



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Alejandro Esteban Pimentel Alarcón

Asignatura: Fundamentos de Programación

Grupo: 135

No de Práctica(s): 4

Intearante(s): Torres Alcántara Alan Eliezer

*No. de Equipo de
cómputo empleado:* Suiza 50

No. de Lista o 9032

Semestre: 2020 - 1

Fecha de entrega: 09/09/2019

Observaciones:

CALIFICACIÓN: _____

DIAGRAMAS DE FLUJO

¿Qué es un diagrama de flujo?

Un diagrama de flujo es un diagrama que describe un proceso, sistema o algoritmo informático. Se usan ampliamente en numerosos campos para documentar, estudiar, planificar, mejorar y comunicar procesos que suelen ser complejos en diagramas claros y fáciles de comprender.

Los diagramas de flujo emplean rectángulos, óvalos, diamantes y otras numerosas figuras para definir el tipo de paso, junto con flechas conectoras que establecen el flujo y la secuencia. Pueden variar desde diagramas simples y dibujados a mano hasta diagramas exhaustivos creados por computadora que describen múltiples pasos y rutas.

Diagramas de flujo para algoritmos/programación informática

Como una representación visual del flujo de datos, los diagramas de flujo son útiles para escribir un programa o algoritmo y explicárselo a otros o colaborar con otros en el mismo.

Puedes usar un diagrama de flujo para explicar detalladamente la lógica detrás de un programa antes de empezar a codificar el proceso automatizado. Puede ayudar a organizar una perspectiva general y ofrecer una guía cuando llega el momento de codificar. Los diagramas de flujo pueden:

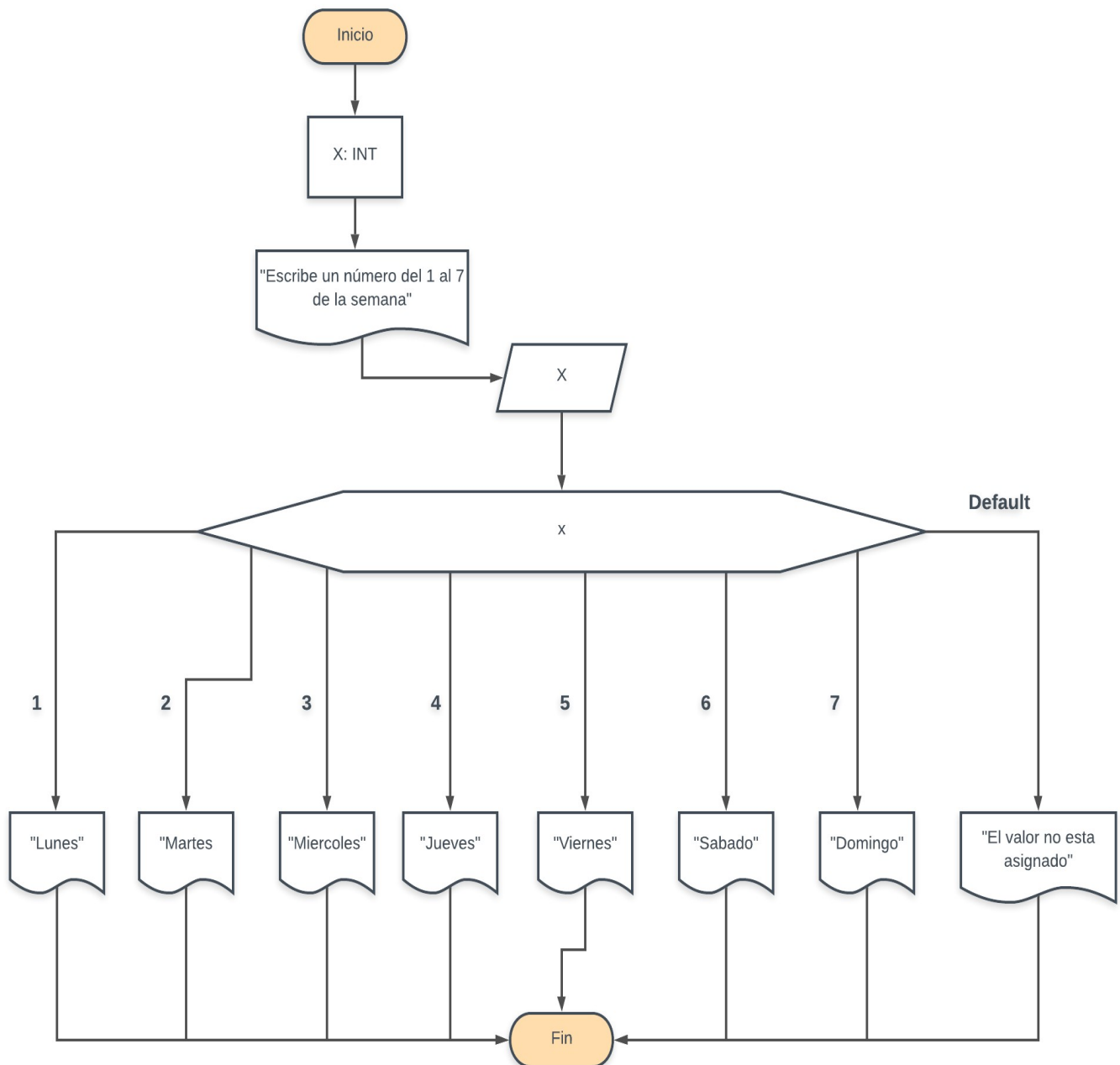
- Demostrar cómo el código está organizado.
- Visualizar la ejecución de un código dentro de un programa.
- Mostrar la estructura de un sitio web o aplicación.
- Comprender cómo los usuarios navegan por un sitio web o programa.

Objetivo

Elaborar diagramas de flujo que representen soluciones algorítmicas vistas como una serie de acciones que comprendan un proceso.

Actividad 1

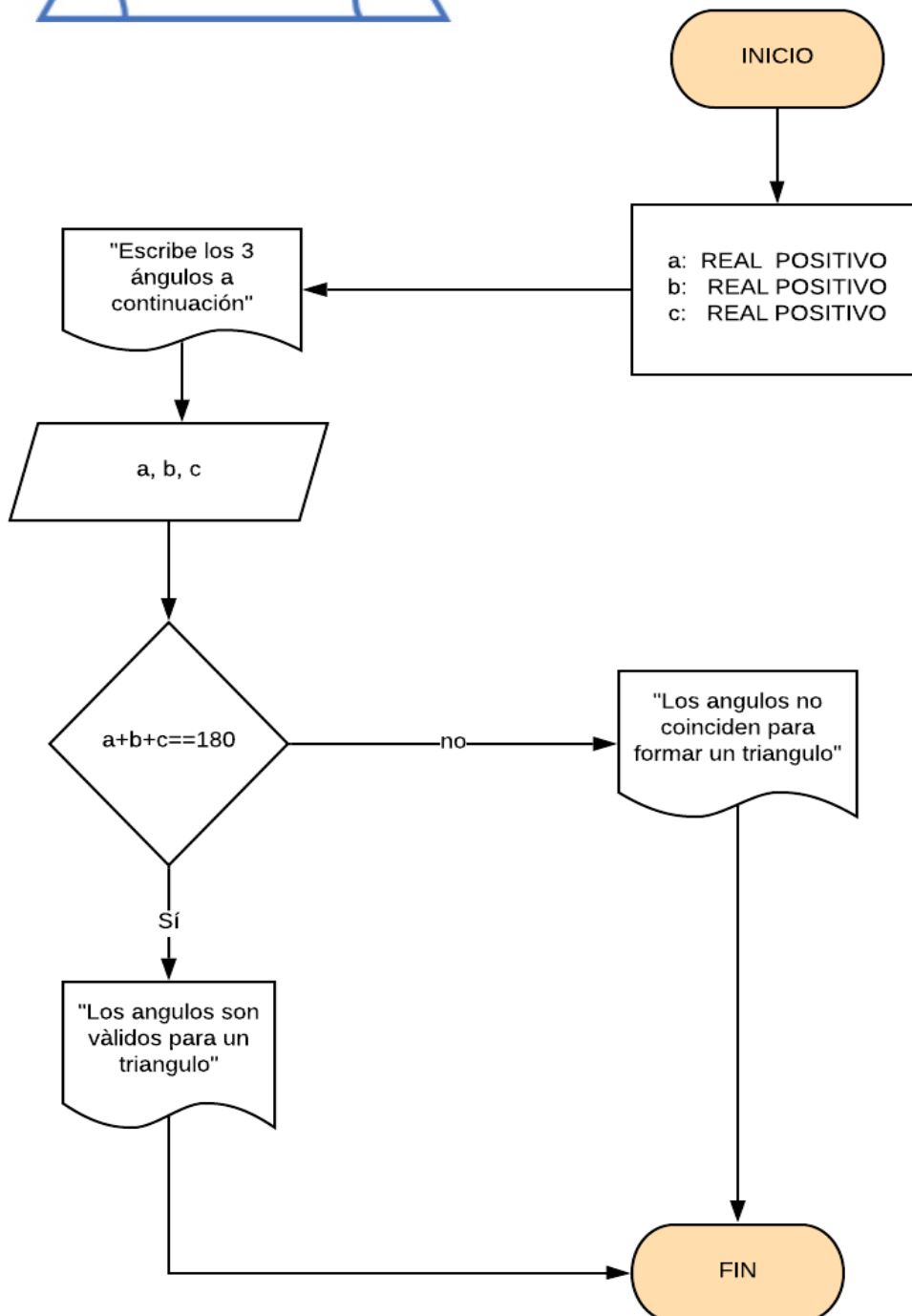
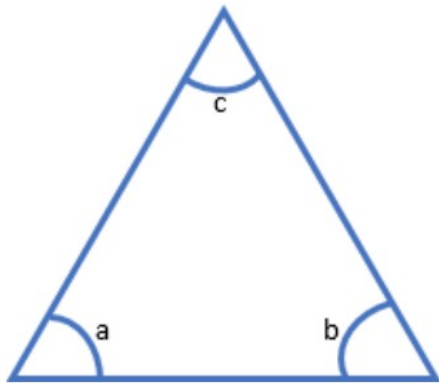
- Diagrama de flujo que reciba un número del 1 al 7, y que indique a qué día de la semana corresponde



En este caso en particular, ocupé SWITCH para representar los valores enteros que van del 1 al 7

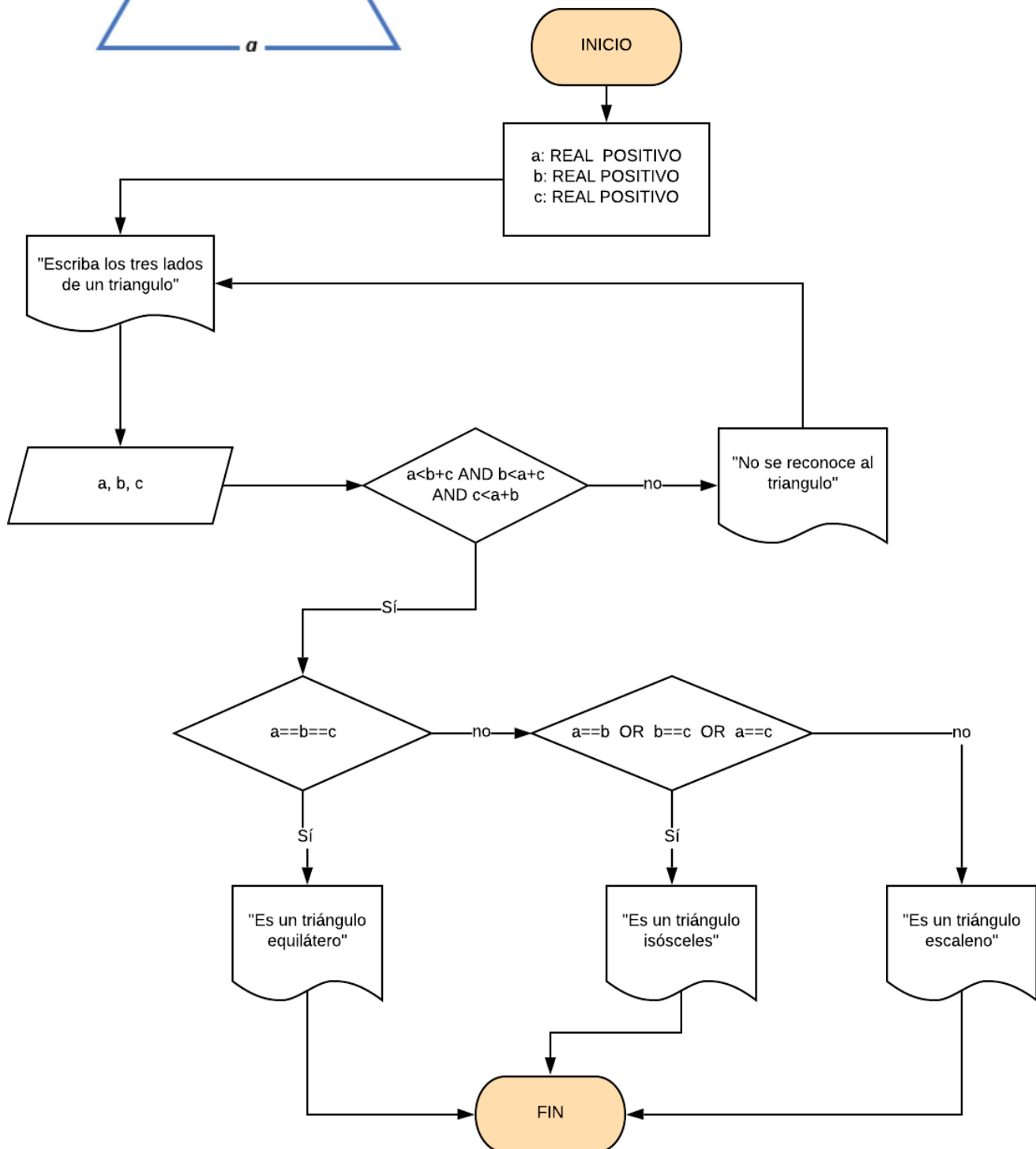
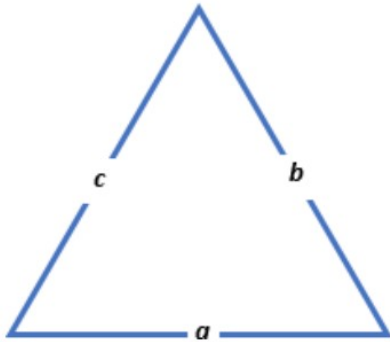
Actividad 2

- Diagrama de flujo que reciba tres números y verifique si son válidos como los ángulos de un triángulo.



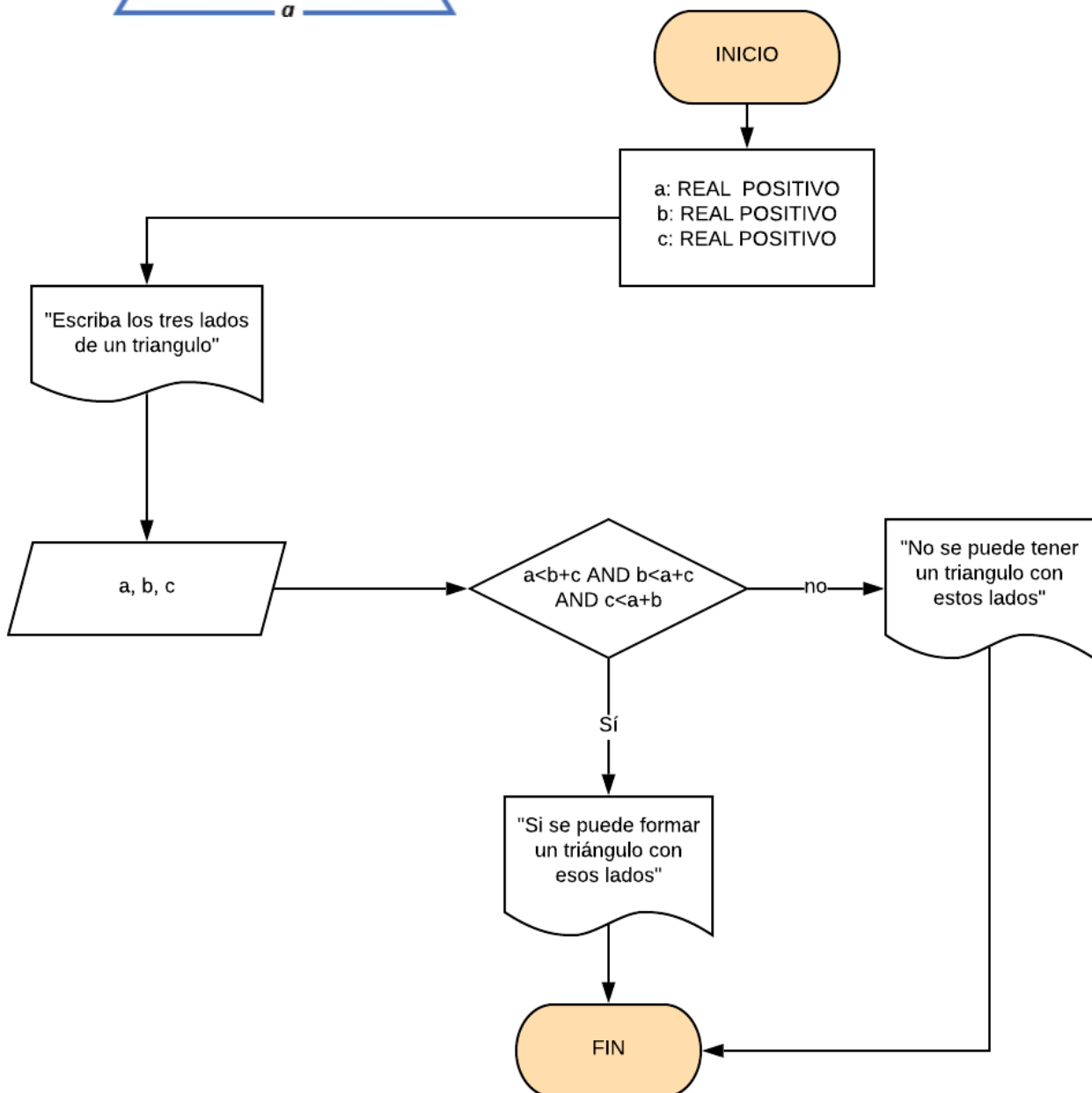
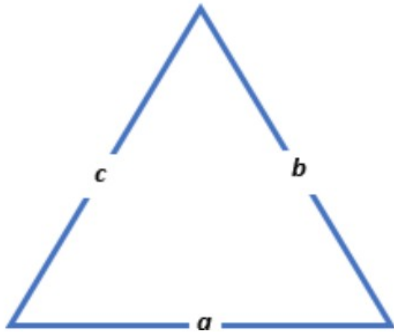
Actividad 3

- Diagrama de flujo que reciba tres números como los lados de un triángulo, y que responda si se trata de un triángulo equilátero, isósceles, o escaleno.



Actividad 4

- Diagrama de flujo que reciba tres números como los lados de un triángulo, y que responda si se puede formar un triángulo con lados de esa longitud, o no.



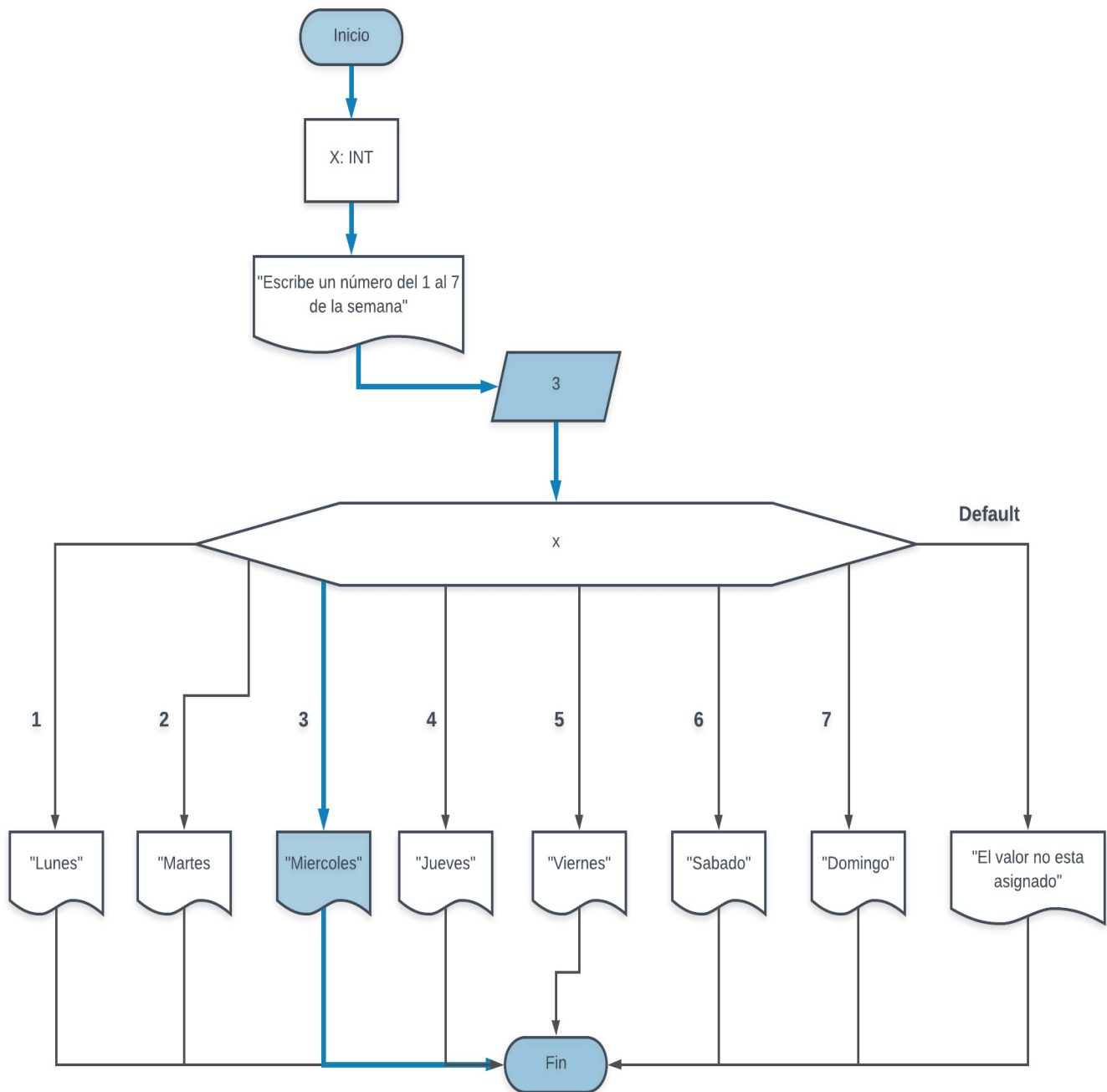
Actividad 5

Verificar las actividades anteriores con los datos:

- Números a días: 3,7,-2,0,9,"Lunes"
- Ángulos de triángulo:
 - 30,30,120
 - -90,90,180
 - 0,30,150
 - 270,60,30
- Tipos de triángulos:
 - 45,50,80
 - 20,20,20
 - 10,100,10
 - 0,4,20
- Triángulo aceptable:
 - 20,40,20
 - 60,100,200
 - -3,6,12
 - 4,5,9

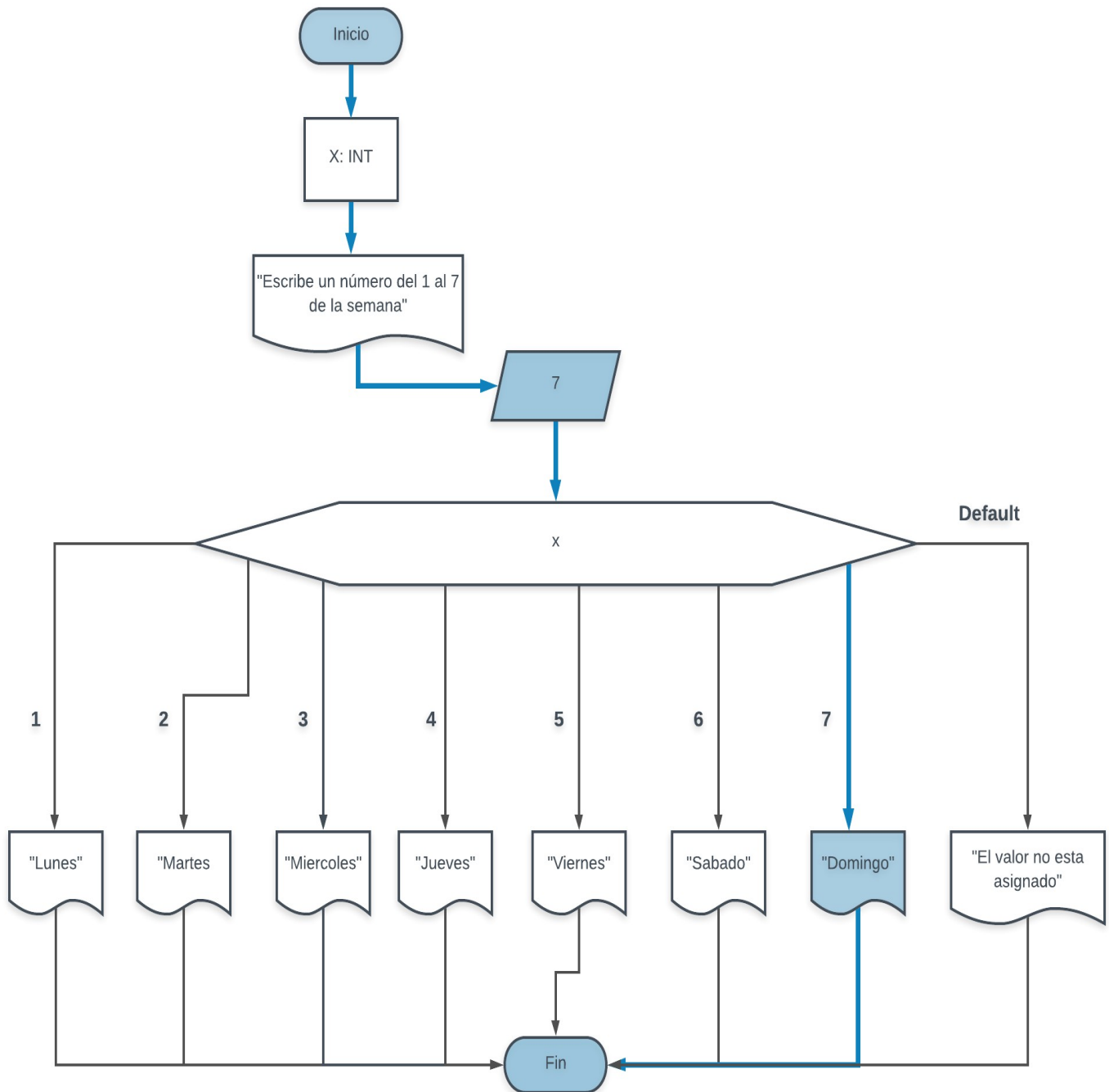
Números a días:

a.



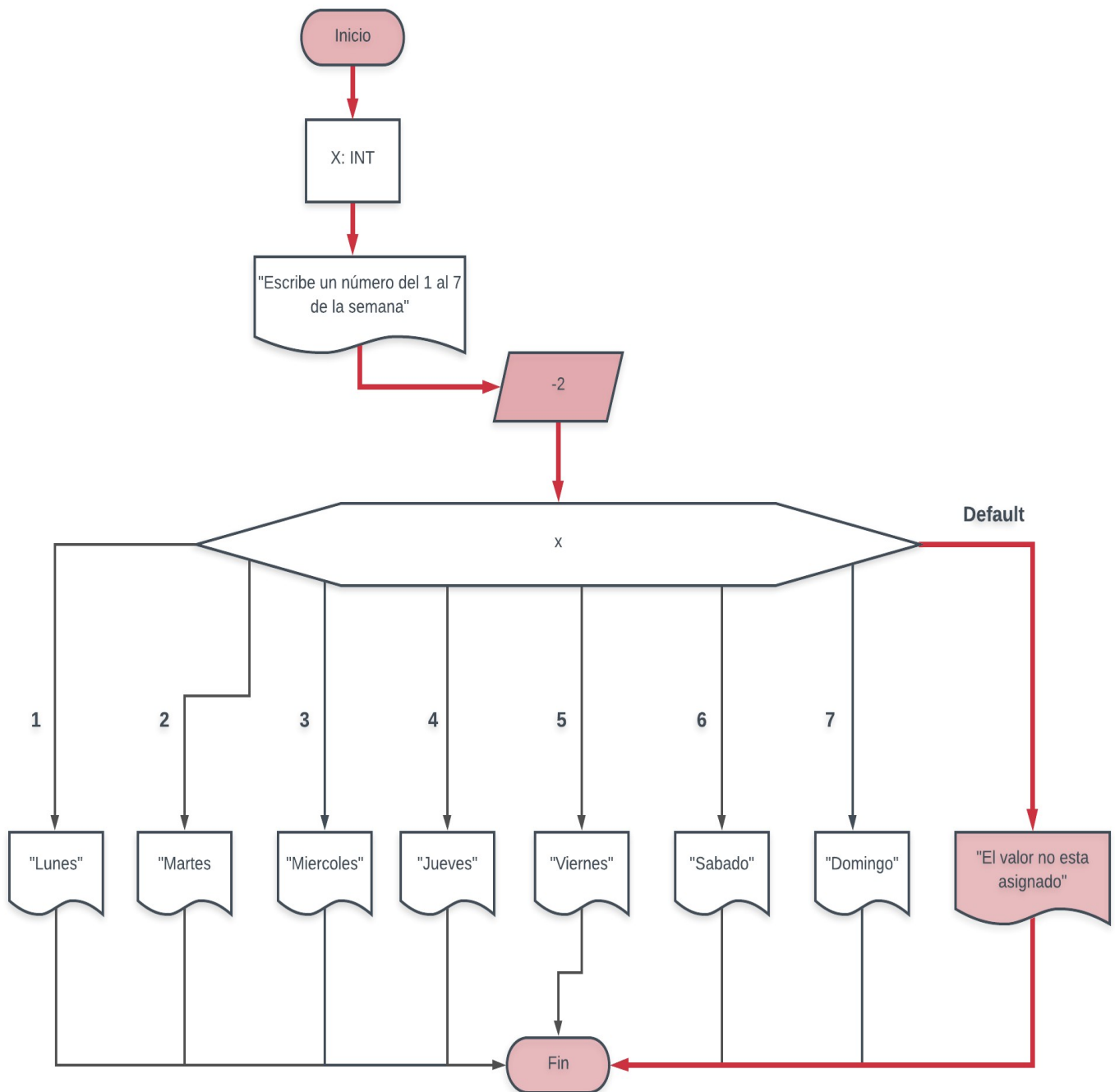
Dentro del sistema se encuentra asignado el valor “3”, por lo cual no hubo ningún problema y se desarrolló el programa correctamente.

b.



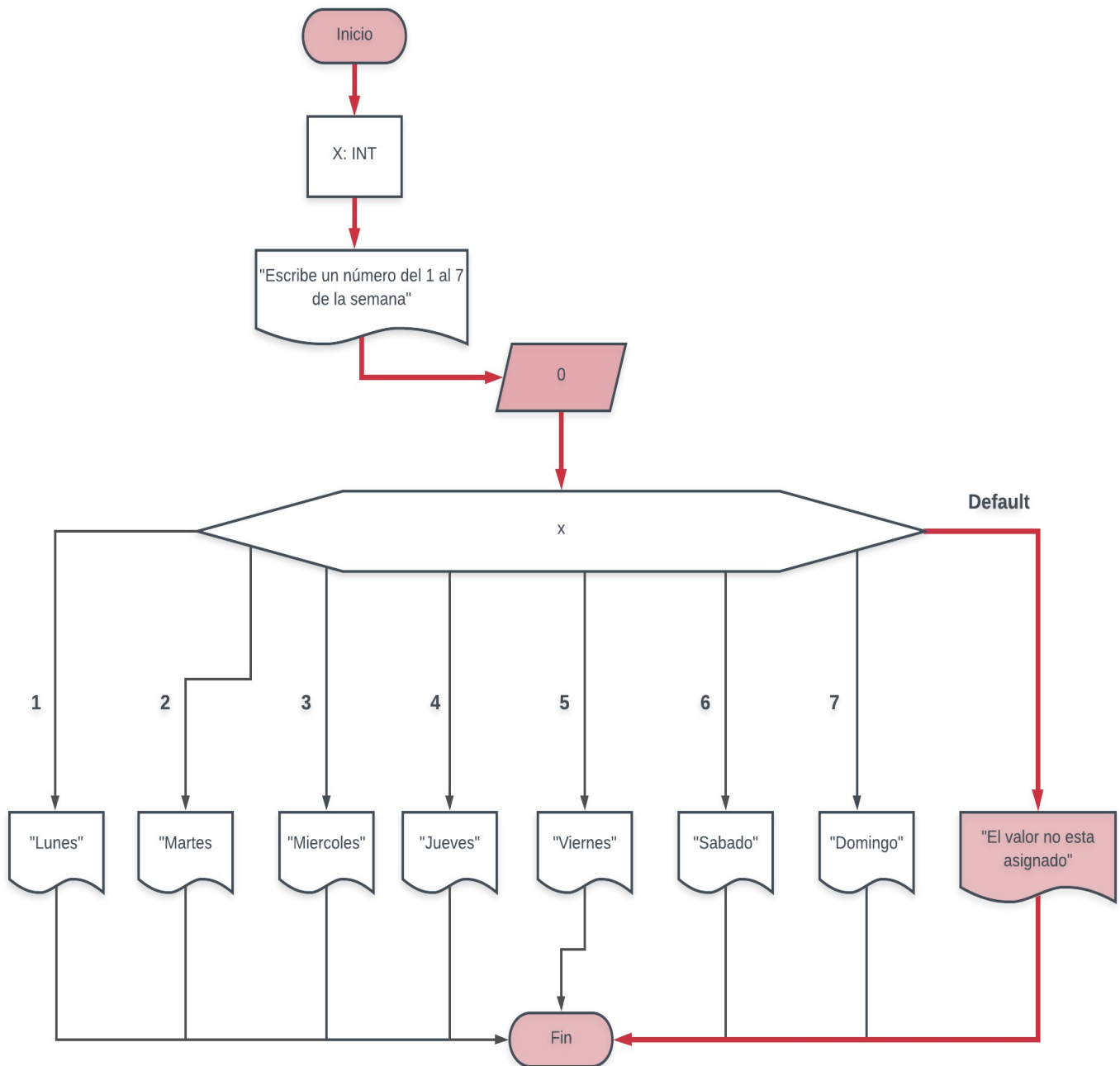
Dentro del sistema se encuentra asignado el valor “7”, por lo cual no hubo ningún problema y se desarrolló el programa correctamente.

C.



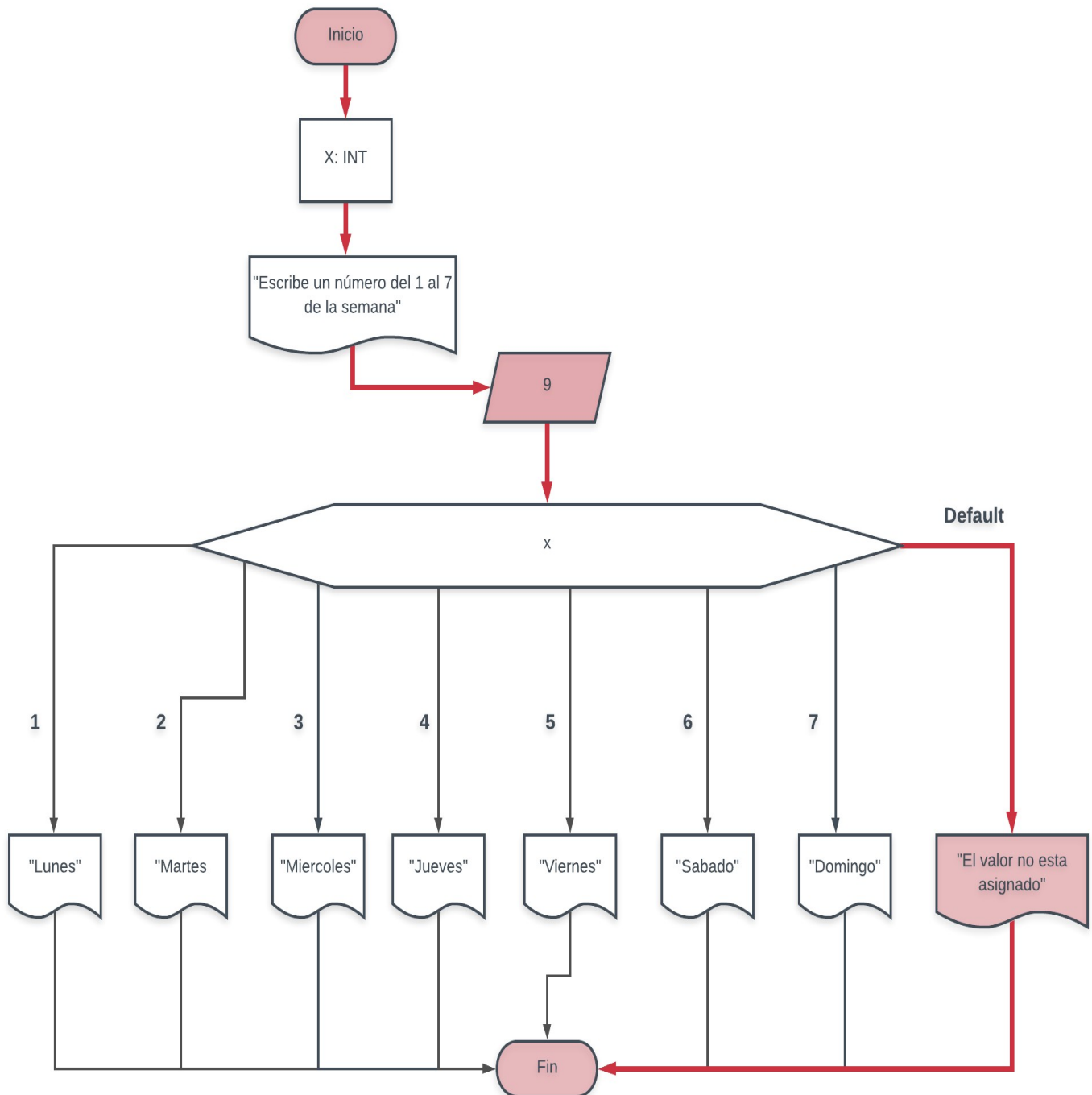
Dentro del sistema no se encuentra asignado el valor “-2”, sin embargo, no hubo ningún problema ya que con SWITCH se puede asignar que cualquier valor que no cumpla las condiciones se manda a DEFAULT y se desarrolla el programa correctamente.

d.



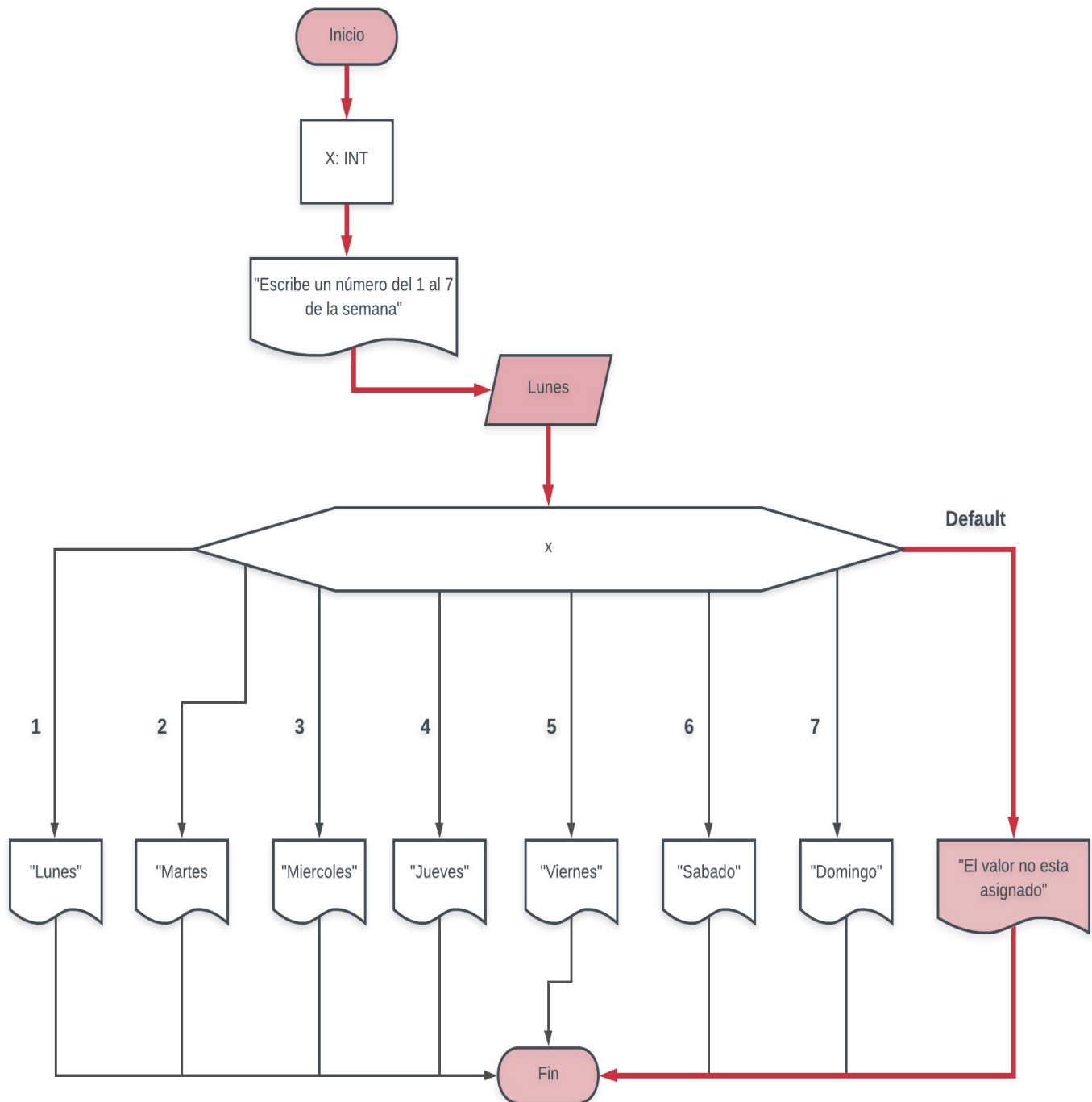
Dentro del sistema no se encuentra asignado el valor “0”, sin embargo, no hubo ningún problema ya que con SWITCH se puede asignar que cualquier valor que no cumpla las condiciones se manda a DEFAULT y se desarrolla el programa correctamente.

e.



Dentro del sistema no se encuentra asignado el valor “9”, sin embargo, no hubo ningún problema ya que con SWITCH se puede asignar que cualquier valor que no cumpla las condiciones se manda a DEFAULT y se desarrolla el programa correctamente.

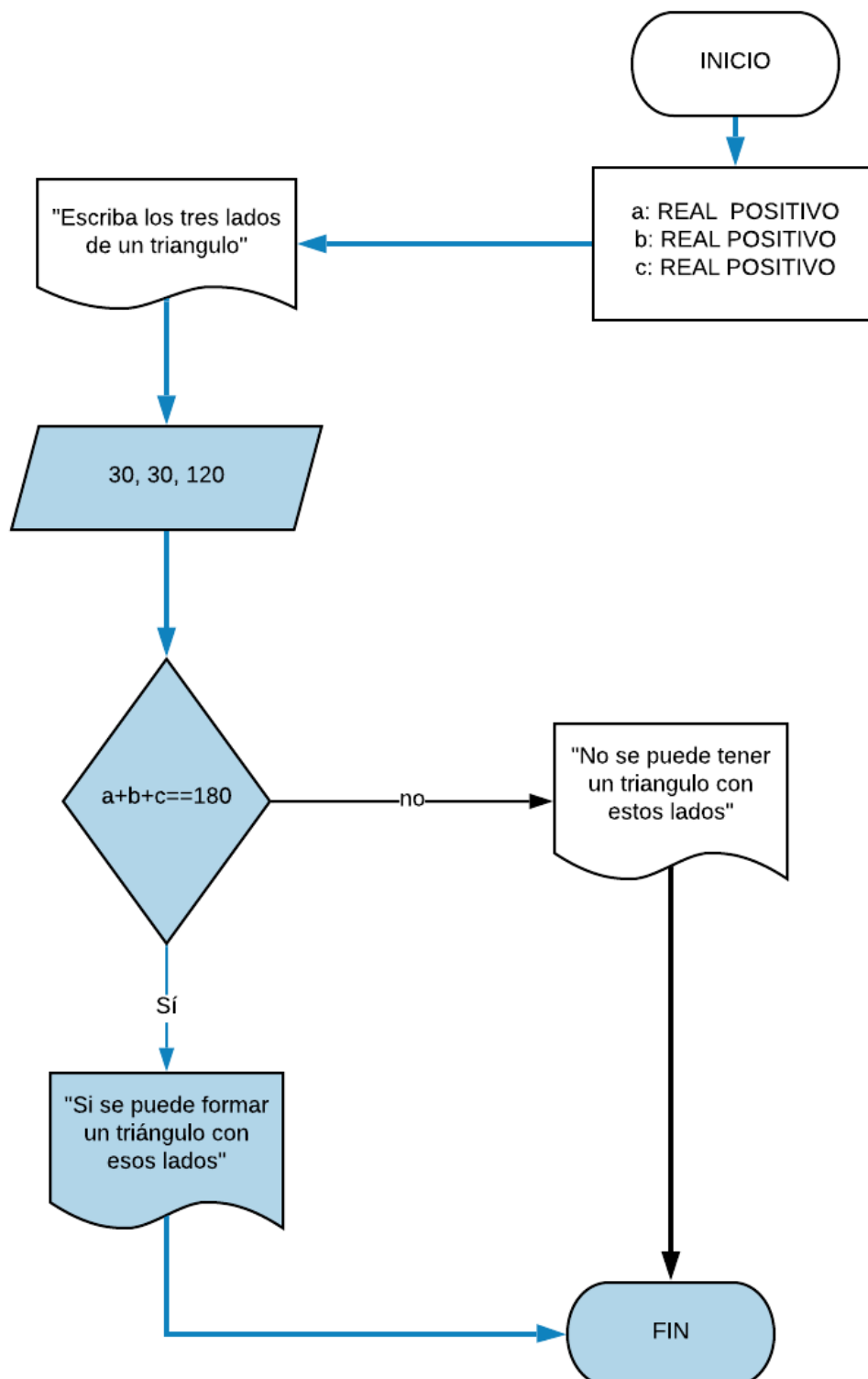
f.



Dentro del sistema no se encuentra asignado el valor “lunes”, sin embargo no hubo ningún problema ya que con SWITCH se puede asignar que cualquier valor que no cumpla las condiciones se manda a DEFAULT y se desarrolla el programa correctamente.

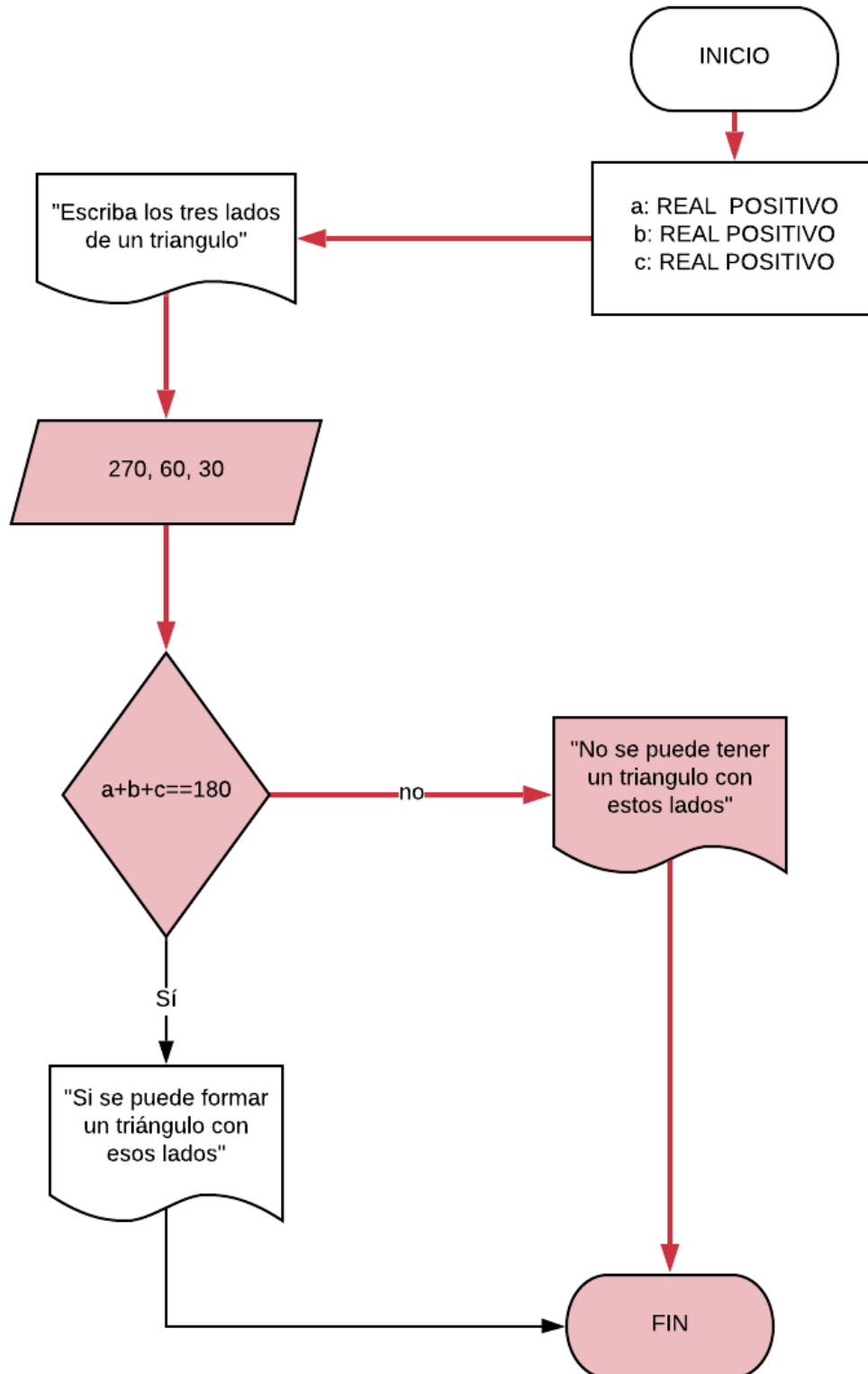
Ángulos de triángulo

a.



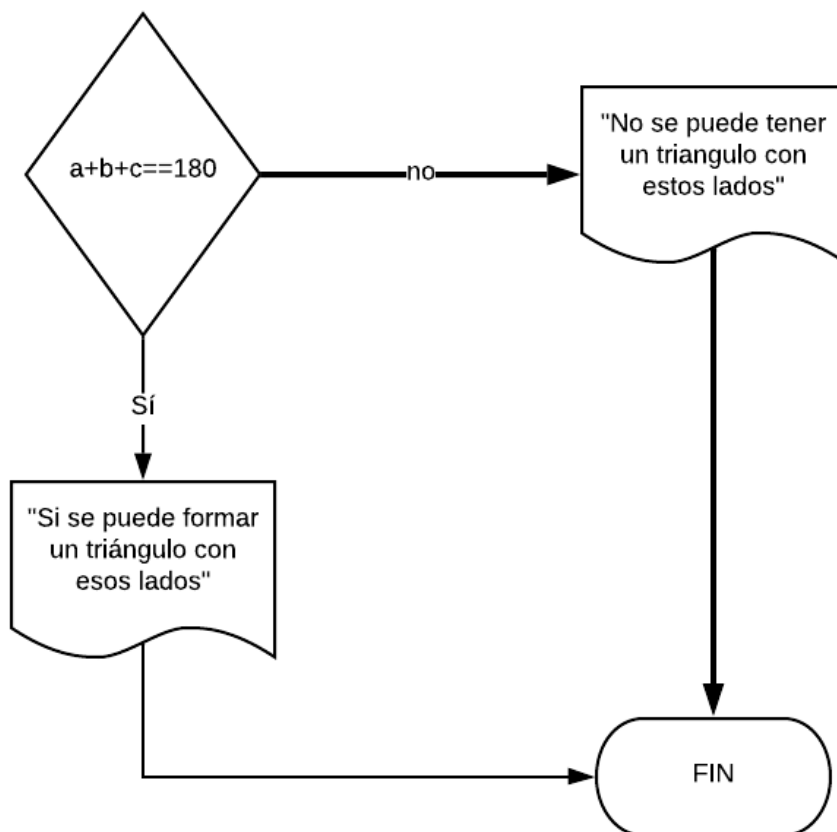
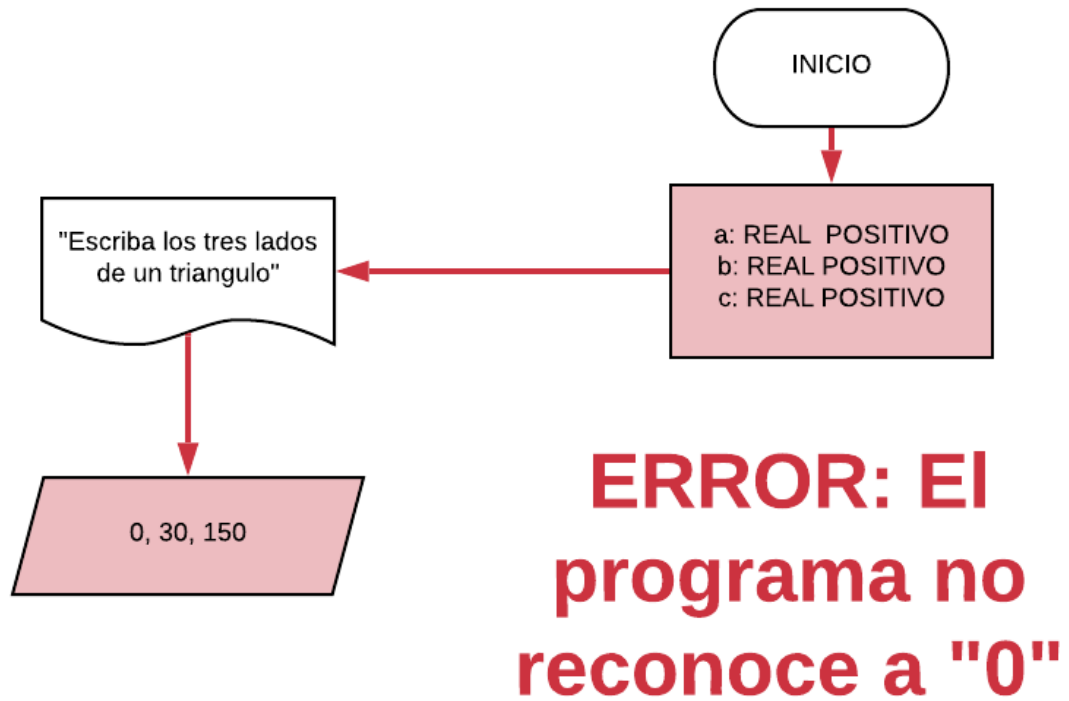
30,30 y 120 cumplen con la condición $a+b+c==180$, por lo que se imprime que si se puede formar un triángulo con esos lados, el programa se desarrolla correctamente.

b.



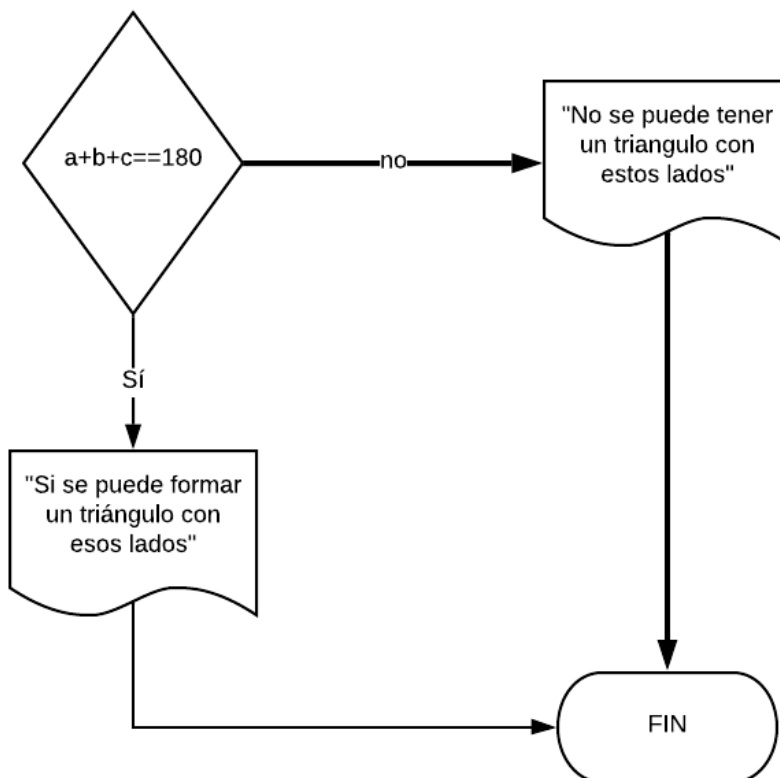
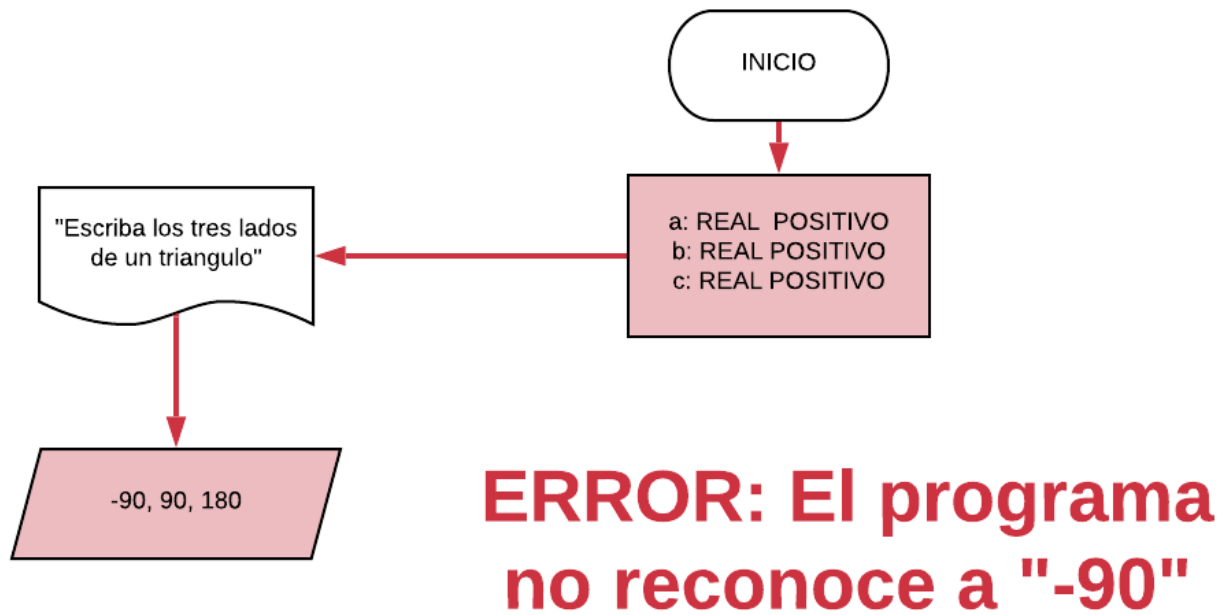
270, 60 y 30 no cumplen con la condición $a+b+c==180$, por lo que se imprime que no se puede formar un triángulo con esos lados, el programa se desarrolla correctamente.

C.



En este caso el programa no reconoce a "0" ya que no está establecido en los datos del proceso, por lo que hay un ERROR y el programa ya no se puede desarrollar.

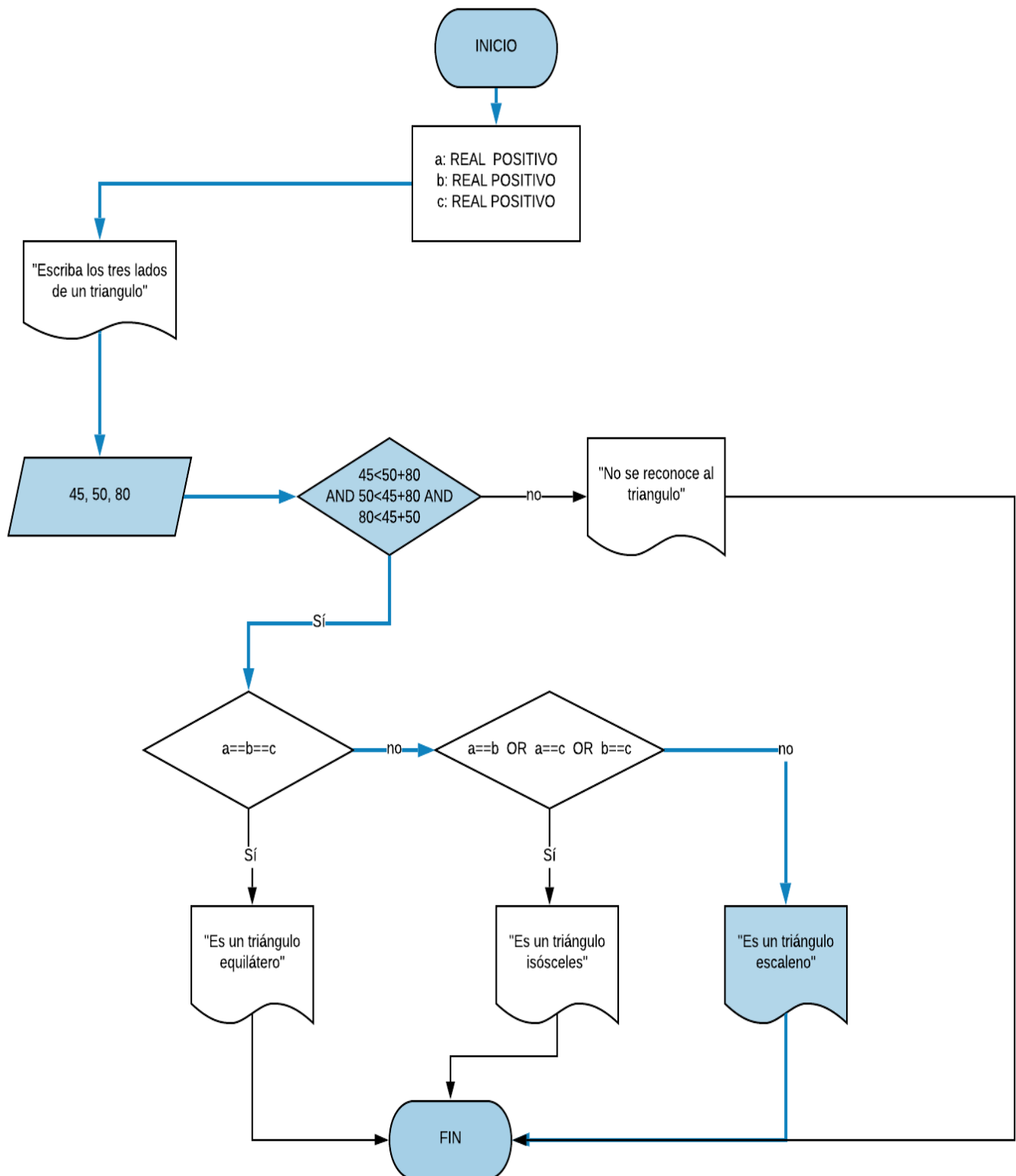
d.



En este caso el programa no reconoce a "-90" ya que no está establecido en los datos del proceso, por lo que hay un ERROR y el programa ya no se puede desarrollar.

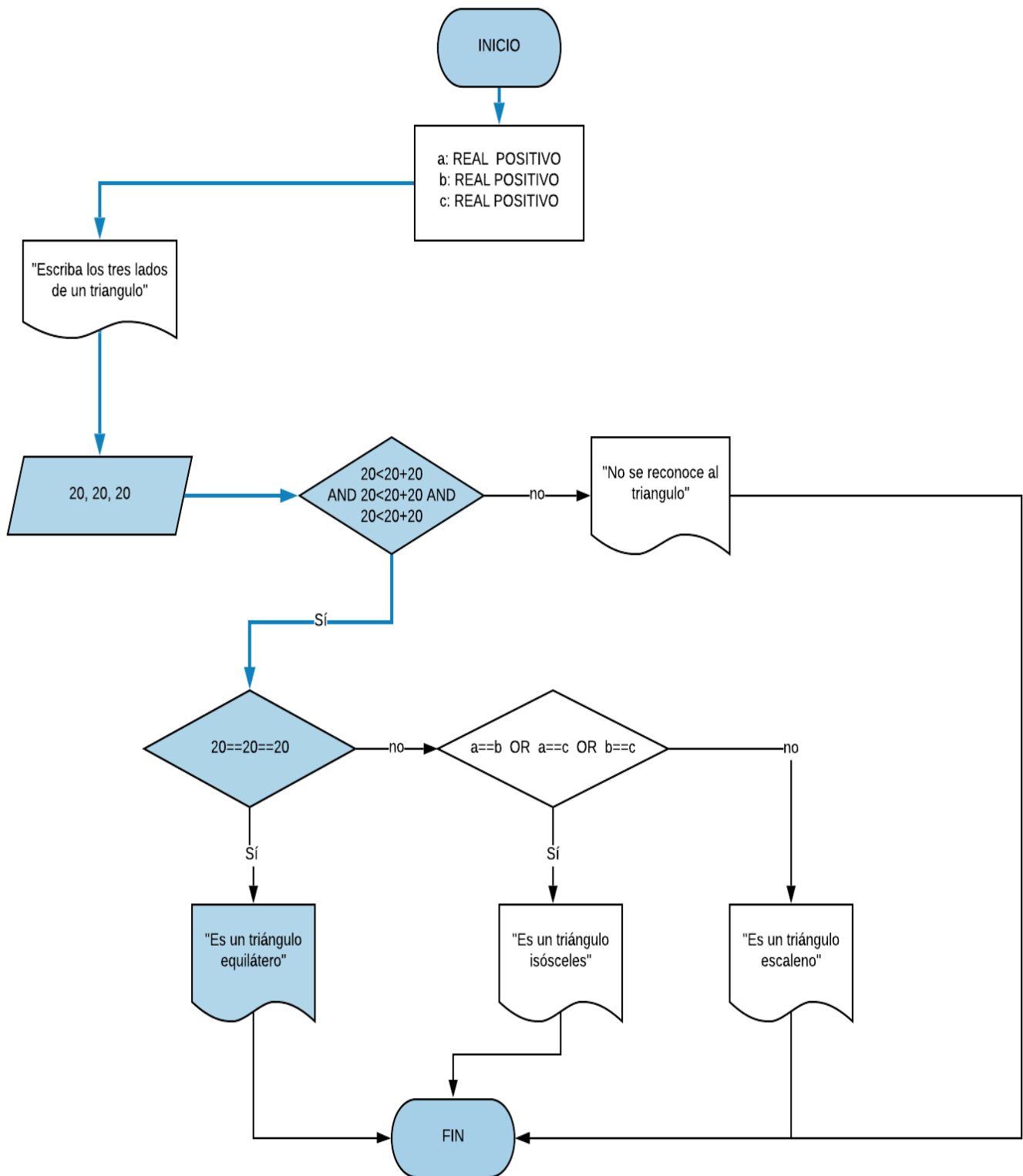
Tipos de triángulos

a.



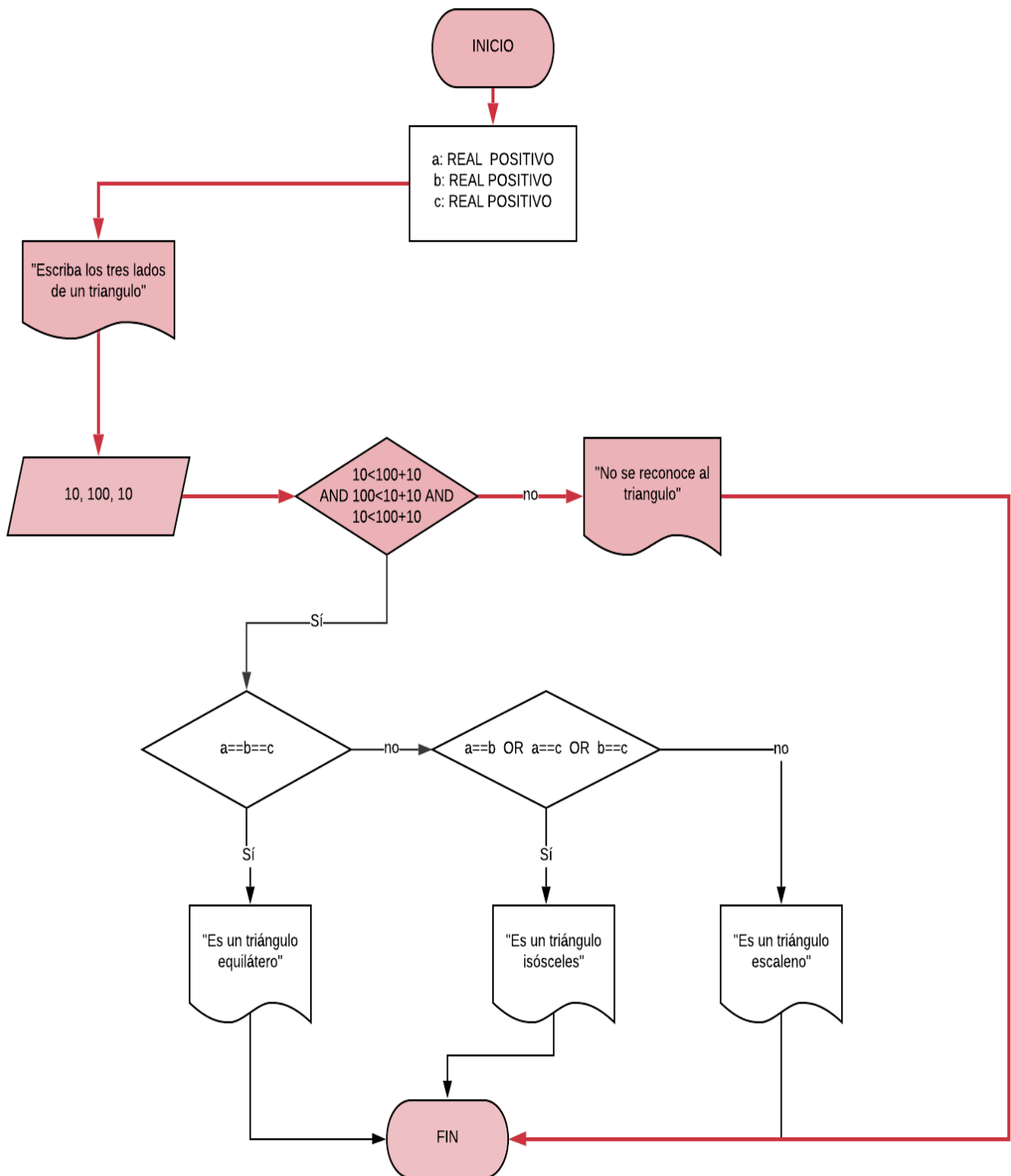
Los valores 40,50 y 80 cumplen con el primer condicional, por lo que continua a las siguientes condicionales, como no cumple con ninguno, se sabe que es un triángulo escaleno el programa se desarrolla correctamente.

b.



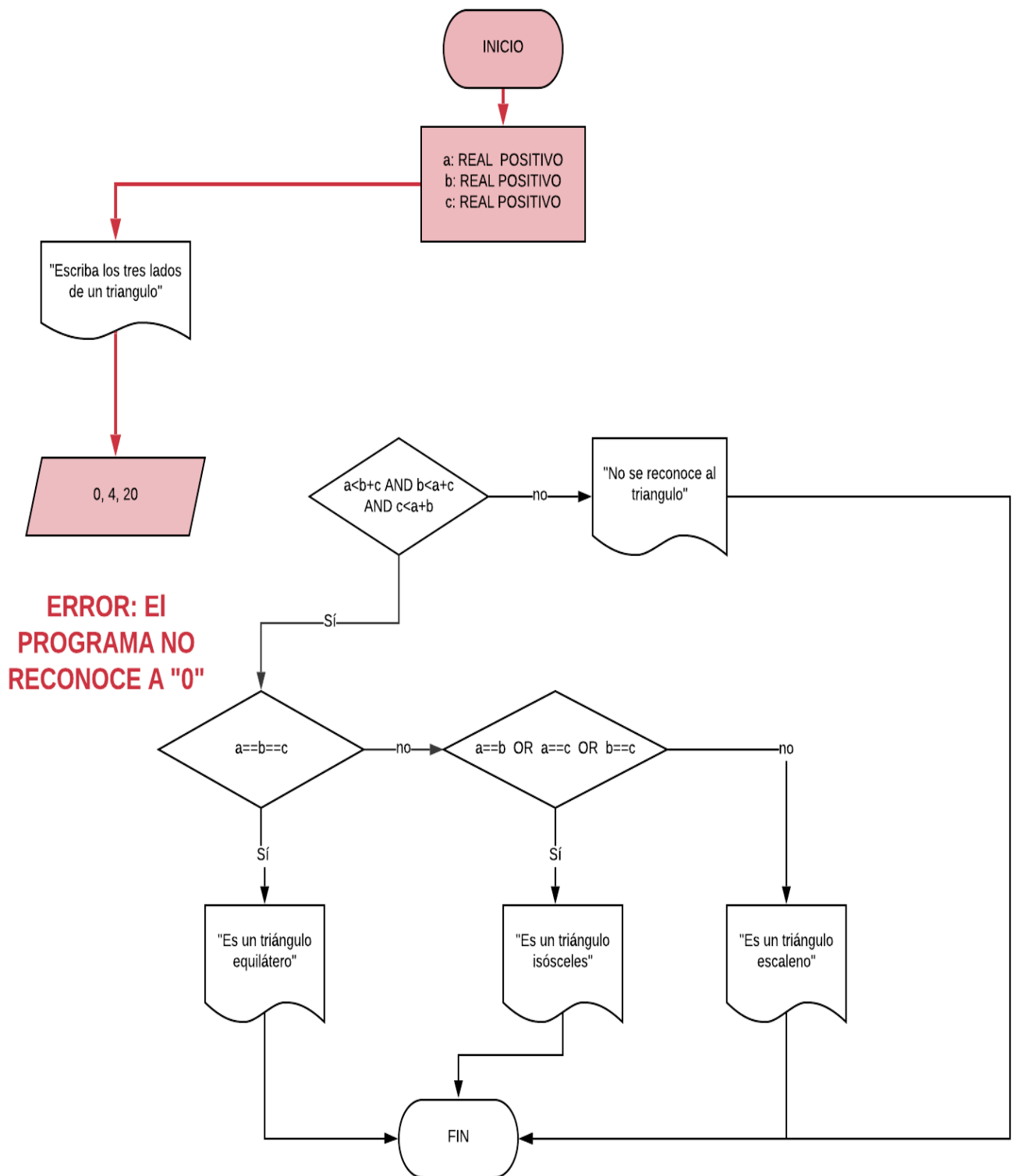
Los valores 20,20 y 20 cumplen con el primer condicional, por lo que continua a las siguientes condicionales, como cumple con el primero, se sabe que es un triángulo equilátero y el programa se desarrolla correctamente.

C.



Como los valores 10,100 y 10 no cumplen con el primer condicional, se sabe que no es un triángulo válido y termina el programa desarrollándose correctamente.

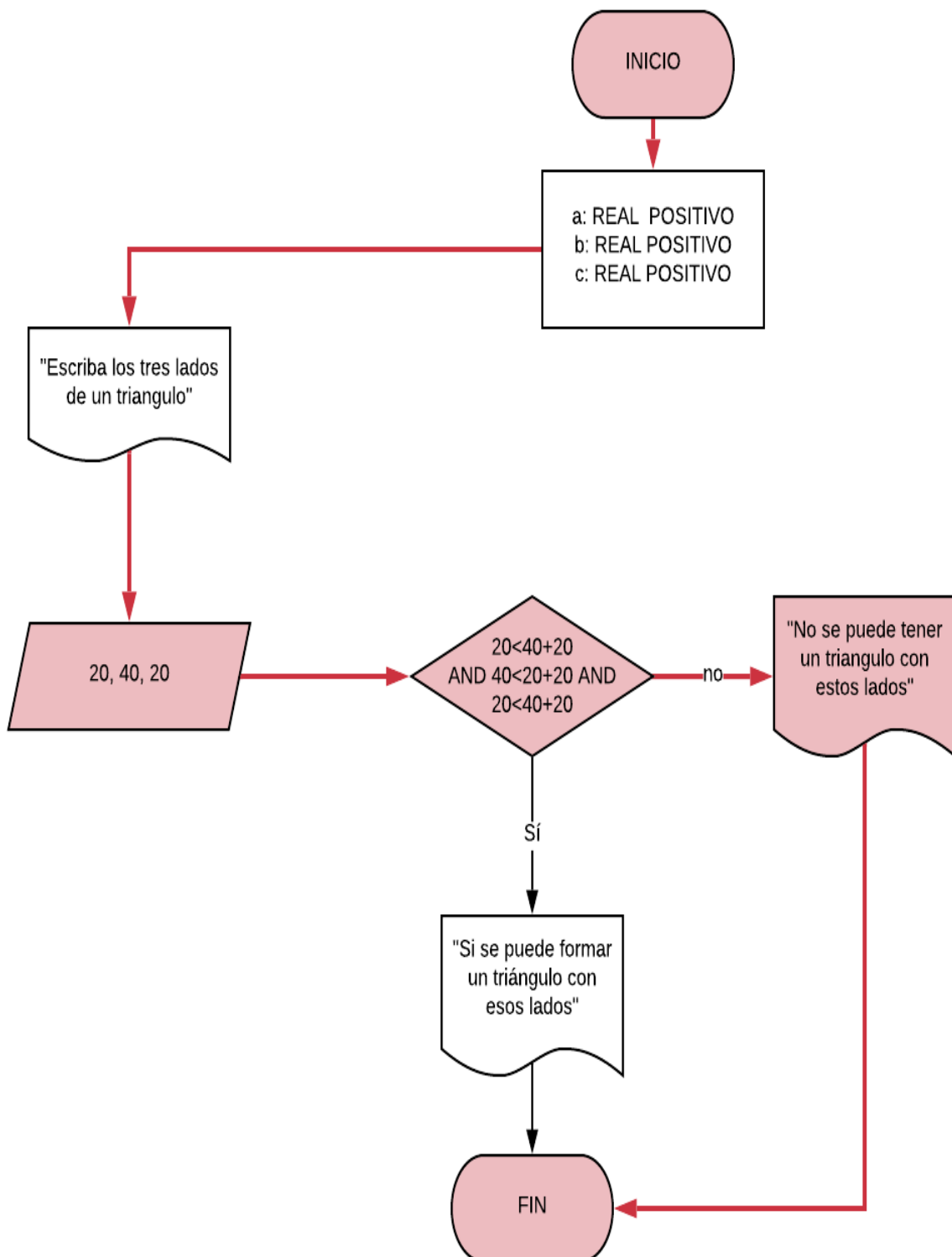
d.



El programa no reconoce al valor "0" por lo que se estableció en las variables del proceso por lo que hay un ERROR y el algoritmo no se desarrolla correctamente.

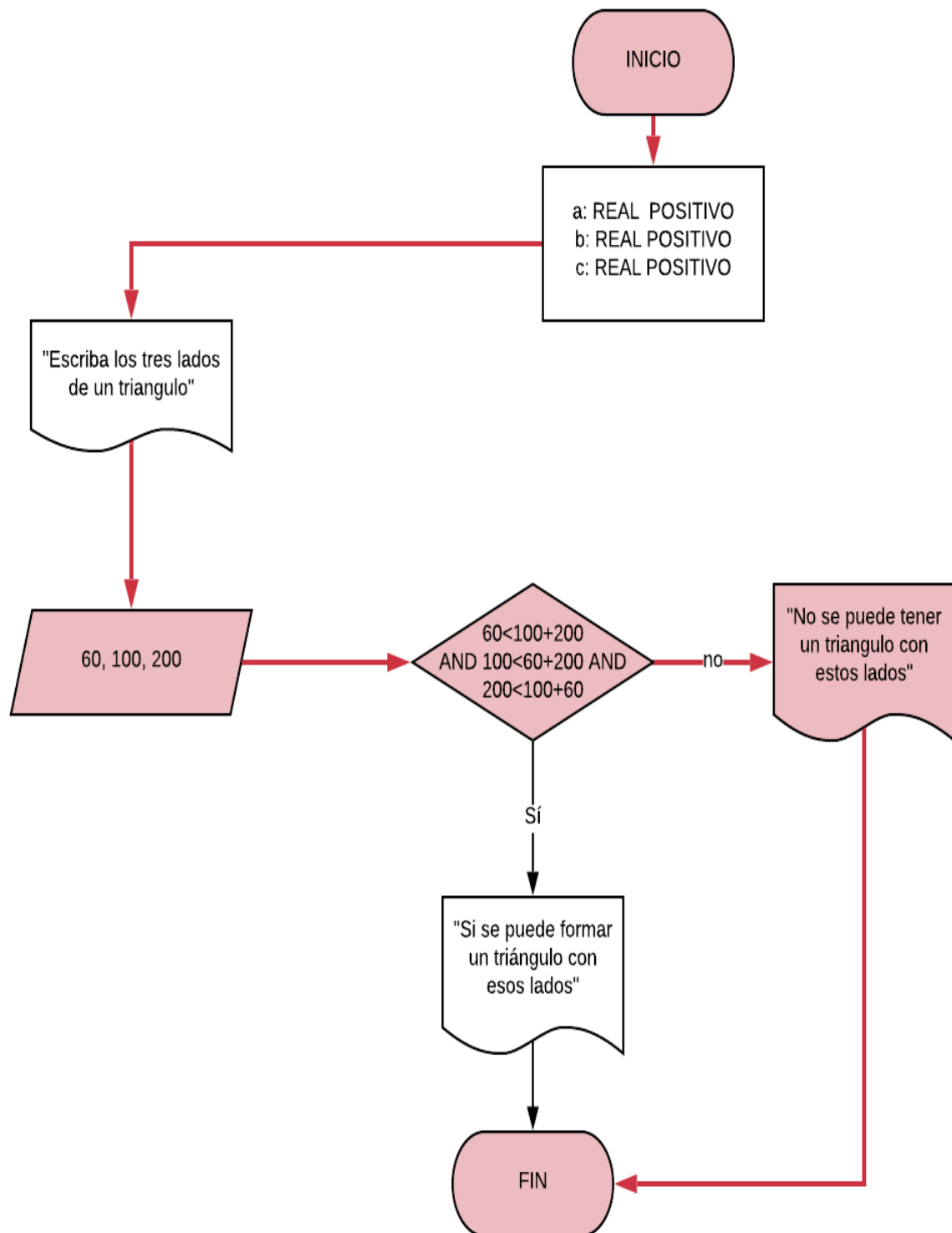
Triángulo aceptable

a.



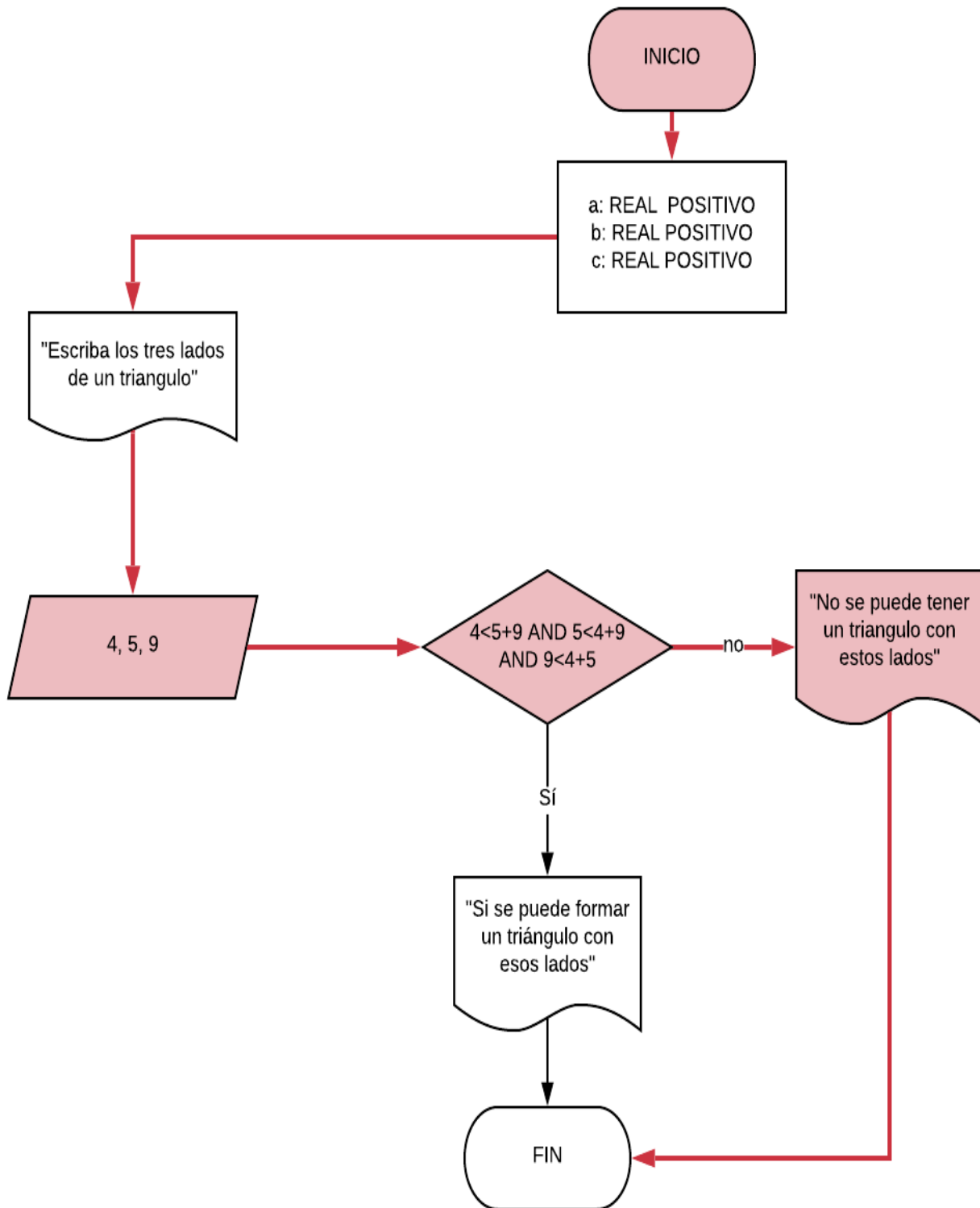
Como los valores 20, 40 y 20 no cumplen con el primer condicional, sabemos que no se puede obtener un triángulo con esos lados y el programa se desarrolla correctamente.

b.



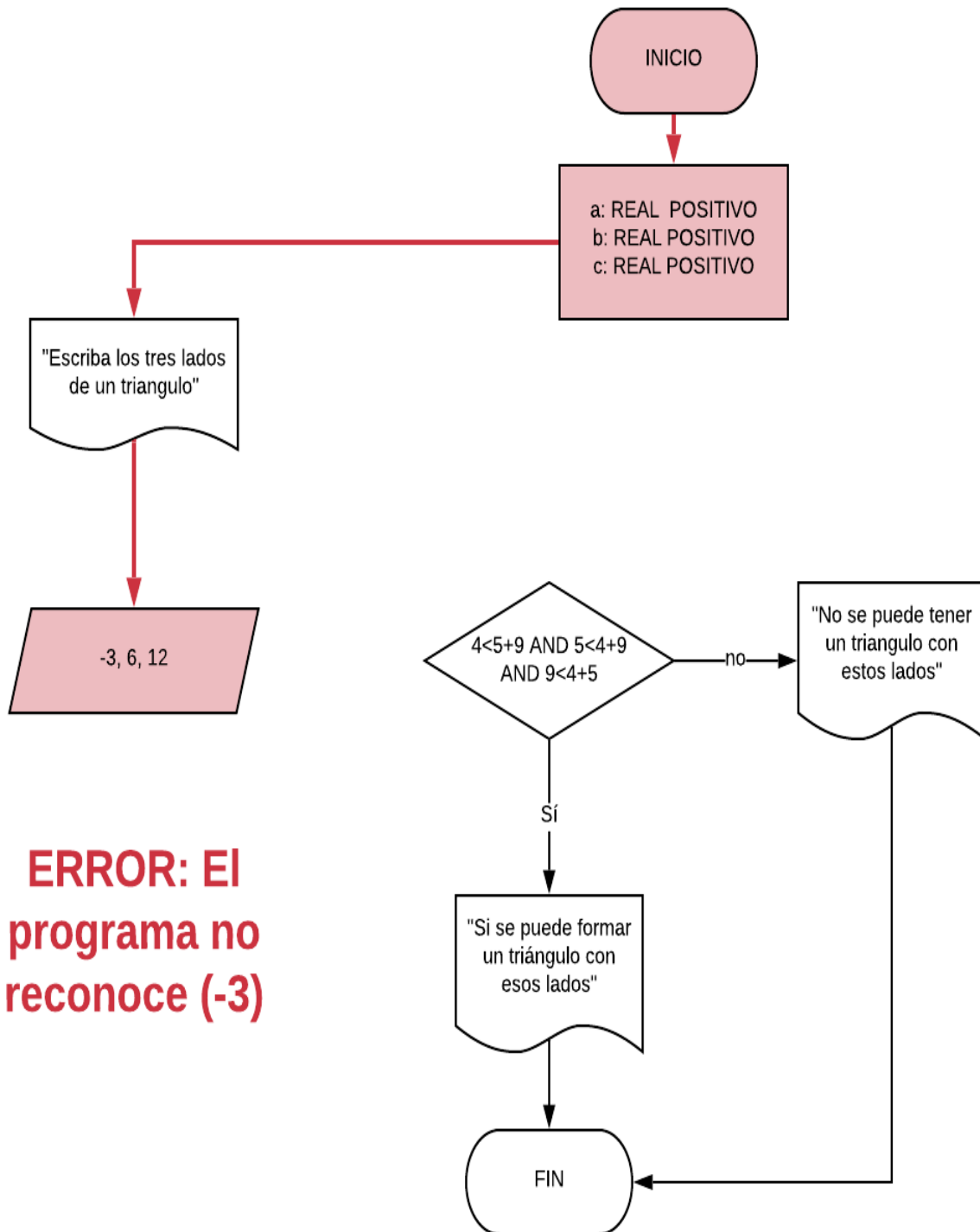
Como los valores 20, 40 y 20 no cumplen con el primer condicional, sabemos que no se puede obtener un triángulo con esos lados y el programa se desarrolla correctamente.

C.



Como los valores 20, 40 y 20 no cumplen con el primer condicional, sabemos que no se puede obtener un triángulo con esos lados y el programa se desarrolla correctamente.

d.



El programa no reconoce al valor “-3” por lo que se estableció en las variables del proceso, se produce un ERROR y el algoritmo no se desarrolla correctamente.

Conclusión

Cada cuadro dentro del diagrama de flujo tiene un significado o separación que nos permite conocer el desarrollo interno del programa, por lo cual debemos ser cuidadosos en el uso de cada uno, unos ejemplos son: Rombo (desarrolla una interrogante), romboide (muestra la entrada del programa), rectángulo (proceso), óvalo (inicio, continuación o fin), flecha (conecta cuadros entre sí), entre otros.

Los diagramas de flujo son muy útiles porque nos permiten desarrollar un programa, algoritmo, sistema o proceso de una manera sencilla y fácil de entender. En el caso específico de la programación, con el apoyo del diagrama de flujo logramos encontrar errores en nuestro algoritmo, en especial cuando lo probamos con una situación en específico.