



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD: INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA EN SISTEMAS

CARRERA: SOFTWARE

CASOS DE PRUEBA

FECHA

05-10-2022

INTEGRANTES

JUAN ROMAN 6751

CHRISTIAN OBANDO 6711

FABRICIO RODRIGUEZ 6491

Descripción del producto de software

Desarrollo de un software con la capacidad de identificar el tipo de triángulo según los valores de sus lados.

Funcionalidades

| | |
|-----------|--|
| F1 | El sistema deberá permitir el ingreso de valores enteros en los lados del triángulo. |
| F2 | El sistema verificara que el triángulo exista. |
| F3 | El sistema identificara el tipo de triángulo. |

Enlace del código:<https://github.com/CrissCrazz/Polinomio-y-Triangulo/tree/main/Triangle>

Funcionalidad 1 : de Ingreso y validación de datos

| |
|---|
| Reglas : $((a,b,c) \in \mathbb{Z}) \quad \& \quad ((a,b,c) \leq 2^80) \quad \& \quad (a \neq 0)$ |
|---|

Casos de prueba

| Caso de prueba | Entradas | Salida |
|------------------|-----------|--------------------------------|
| Caso de prueba 1 | (a, a, a) | (mensaje “ingreso incorrecto”) |

Captura de prueba unitaria

```
1 triangle.py > type_triangle
2 from tkinter import *
3 from tkinter import messagebox
4
5 #Ejercicio de triangulo
6 #Codigo Fuente
7
8 def type_triangle():
9
10
11
12     if ((a.get()+b.get()) > c.get()) and (a.get()+c.get()) > b.get() and (b.get()+c.get()) > a.get() and (a.get().is_integer() and b.get().is_integer() and c.get().is_integer()):
13
14         messageRta = "Ingreso correcto "
15         if (a.get() ==b.get() and a.get()==c.get()):
16             messageTipo = "Es un Triangulo Equilatero"
17
18         elif (a.get() != b.get() and a.get() != c.get()):
19             messageTipo = "Es un Triangulo Escaleno"
20
21         else:
22             messageTipo = "Es un Triangulo Isocoles"
23
24         messagebox.showinfo(message=messageTipo,title=messageRta)
25
26
27     else:
28
29         messageRta = "No es triangulo, ingreso incorrecto, vuelva a ingresar sus datos"
30         messagebox.showerror(message=messageRta)
31
32
33
34 #interfaz grafica
```

Triangulos

Ingrese el primer lado del triangulo: a

Ingrese el segundo lado del triangulo: a

Ingrese el tercer lado del triangulo: a

Comprobar

```
[craaz28@w Python]$ /bin/python /home/craaz28/Documents/Python/Triangle/triangle.py
Exception in Tkinter callback
Traceback (most recent call last):
  File "/usr/lib/python3.10/tkinter/_init_.py", line 1921, in __call__
    return self.func(*args)
  File "/home/craaz28/Documents/Python/Triangle/triangle.py", line 12, in type_triangle
    if ((a.get()+b.get()) > c.get()) and (a.get()+c.get()) > b.get() and (b.get()+c.get()) > a.get() and (a.get().is_integer() and b.get().is_integer() and c.get().is_integer()):
  File "/usr/lib/python3.10/tkinter/_init_.py", line 593, in get
    return self._tk_getdouble(self._tk_globalgetvar(self._name))
tkinter.TclError: expected floating-point number but got "a"
```

| Caso de prueba | Entradas | Salida |
|------------------|-----------|--------------------------------|
| Caso de prueba 2 | (1, A, 2) | (mensaje “ingreso incorrecto”) |

Captura de prueba unitaria

```
1 triangle.py > test.py
2 from tkinter import *
3 from tkinter import messagebox
4
5 #Ejercicio de triangulo
6 #Codigo Fuente
7
8 def type_triangle():
9
10
11
12     if ((a.get()+b.get()) > c.get()) and (a.get()+c.get()) > b.get() and (b.get()+c.get()) > a.get() and (a.get().is_integer() and b.get().is_integer() and c.get().is_integer()):
13
14         messageRta = "Ingreso correcto "
15         if (a.get() ==b.get() and a.get()==c.get()):
16             messageTipo = "Es un Triangulo Equilatero"
17
18         elif (a.get() != b.get() and a.get() != c.get()):
19             messageTipo = "Es un Triangulo Escaleno"
20
21         else:
22             messageTipo = "Es un Triangulo Isocoles"
23
24         messagebox.showinfo(message=messageTipo,title=messageRta)
25
26
27     else:
28
29         messageRta = "No es triangulo, ingreso incorrecto, vuelva a ingresar sus datos"
30         messagebox.showerror(message=messageRta)
31
32
33
34 #interfaz grafica
```

Triangulos

Ingrese el primer lado del triangulo: 1

Ingrese el segundo lado del triangulo: A

Ingrese el tercer lado del triangulo: 2

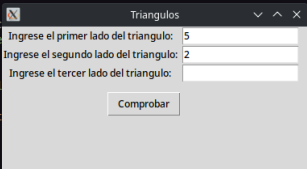
Comprobar

```
[craaz28@w Python]$ /bin/python /home/craaz28/Documents/Python/Triangle/triangle.py
Exception in Tkinter callback
Traceback (most recent call last):
  File "/usr/lib/python3.10/tkinter/_init_.py", line 1921, in __call__
    return self.func(*args)
  File "/home/craaz28/Documents/Python/Triangle/triangle.py", line 12, in type_triangle
    if ((a.get()+b.get()) > c.get()) and (a.get()+c.get()) > b.get() and (b.get()+c.get()) > a.get() and (a.get().is_integer() and b.get().is_integer() and c.get().is_integer()):
  File "/usr/lib/python3.10/tkinter/_init_.py", line 593, in get
    return self._tk_getdouble(self._tk_globalgetvar(self._name))
tkinter.TclError: expected floating-point number but got "A"
```

| Caso de prueba | Entradas | Salida |
|------------------|----------|--------------------------------|
| Caso de prueba 3 | (5,2,) | (mensaje “ingreso incorrecto”) |

Captura de prueba unitaria

```
9 def type_triangle():
10
11
12 if ((a.get()+b.get()) > c.get() and (a.get()+c.get()) > b.get() and (b.get()+c.get()) > a.get()) and (a.get().is_integer() and b.get().is_integer() and c.get().is_integer()):
13
14     messageRta = "Ingreso correcto "
15     if (a.get() == b.get() and a.get() == c.get()):
16         messageTipo = "Es un Triangulo Equilatero"
17
18     elif (a.get() != b.get() and a.get() != c.get()):
19         messageTipo = "Es un Triangulo Escalo"
20
21     else:
22         messageTipo = "Es un Triangulo Isoce"
23
24     messagebox.showinfo(message=messageTipo, title="Triangulos")
25
26
27 else:
28
29     messageRta = "No es triangulo, ingreso incorrecto, vuelva a ingresar sus datos"
30     messagebox.showerror(message=messageRta)
31
32
33
34 #interfaz grafica
```

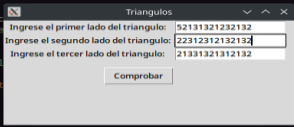


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
_tkinter.TclError: expected floating-point number but got ""
Exception in Tkinter callback
Traceback (most recent call last):
  File "/usr/lib/python3.10/tkinter/_init_.py", line 1921, in __call__
    return self.func(*args)
  File "/home/craaz/Documents/Python/Triangle/triangle.py", line 12, in type_triangle
    if ((a.get()+b.get()) > c.get() and (a.get()+c.get()) > b.get() and (b.get()+c.get()) > a.get()) and (a.get().is_integer() and b.get().is_integer() and c.get().is_integer()) :
  File "/usr/lib/python3.10/tkinter/_init_.py", line 593, in get
    return self._tk.getdouble(self._tk.globalgetvar(self._name))
_tkinter.TclError: expected floating-point number but got ""
```

| Caso de prueba | Entradas | Salida |
|------------------|---|---------------------------------------|
| Caso de prueba 4 | (52113131313131313, 52113131313131313, 52113131313131313) | (mensaje “valor ingresado no valido”) |

Captura de prueba unitaria

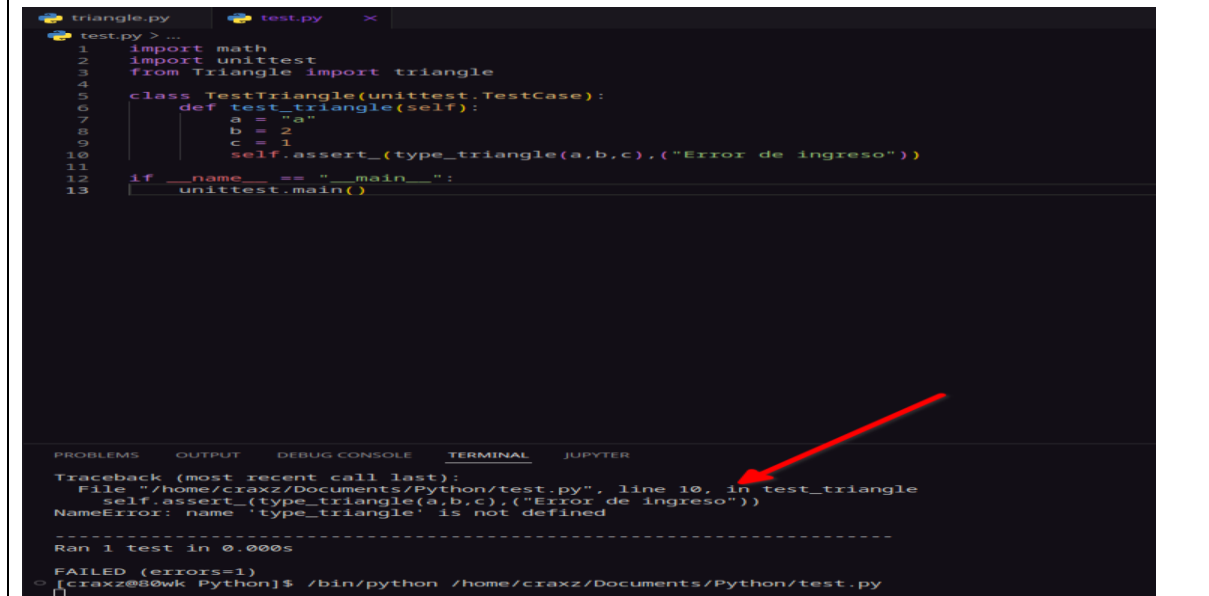
```
1 from tkinter import *
2 from tkinter import messagebox
3
4 #ejercicio de triangulo
5 #codigo fuente
6
7
8 def type_triangle():
9
10
11
12 if ((a.get()+b.get()) > c.get() and (a.get()+c.get()) > b.get() and (b.get()+c.get()) > a.get()) and (a.get().is_integer() and b.get().is_integer() and c.get().is_integer()):
13
14     messageRta = "Ingreso correcto "
15     if (a.get() == b.get() and a.get() == c.get()):
16         messageTipo = "Es un Triangulo Equilatero"
17
18     elif (a.get() != b.get() and a.get() != c.get()):
19         messageTipo = "Es un Triangulo Escalo"
20
21     else:
22         messageTipo = "Es un Triangulo Isoce"
23
24     messagebox.showinfo(message=messageTipo, title="Triangulos")
25
26
27 else:
28
29     messageRta = "No es triangulo, ingreso incorrecto, vuelva a ingresar sus datos"
30     messagebox.showerror(message=messageRta)
31
32
33
34 #interfaz grafica
```



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
_tkinter.TclError: expected floating-point number but got ""
Exception in Tkinter callback
Traceback (most recent call last):
  File "/usr/lib/python3.10/tkinter/_init_.py", line 1921, in __call__
    return self.func(*args)
  File "/home/craaz/Documents/Python/Triangle/triangle.py", line 12, in type_triangle
    if ((a.get()+b.get()) > c.get() and (a.get()+c.get()) > b.get() and (b.get()+c.get()) > a.get()) and (a.get().is_integer() and b.get().is_integer() and c.get().is_integer()) :
  File "/usr/lib/python3.10/tkinter/_init_.py", line 593, in get
    return self._tk.getdouble(self._tk.globalgetvar(self._name))
_tkinter.TclError: expected floating-point number but got ""
```

| Caso de prueba | Entradas | Salida |
|------------------|-----------|--------------------------------|
| Caso de prueba 5 | (a, 2, 1) | (mensaje “ingreso incorrecto”) |

Captura de prueba unitaria



```

triangle.py  test.py  X
test.py > ...
1  import math
2  import unittest
3  from Triangle import triangle
4
5  class TestTriangle(unittest.TestCase):
6      def test_triangle(self):
7          a = "a"
8          b = 2
9          c = 1
10         self.assert_(type_triangle(a,b,c),("Error de ingreso"))
11
12     if __name__ == "__main__":
13         unittest.main()

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER

Traceback (most recent call last):
  File "/home/craxz/Documents/Python/test.py", line 10, in test_triangle
    self.assert_(type_triangle(a,b,c),("Error de ingreso"))
NameError: name 'type_triangle' is not defined

-----
Ran 1 test in 0.000s

FAILED (errors=1)
[craxz@80wk Python]$ /bin/python /home/craxz/Documents/Python/test.py

```

| Caso de prueba | Entradas | Salida |
|------------------|-----------|--------------------------------|
| Caso de prueba 6 | (6, a, 1) | (mensaje “ingreso incorrecto”) |

Captura de prueba unitaria



```

triangle.py  test.py  X
test.py > TestTriangle > test_triangle
1  import math
2  import unittest
3  from Triangle import triangle
4
5  class TestTriangle(unittest.TestCase):
6      def test_triangle(self):
7          a = 0
8          b = "a"
9          c = 1
10         self.assert_(type_triangle(a,b,c),("Error de ingreso"))
11
12     if __name__ == "__main__":
13         unittest.main()

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER

Traceback (most recent call last):
  File "/home/craxz/Documents/Python/test.py", line 10, in test_triangle
    self.assert_(type_triangle(a,b,c),("Error de ingreso"))
NameError: name 'type_triangle' is not defined

-----
Ran 1 test in 0.001s

FAILED (errors=1)
[craxz@80wk Python]$ /bin/python /home/craxz/Documents/Python/test.py

```

| Caso de prueba | Entradas | Salida |
|------------------|----------|--------------------------------|
| Caso de prueba 7 | (6, a,) | (mensaje “ingreso incorrecto”) |

Captura de prueba unitaria

```

triangle.py
test.py X
test.py > ...
1 import math
2 import unittest
3 from Triangle import triangle
4
5 class TestTriangle(unittest.TestCase):
6     def test_triangle(self):
7         a = 0
8         b = "a"
9         c = null
10        self.assert_(type_triangle(a,b,c),("Error de ingreso"))
11
12 if __name__ == "__main__":
13     unittest.main()

```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** JUPYTER

```

Traceback (most recent call last):
  File "/home/craxz/Documents/Python/test.py", line 10, in test_triangle
    self.assert_(type_triangle(a,b,c),("Error de ingreso"))
NameError: name 'type_triangle' is not defined
-----
Ran 1 test in 0.001s
FAILED (errors=1)
[craxz@80wk Python]$ /bin/python /home/craxz/Documents/Python/test.py

```

Funcionalidad 2: El sistema verificara que el triángulo exista.

Reglas:

Si $(a \geq a+b)$ OR $(b \geq a+c)$ OR $(b \geq a+c)$ AND $(A,B,C) \neq 0$ existe

| Caso de prueba | Entradas | Salida |
|------------------|-----------|-----------------------------|
| Caso de prueba 8 | (0, 0, 0) | (mensaje “no es triangulo”) |

Captura de prueba unitaria

triangle.py x test.py

Triangle > triangle.py > type_triangle

1 from tkinter import *
2 from tkinter import messagebox
3
4 #Ejercicio de triangulo
5 #Codigo Fuente
6
7
8
9
10 def type_triangle():
11
12 if ((a.get()+b.get()) > c.get() and (a.get()+c.get()) > b.get() and (b.get()+c.get()) > a.get()) and b.get().is_integer() and c.get().is_integer():
13
14 messageRta = "Ingreso correcto "
15 if (a.get() ==b.get() and a.get()==c.get()):
16 messageTipo = "Es un Triangulo Equilatero"
17
18 elif (a.get() != b.get() and a.get() != c.get()):
19 messageTipo = "Es un Triangulo Escaleno"
20
21 else:
22 messageTipo = "Es un Triangulo Isocoles"
23
24 messagebox.showinfo(message=messageTipo,title=messageRta)
25
26
27 else:
28 messageRta = "No es triangulo, ingreso incorrecto, vuelva a ingresar sus datos"
29 messagebox.showerror(message=messageRta)
30
31
32
33
34 #interfaz grafica

Triangulos

Ingrese el primer lado del triangulo: 0.0
Ingrese el segundo lado del triangulo: 0.0
Ingrese el tercer lado del triangulo: 0.0

Comprobar

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

Ran 1 test in 0.001s
OK
[craxz@80wk Python]\$ /bin/python /home/craxz/Documents/Python/test.py

Ran 1 test in 0.001s
OK
[craxz@80wk Python]\$ /bin/python /home/craxz/Documents/Python/Triangle/triangle.py

Funcionalidad 3: El sistema identificara el tipo de triangulo.

Reglas:

Si (a=b Y a=b) **Equilátero**

Si (a=b AND a≠c) OR (b=c AND b≠c) **Isosceles**

Si (a≠b AND a≠c AND b≠c) **Escaleno**

| Caso de prueba | Entradas | Salida |
|---|-----------|-----------------------------|
| Caso de prueba 9 | (0, 0, 0) | (mensaje “no es triangulo”) |
| Captura de prueba unitaria | | |
| <div><pre>messageRta = "Ingreso correcto " if (a.get() ==b.get() and a.get()==c.get()): messageTipo = "Es un Triangulo Equilatero" elif (a.get() ==b.get() and a.get()!=c.get()) or (b.get()==c.get() and b.get()!=a.get()): messageTipo = "Es un Triangulo Isosceles" else: messageTipo = "No es triangulo, ingreso incorrecto, vuelva a ingresar sus datos" messagebox.showinfo(message=messageTipo, title="Triangulos") else: messageRta = "No es triangulo, ingreso incorrecto, vuelva a ingresar sus datos" messagebox.showerror(message=messageRta)</pre><div><div>Ingreso correcto</div><div>Es un Triangulo Escaleno</div><div>OK</div></div><div><div>Triangulos</div><div>Ingrese el primer lado del triangulo: 5</div><div>Ingrese el segundo lado del triangulo: 2</div><div>Ingrese el tercer lado del triangulo: 4</div><div>Comprobar</div></div></div> | | |

Matriz de trazabilidad

| | F1 | F2 | F3 |
|-----|----|----|----|
| Cp1 | X | | |
| Cp2 | X | | |
| Cp3 | X | | |
| Cp4 | X | | |
| Cp5 | X | | |
| Cp6 | X | | |
| Cp7 | X | | |
| Cp8 | | X | |
| Cp9 | | | X |