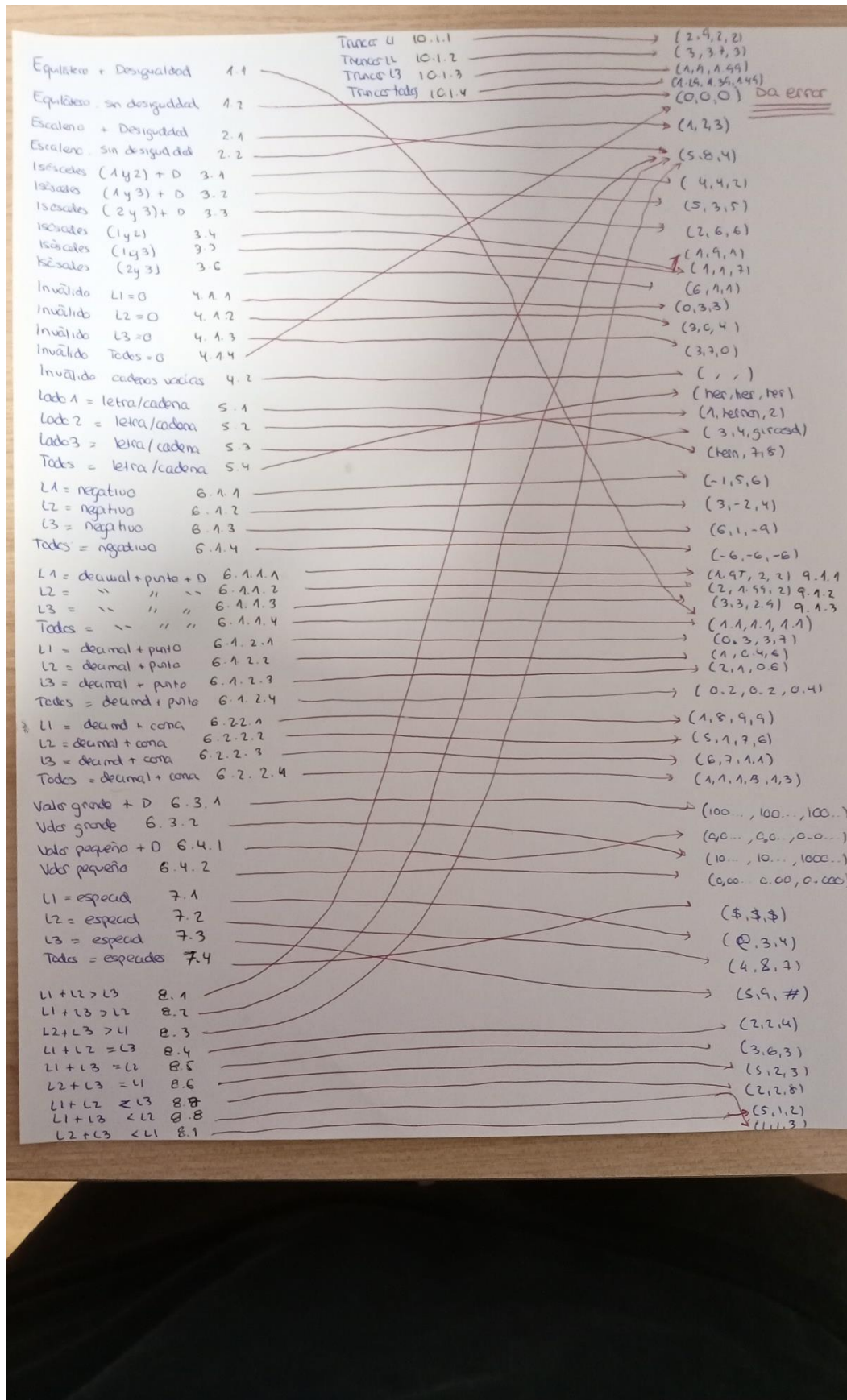
CASOS DE PRUEBA

Aquí presentamos una tabla detallando los casos de prueba para cada situación, incluyendo un identificador único (ID), valores de entrada (x,x,x), trazabilidad hacia las situaciones de prueba que cubre, el resultado esperado y el resultado obtenido:

Id	Valores de entrada	Trazabilidad	Resultado esperado	Resultado obtenido
1	(0,0,0)	1.2	Inválido	Equilátero
2	(0,3,3)	4.1.1	Inválido	Isósceles
3	(3,0,4)	4.1.2	Inválido	Escaleno
4	(3,7,0)	4.1.3	Inválido	Escaleno
5	(1,2,3)	2.2	Inválido	Inválido
6	(5,8,4)	2.1,8.1,8.2,8.3	Escaleno	Escaleno
7	(6,1,1)	3.6	Inválido	Inválido
8	(1,1,7)	3.4	Inválido	Inválido
9	(1,9,1)	3.5	Inválido	Inválido
10	(4,4,2)	3.1	Isósceles	Isósceles
11	(5,3,5)	3.2	Isósceles	Isósceles
12	(2,6,6)	3.3	Isósceles	Isósceles
13	(2,2,4)	8.4	Inválido	Inválido
14	(3,6,3)	8.5	Inválido	Inválido
15	(5,2,3)	8.6	Inválido	Escaleno
16	(2,2,8)	8.7	Inválido	Inválido
17	(1,3,1)	8.8	Inválido	Inválido
18	(5,1,2)	8.9	Inválido	Inválido
19	(her,her,her)	5.4	Inválido	Inválido
20	(1,hernan,2)	5.2	Inválido	Inválido
21	(3,4,girasol)	5.3	Inválido	Inválido
22	(hern,7,8)	5.1	Inválido	Inválido
23	(-1,5,6)	6.1.1	Inválido	Inválido
24	(3,-2,4)	6.1.2	Inválido	Inválido
25	(6,1,-9)	6.1.3	Inválido	Inválido
26	(-6,-6,-6)	6.1.4	Inválido	Inválido
27	(\$,\$,\$)	7.4	Inválido	Inválido
28	(@,3,4)	7.1	Inválido	Inválido
29	(4,&,7)	7.2	Inválido	Inválido
30	(5,9,#)	7.3	Inválido	Inválido
31	(, ,)	4.2	Inválido	Inválido

32	(1,1,1,3,1,3)	6.2.2.4	Inválido	Equilátero
33	(1,8,9,9)	6.2.2.1	Inválido	Isósceles
34	(5,1,7,6)	6.2.2.2	Inválido	Escaleno
35	(6,7,1,1)	6.2.2.3	Inválido	Escaleno
36	(1.1,1.1,1.1)	6.1.1.4, 1.1	Equilátero	Equilátero
37	(0.3,3,7)	6.1.1.2.1	Inválido	Inválido
38	(1,0.4,6)	6.1.1.2.2	Inválido	Inválido
39	(2,1,0.6)	6.1.1.2.3	Inválido	Inválido
40	(0.2,0.2,0.4)	6.2.1.2.4	Inválido	Inválido
41	(1.95,2,2)	9.1.1, 6.1.1.1	Isósceles	Isósceles
42	(2,1.99,2)	9.1.2, 6.1.1.2	Isósceles	Isósceles
43	(3,3,2.9)	9.1.3, 6.1.1.3	Isósceles	Isósceles
44	(2.95,1,1)	9.2.1	Inválido	Inválido
45	(2,3.95,1)	9.2.2	Inválido	Inválido
46	(1,2,3.96)	9.2.3	Inválido	Inválido
47	(2.95,2.96,2.97)	9.1.4	Escaleno	Equilátero
48	(2.9,2,2)	10.1.1	Isósceles	Equilátero
49	(3,3.7,3)	10.1.2	Isósceles	Equilátero
50	(1,1,1.99)	10.1.3	Isósceles	Equilátero
51	(1.29,1.39,1.49)	10.1.4	Escaleno	Equilátero
52	(10000000000,10000000000,10000000000)	6.3.1	Equilátero	Equilátero
53	(0.0000000001,0.0000000001,0.0000000001)	6.4.1	Equilátero	Equilátero
54	(10000000000,1000000000,10000000000)	6.3.2	Inválido	Inválido
55	(0.0000000001, 0.00000001, 0.0000000000000001)	6.4.2	Inválido	Inválido
56	(3,4,5)	2.1	Escaleno	Rectángulo

Quiero añadir, que el caso de prueba (0,0,0) es inválido, para dos situaciones tanto para el equilátero que no cumple la desigualdad triangular, como para introducir todos los campos 0, no se puede probar el mismo así que este ultimo caso no lo hice, pero comprobé el 0 en cada uno de los campos.



3. Casos de prueba (algunos)

REPORTE DE FALLOS

Procedemos a registrar y documentar los problemas/fallos encontrados durante el proceso de prueba del software:

Id-fallo	Descripción	Prioridad	Trazabilidad
1	Meter lados con valor igual a 0	1	CP1, CP2, CP3, CP4
2	Introducir valores decimales separados por coma	1	CP33, CP34, CP35, CP36
3	Al introducir valores decimales los trunca	3	CP48, CP49, CP50, CP51, CP52
4	Al comprobar un valor aparece un resultado no recogido	3	CP57
5	Al comprobar un triángulo invalido devuelve escaleno	2	CP16

(Prioridad en sentido creciente, siendo 1 el menos prioritario y el 3 el más prioritario)

Vamos a comentar cada uno de los fallos de forma más detallada:

Fallo 1

Cuando ingresamos valores nulos (es decir, iguales a 0) en uno o todos los lados del triángulo, se observa que el resultado esperado, que debería ser "Inválido", no se obtiene. El equipo de desarrollo de la aplicación no ha implementado una validación para estos valores, los cuales deberían ser rechazados al introducirlos en los campos correspondientes. Es importante destacar que un triángulo no puede tener lados con valores nulos, ya que esto viola las reglas geométricas básicas. Aunque se espera que los usuarios posean un conocimiento básico sobre la creación de triángulos y comprendan que los valores nulos o negativos son inválidos, la falta de una validación puede generar resultados inesperados y erróneos.

Pasos para reproducir el fallo-1:

1. Iniciar la aplicación.
2. Ingresar un valor nulo (0) en uno o todos los campos correspondientes a los lados del triángulo.
3. Realizar el cálculo para determinar el tipo de triángulo.

Resultado esperado:

Se espera que la aplicación valide los valores ingresados y muestre un mensaje de error indicando que los valores nulos no son válidos para definir un triángulo, o bien al intentar introducir un el número nulo, en este caso es 0, no se nos permita.

Fallo 2

Ingresar valores decimales utilizando la coma como separador, genera resultados incorrectos en la aplicación. Aunque en diferentes regiones del

mundo se utiliza tanto la coma como el punto como separadores decimales, para el desarrollo de la aplicación se ha acordado que el punto será el separador permitido. Sin embargo, la aplicación no ha validado adecuadamente los números decimales separados por coma. Como consecuencia, estos valores se interpretan como números decimales válidos cuando en realidad no lo son. Este problema puede llevar a cálculos incorrectos y resultados imprecisos.

Pasos para reproducir el fallo-2:

1. Abrir la aplicación.
2. Ingresar un valor decimal utilizando la coma como separador, por ejemplo, "3,5".
3. Realizar el cálculo para determinar el tipo de triángulo.

Resultado esperado:

La aplicación debe validar correctamente los valores decimales ingresados, utilizando únicamente el punto como separador decimal. Debería mostrar un mensaje de error indicando que el valor ingresado no es válido debido al uso incorrecto de la coma como separador.

Fallo 3

Al introducir un valor con parte decimal en la aplicación, se observa que la parte decimal se trunca, es decir, se elimina, quedando solo la parte entera. Este comportamiento genera resultados incorrectos en los cálculos realizados por la aplicación. El truncamiento de la parte decimal es una operación inesperada y no deseada, ya que modifica la precisión de los valores introducidos y puede llevarnos a interpretaciones erróneas de los datos.

Pasos para reproducir el fallo-3:

1. Abrir la aplicación.
2. Ingresar un valor decimal con parte decimal, por ejemplo, "3.75".
3. Realizar el cálculo para determinar el tipo de triángulo.

Resultado esperado:

La aplicación debería mantener la precisión de los valores decimales introducidos y realizar cálculos precisos teniendo en cuenta la parte decimal. No se debe truncar la parte decimal de los valores decimales ingresados.

Fallo 4

Al introducir ciertos valores para comprobar que la aplicación nos devuelve un triángulo correcto, nos encontramos con una salida no esperada, ya que devuelve un tipo de triángulo, pero que no habíamos recogido en la especificación.

Nos sale triángulo rectángulo ya que los valores introducidos cumplen el teorema de Pitágoras:

Caso de prueba	$a = \sqrt{c^2 - b^2}$	$b = \sqrt{c^2 - a^2}$	$c = \sqrt{a^2 + b^2}$
CP57	$3 = \sqrt{5^2 - 4^2}$	$4 = \sqrt{5^2 - 3^2}$	$5 = \sqrt{3^2 + 4^2}$

Pasos para reproducir el fallo-4:

1. Abrir la aplicación.
2. Ingresar los valores (3,4,5).
3. Realizar el cálculo para determinar el tipo de triángulo.
4. Salida obtenida: Rectángulo

Resultado esperado:

La aplicación no debería permitir devolver ese tipo de rectángulo ya que en ningún momento se nos comunicó que debíamos abordar este tipo, por tanto, debería de devolver escaleno en todo caso, a no ser que el cliente que nos pida probar este software reconsidere la opción de permitir más tipos de triángulos.

Fallo 5

Al intentar comprobar un triángulo que debería ser inválido, nos devuelve escaleno, el problema, es que no se comprueba correctamente la desigualdad triangular, en este caso $L_2 + L_3 = L_1$, algo incorrecto.

Pasos para reproducir el fallo-5:

1. Abrir la aplicación.
2. Ingresar los valores (5,2,3).
3. Realizar el cálculo para determinar el tipo de triángulo.

Resultado esperado:

El problema puede solventarse, si se comprobase todas las posibilidades de la desigualdad triangular, así evitamos resultados erróneos como en este caso.

RESUMEN EJECUTIVO

A lo largo de las pruebas realizadas se han detectado varios fallos de diferente gravedad:

El fallo más crítico es el fallo 4, ya que en ningún momento en la especificación del programa se dijo de la existencia del tipo de triángulo rectángulo, algo que sería necesario solventar para evitar que a la hora de introducir ciertos valores que satisfagan el teorema de Pitágoras devuelvan este tipo.

Otro muy importante es el fallo 3, ya que el software trunca los números decimales, algo que acordamos que no debería de pasar, habría que tratar este problema analizando cada uno de los números teniendo en cuenta la totalidad de su parte decimal, sin hacer redondeos, o en este caso, que es el problema en sí, truncamientos.

Los otros tres fallos también hay que tenerlos en cuenta, no considero que sean los más importantes, aunque cualquier fallo que exista se debe subsanar. En este caso introducir valores nulos, me parece que si el usuario tiene conocimientos básicos de matemáticas sabrá cuando se puede formar un triángulo, y con valores nulos no será posible, mientras que introducir decimales donde el separador sea una coma, antes dije que en diferentes regiones se usa el punto o bien la coma, en nuestro caso siempre será el punto, por tanto, no podemos tratarlo de la misma manera, también pasa en diferentes aplicaciones matemáticas donde solo es válido el decimal separado por punto, como por ejemplo Matlab, y por último, la posibilidad de la desigualdad triangular que no se cumple, pues debería solventarse abarcando todo tipo de posibilidad, para evitar un error de este tipo.

ANEXO

Imagen 1. Interface de la aplicación

Imagen 2. Desigualdad triangular

Imagen 3. Casos de prueba, algunos, ya que todos no entraban en la hoja.