CONDICIONALES

Fundamentos de Informática

Grado en Ingeniería Química

Escuela Técnica Superior de Ingenieros Industriales y de Telecomunicación

UC | ETS de Ingenieros Industriales y de Telecomunicación

ESTRUCTURAS DE CONTROL CONDICIONALES

No todas las sentencias que forman los programas se ejecutan. Hay ocasiones en que un determinado conjunto de sentencias se deben ejecutar sólo si una determinada condición es cierta y sino no.

Los valores lógicos: constantes, variables y expresiones lógicas, permiten *controlar* la ejecución de las sentencias de un programa.

Hay dos tipos de expresiones lógicas: las expresiones lógicas relacionales y las expresiones lógicas combinacionales.

- Las expresiones lógicas relacionales comparan los valores de dos expresiones aritméticas o dos expresiones de tipo carácter.

- Las expresiones lógicas combinacionales representan operaciones lógicas entre constantes, variables y otras expresiones lógicas.

Expresiones lógicas relacionales

Las expresiones lógicas relacionales comparan los valores de dos expresiones aritméticas o dos expresiones de tipo carácter.

La evaluación de una expresión lógica relacional produce un resultado de tipo lógico: True o False

Sintaxis:

operando1 OPERADOR_LÓGICO_RELACIONAL *operando2*

- *operando* es una expresión, variable o constante aritmética o de tipo carácter.

Operador LÓGICO relacional	Significado
= =	Igual que
!=	Distinto que
<	Menor que
<=	Menor o igual que
>	Mayor que
>=	Mayor o igual que

Expresiones lógicas relacionales (ejemplos)

Sean la sentencias:
>>> i=3; j=5

OPERACIÓN	RESULTADO
>>> 3 <= i	True
>>> j**2-1 >= 0	True
>>> i = = j	False
>>> i != 10	True
>>> 'ANA' < 'PEPE'	True

No son válidas las siguientes expresiones lógicas relacionales:

>>> <= 5	Falta operando1
>>> i = 3	Es una sentencia de asignación

• No confundir el operador lógico relacional de igualdad ==

con el operador de asignación =

Expresiones lógicas combinacionales

Las expresiones lógicas combinacionales representan operaciones lógicas entre constantes, variables y otras expresiones lógicas.

La evaluación de una expresión lógica combinacional produce un resultado de tipo lógico: True o False

Sintaxis:

operando1 OPERADOR_LÓGICO_COMBINACIONAL *operando2*

- *operando* es una expresión relacional, variable lógica o constante lógica. *Operando1* no existe cuando el operador lógico usado es unario.

Operador	TIPO	SIGNIFICADO
not	Unario	Es lo opuesto a *operando2*
and	Binario	Es True si y sólo si *operando1* y *operando2* son True
or	Binario	Es True si, al menos, uno de los dos operandos es True

Tablas de verdad de los operadores lógicos combinacionales

a	b	not b	a and b	a or b
True	True	False	True	True
True	False	True	False	True
False	True	False	False	True
False	False	True	False	False

Precedencias lógicas-aritméticas

Expresiones lógicas combinacionales

Operador
( )
not
and
or

Mayor

Prioridad

Menor

Si una expresión lógica contiene dos o más operadores de la misma precedencia, and y or se evalúan de izda a derecha.

Cuando hay paréntesis anidados, éstos se evalúan desde el más interno hasta el más externo.

Combinación de expresiones

Operador	
**	Operadores aritméticos
+, - (unarios)	
*, /, //, %	
+, -	
>, >=, <, <=, ==, !=	Operadores lógicos relacionales
not	Operadores lógicos combinacionales
and	
or	

Mayor

Prioridad

Menor

Evaluación con cortocircuitos

Python evalúa en una expresión lógica lo mínimo para conocer el resultado.

Cuando el primer término de un **or** es cierto, no se evalúa el segundo término, pues, independientemente de lo que sea, el resultado es cierto.

Cuando el primer término de un **and** es falso, no se evalúa el segundo término, pues, independientemente de lo que sea, el resultado es falso.

Ejemplo:

if a == 0 or 1/a > 1 : no puede dar nunca error de división por cero, pues si a vale cero, la primera condición es cierta y ya no evalúa la segunda, pues el resultado es cierto.

if a != 0 and 1/a > 1 : no puede dar nunca error de división por cero, pues si a vale cero, la primera condición es falsa y ya no evalúa la segunda, pues el resultado es falso.

Leyes de Morgan y Rarezas de Python

not (op1 **and** op2) es equivalente a: not op1 **or** not op2
not (op1 **or** op2) es equivalente a: not op1 **and** not op2
Ejemplo:
not (nota < 0 **or** nota > 10) es equivalente a:
nota >=0 **and** nota <=10

Python evalúa 2 < 5 < 8 como 2 < 5 **and** 5 < 8 ,
como en la notación matemática habitual.

Sentencia de asignación lógica

Asigna un valor lógico (True o False) a una variable.

variable = expresión_lógica

- *variable* es el nombre de una variable, o elemento de lista.
- *expresión_lógica* es una expresión lógica válida.

Su funcionamiento es:

1. Se evalúa la expresión_lógica.
2. Se asigna el valor obtenido a la *variable* y, por tanto, la *variable* es del tipo *bool*.

Ejemplo:

```
>>> i = 10
>>> log1 = (i == 10)
```

Bloque if (I)

Permite que un bloque de sentencias (puede ser sólo una) sea ejecutado si y sólo si el valor de una expresión lógica es cierta. Si la expresión lógica es falsa se salta ese bloque de sentencias y se ejecuta la primera sentencia sin indentación por debajo.

Sintaxis:

if *expresión lógica*:
    bloque

if *expresión lógica*: sentencia

Notar que *bloque está indentado*.

if *expresión lógica*:
    bloque1
else:
    bloque2

La estructura del bloque if puede ser más complicada. A veces, se quiere ejecutar diferentes bloques de sentencias dependiendo de que otras tantas condiciones sean ciertas. Por ejemplo:

if *expresión lógica1*:
    *bloque*1
elif *expresión lógica2*:
    bloque2
else:
    bloque3

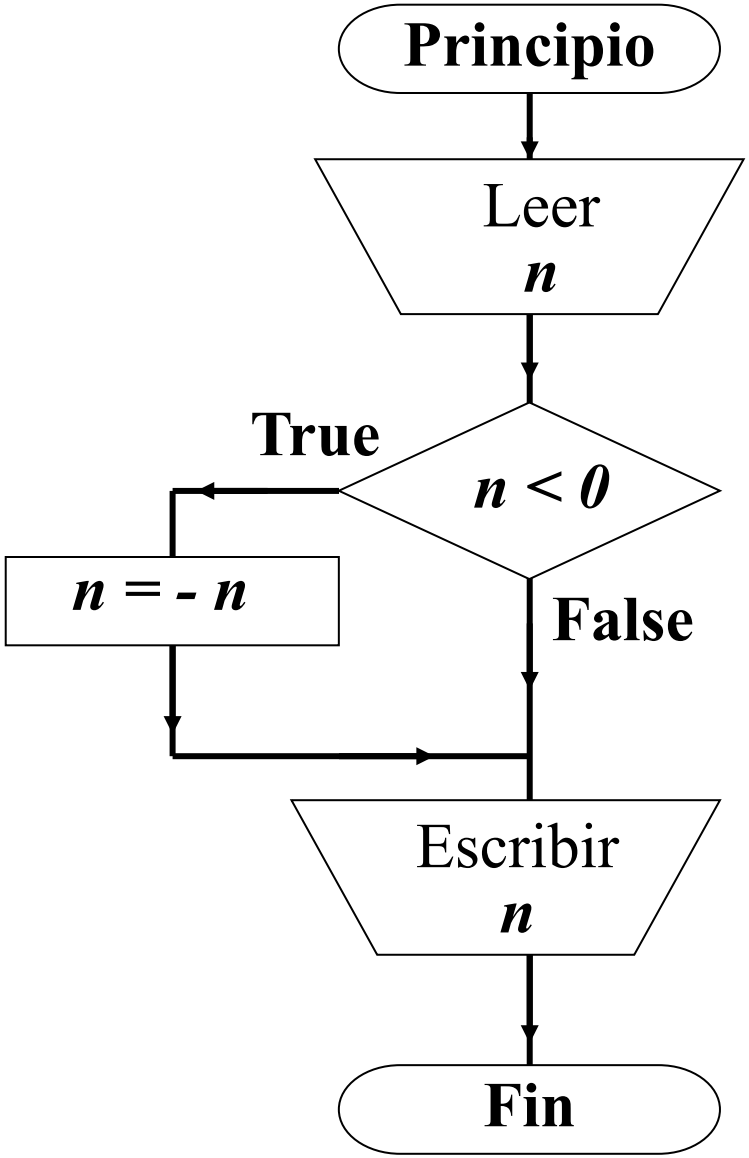
Bloque if (II)

La forma general responde a la siguiente sintaxis:

```
if expresión lógica1:
        bloque1
[elif expresión lógica2:
        bloque2]
…
[else:
        bloque3]
```

- Cuando una expresión lógica es cierta se ejecuta el bloque de sentencias correspondiente y se salta a la primera sentencia ejecutable sin indentación por debajo.
- Cuando todas las expresiones son falsas y el bloque if incluye la cláusula else se ejecuta su bloque de sentencias.
- Cuando el bloque if no incluye la cláusula else y ninguna de las expresiones lógicas son ciertas no se ejecuta ninguno de los bloques de sentencias dados.

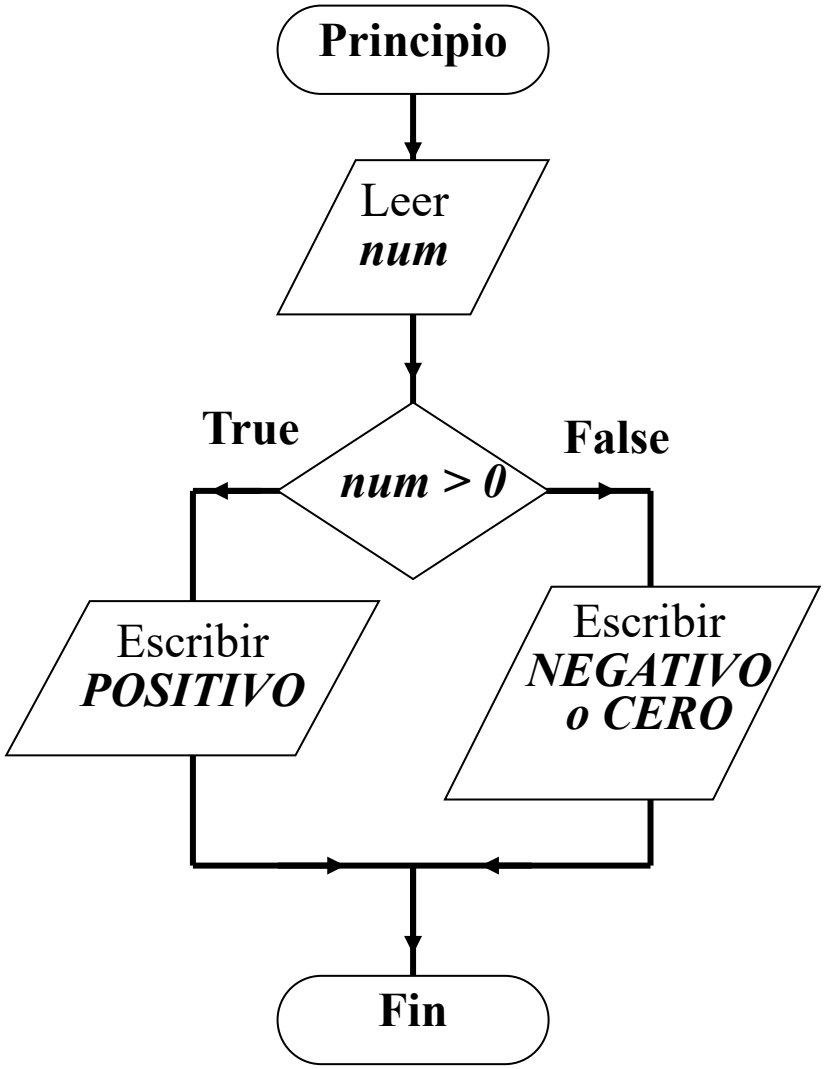
Ejemplo de if (Valor absoluto)



if condicion: sentencia

Ejercicio 1

Pedir un número real por teclado y escribir si es positivo o no.




```
if condicion:
        bloque1
else:
        bloque2
```

Ejercicio 1

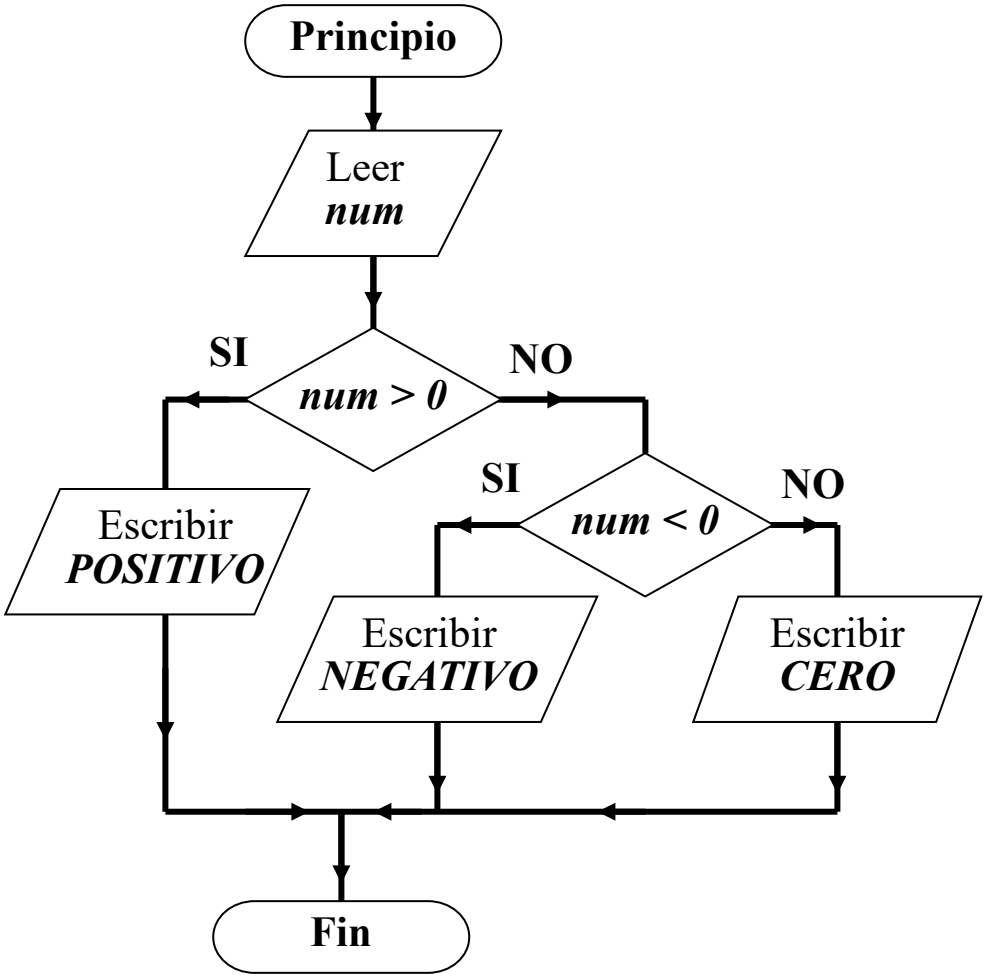
cap2_1.py - C:\DATOS\Curso 2022-2023\GIQ\LAB_PYTHON_22_23\cap2_Condiciones\resueltos\cap2_1.py (3.10.7)

File Edit Format Run Options Window Help

```
1 #PB sep-2022
2 num=float(input("Dame un numero:"))
3 if num>0:
4     print("Es positivo")
5 else:
6     print("Es negativo o cero")
```

Ejercicio 2

Pedir un número real por teclado y escribir si es positivo, negativo o cero.




```
if condicion1:
        bloque1
elif condicion2:
        bloque2
else:
        bloque3
```

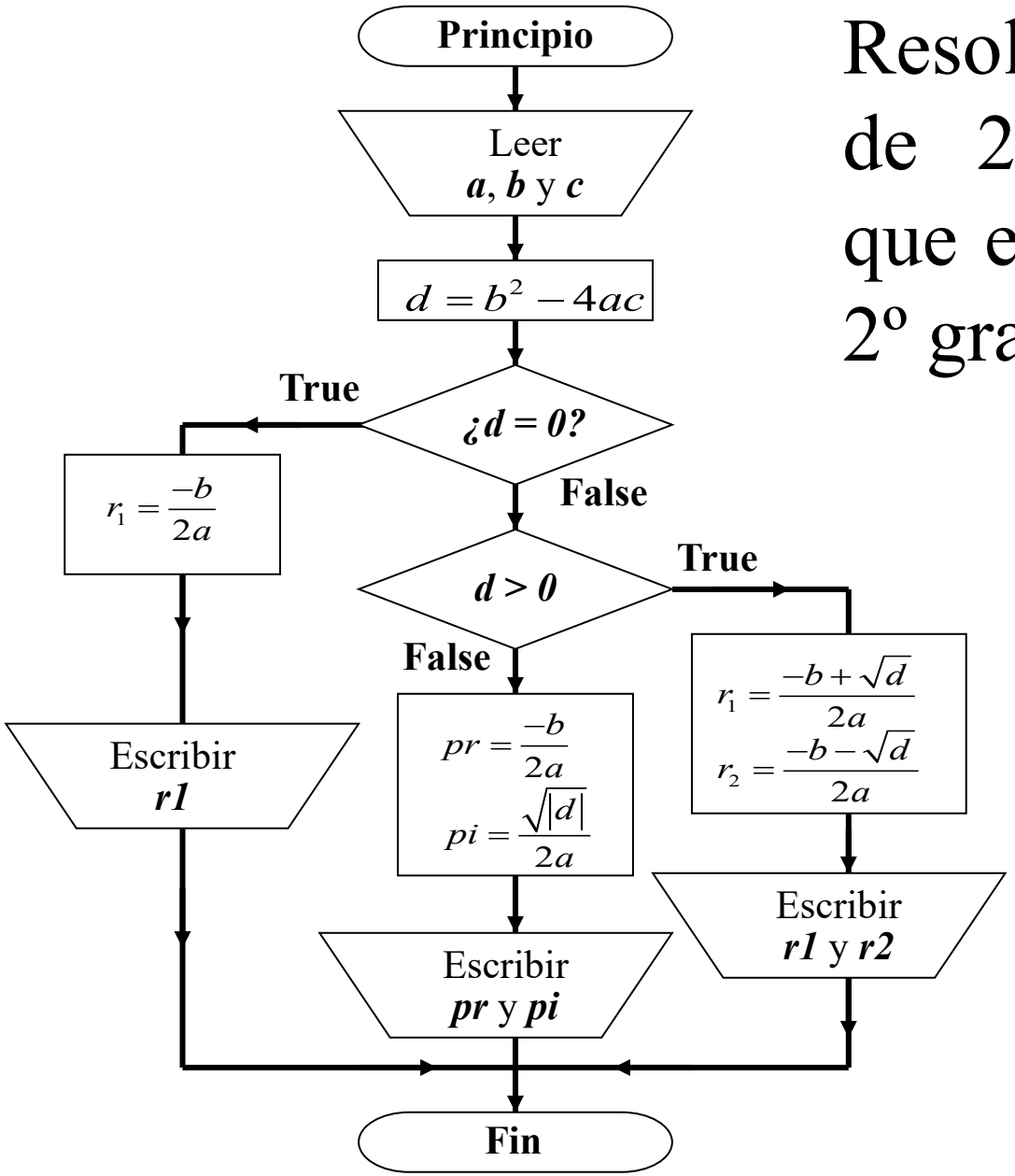
Ejercicio 2

cap2_2.py - C:\DATOS\Curso 2022-2023\GIQ\LAB_PYTHON_22_23\cap2_Condiciones\resueltos\cap2_2.py (3.10.7)

File Edit Format Run Options Window Help

```
1 #PB sep-2022
2 num=float(input("Dame un numero:"))
3 if num>0:
4     print("Es positivo")
5 elif num<0:
6     print("Es negativo")
7 else:
8     print("Es cero")
```

Ejercicio 3



Resolver una ecuación de 2º grado. Suponer que es efectivamente de 2º grado (A≠0).

```
if condicion1:
        bloque1
elif condicion2:
        bloque2
else:
        bloque3
```

Ejercicio 3

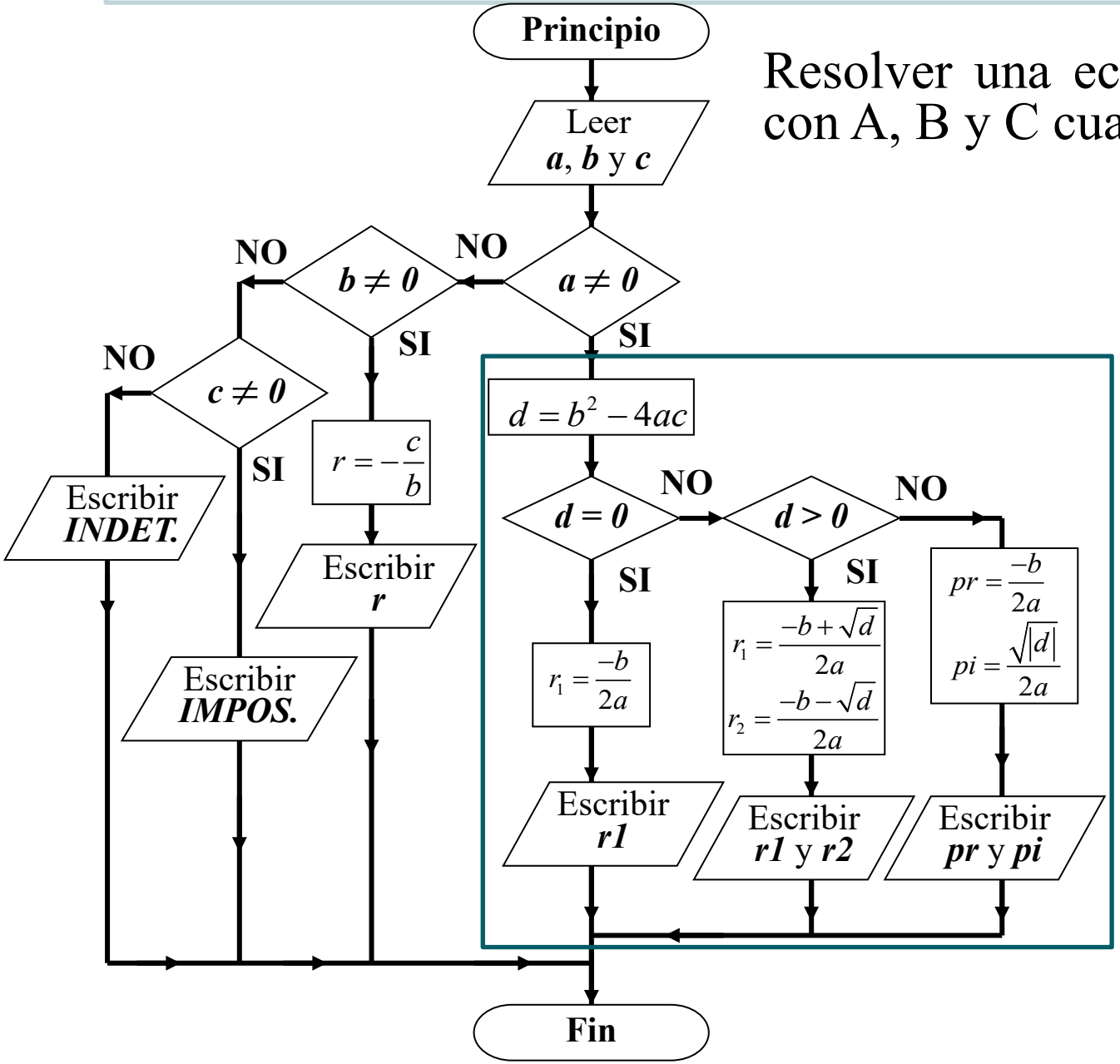
cap2_3.py - C:\DATOS\Curso 2022-2023\GIQ\LAB_PYTHON_22_23\cap2_Condiciones\resueltos\cap2_3.py (3.10.7)

File Edit Format Run Options Window Help

```
1  #PB sep-2022
2  # Suponer que es efectivamente de 2° grado (A≠0).
3  from math import sqrt
4
5  a=float(input("Dame a, x**2, con 'a' distinto de cero:"))
6  b=float(input("Dame b, x**1:"))
7  c=float(input("Dame c, x**0:"))
8
9  d=b**2-4*a*c
10 # discriminate
11 if d==0:
12     r1=-b/(2*a)
13     r2=r1
14     print("Soluciones reales dobles r1=r2:",r1,r2)
15 elif d>0:
16     r1=(-b+sqrt(d))/(2*a)
17     r2=(-b-sqrt(d))/(2*a)
18     print("Las raíces del polinomio son:")
19     print("r1=",r1,"r2=",r2)
20 else:
21     preal=-b/(2*a)
22     pimg=sqrt(abs(d))/(2*a)
23     print("Soluciones complejas")
24     print("Parte real:",preal)
25     print("Parte imaginaria:",pimg)
```

Ejercicio 4

Resolver una ecuación de 2º grado con A, B y C cualquier valor.



Ejercicio 4

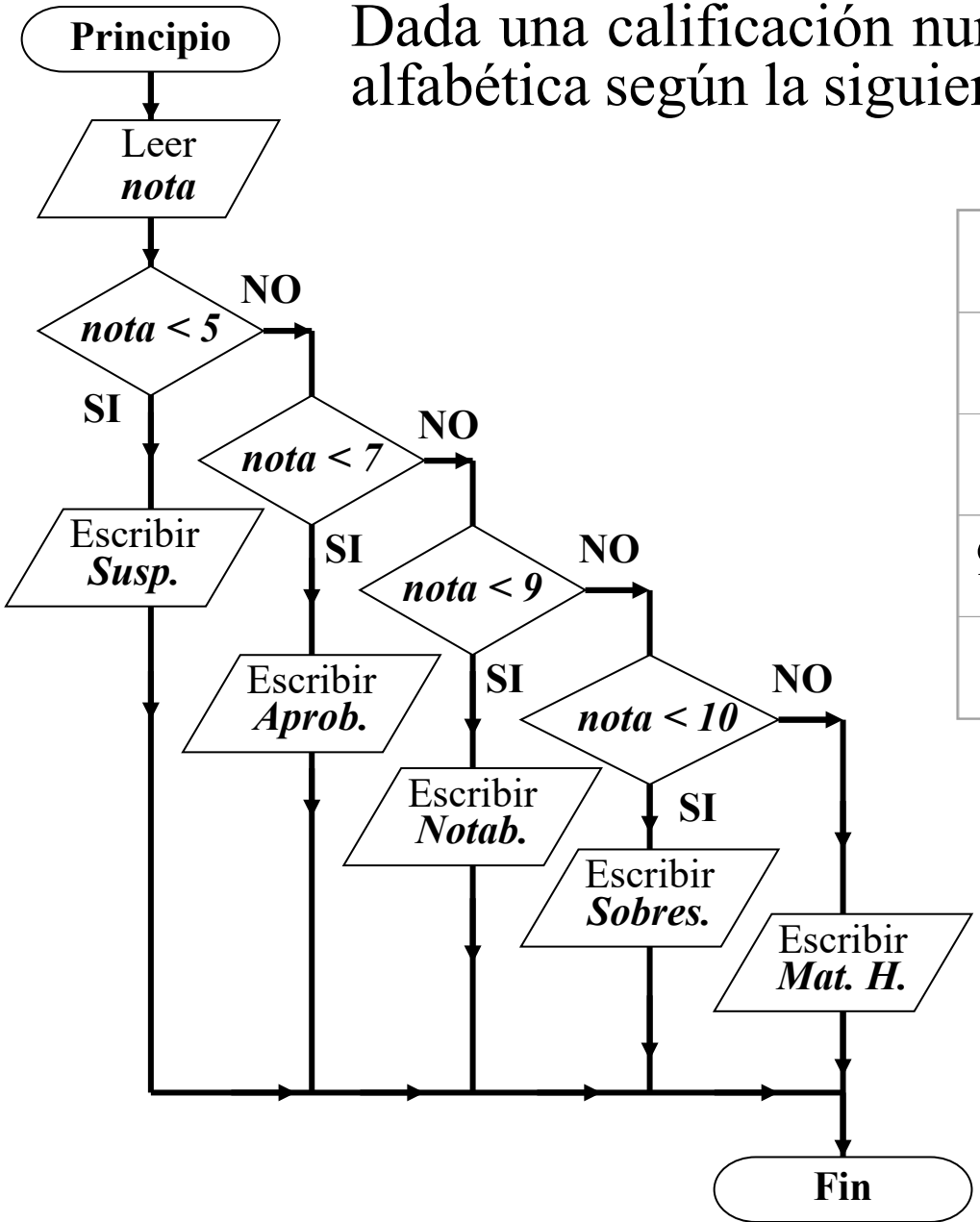
cap2_4.py - C:\DATOS\Curso 2022-2023\GIQ\LAB_PYTHON_22_23\cap2_Condiciones\resueltos\cap2_4.py (3.10.7)

File Edit Format Run Options Window Help

```
#PB sep-2022
from math import sqrt,fabs
print("Introduce a, b, c, con a, b, c cualquier valor")
a=float(input("Dame a: "))
b=float(input("Dame b: "))
c=float(input("Dame c: "))

if a!=0:
    d=b**2-4*a*c
    # discriminante
    if d==0:
        r1=-b/(2*a)
        r2=r1
        print("Soluciones reales dobles r1=r2: ",r1,r2)
    elif d>0:
        r1=(-b+sqrt(d))/(2*a)
        r2=(-b-sqrt(d))/(2*a)
        print("Las raíces del polinomio son:")
        print("r1=",r1,"r2=",r2)
    else:
        preal=-b/(2*a)
        pimg=sqrt(fabs(d))/(2*a)
        print("Soluciones complejas")
        print("Parte real: ",preal)
        print("Parte imaginaria: ",pimg)
elif b!=0:
    r=-c/b
    print("Solución única r= ",r)
elif c!=0:
    print("Ecuación imposible")
else:
   print("Ecuación indeterminada")
```

Ejercicio 5



Dada una calificación numérica obtener la correspondiente alfabética según la siguiente clasificación:

0≤ Nota < 5	Suspenso
5≤ Nota < 7	Aprobado
7≤ Nota < 9	Notable
9≤ Nota < 10	Sobresaliente
Nota = 10	Matricula de Honor

Hipótesis: el usuario no se equivoca. Teclea un número en [0, 10]

Ejercicio 5

cap2_5a.py - C:\DATOS\Curso 2022-2023\GIQ\LAB_PYTHON_22_23\cap2_Condiciones\resuel

File Edit Format Run Options Window Help

```
#PB sep-2022
nota=float(input("Dame una nota"))
if nota<0 or nota >10:
#evaluación con cortocircuitos
    print("Nota incorrecta")
elif nota<5:
    print("SUSPENSO")
elif nota<7:
    print("Aprobado")
elif nota<9:
    print("Notable")
elif nota<10:
    print("Sobresaliente")
else:
    print("MH")
```

cap2_5b.py - C:\DATOS\Curso 2022-2023\GIQ\LAB_PYTHON_22_23\cap2_Condiciones\resuelt

File Edit Format Run Options Window Help

```
#PB sep-2022
#Ineficiente, pero correcto
nota=float(input("Dame una nota"))
if nota<0 or nota >10:
    print("Nota incorrecta")
elif 0<=nota<5:
    print("SUSPENSO")
elif 5<=nota<7:
    print("Aprobado")
elif 7<=nota<9:
    print("Notable")
elif 9<=nota<10:
    print("Sobresaliente")
else:
    print("MH")
```

Ejercicio 6

Realizar una operación entre cuatro posibles según se desee. El programa pide por teclado dos números y una operación escribiendo en pantalla el resultado de la misma.

Variables del programa:
a, b, c : float
oper : str (suma +, resta -, multiplicación *, división /)

Tener en cuenta que:
- ✓ Al dividir, el divisor ≠ 0
- ✓ El operador se puede equivocar al teclear la operación a realizar.

Por tanto, la condición tiene 6 ramas!!

Ejercicio 6

cap2_6.py - C:\DATOS\Curso 2022-2023\GIQ\LAB_PYTHON_22_23\cap2_Condiciones\resueltos\cap2_6.py (3.10.7)

File Edit Format Run Options Window Help

```
1  #PB sep-2022
2  #Calculadora rudimentaria
3  a=float(input("Dame un número "))
4  b=float(input("Dame otro número "))
5  oper=input("Dame una operación (+, -, *, /) ")
6
7  if oper=='+':
8      c=a+b
9      print("c= ",c)
10 elif oper=='-':
11     c=a-b
12     print("c= ",c)
13 elif oper=='*':
14     c=a*b
15     print("c= ",c)
16 elif oper=='/' and b!=0:
17     c=a/b
18     print("c= ",c)
19 elif oper=='/' and b==0:
20     print("No se puede dividir entre cero")
21 else:
22     print("No es una operación válida")
```

Ejercicio 7

Presentar un menú en pantalla con las 3 opciones siguientes:

Pulsa 1 para ejecutar bloque 1.

Pulsa 2 para ejecutar bloque 2.

Pulsa 3 para salir del programa.

Al pulsar 3, escribir en pantalla: “Gracias por usar este programa”. Pulsando cualquier otro número diferente de 1, 2 o 3, escribir en pantalla: “Opción incorrecta”.

Ejercicio 7

cap2_7.py - C:\DATOS\Curso 2022-2023\GIQ\LAB_PYTHON_22_23\cap2_Condiciones\resueltos\cap2_7.py (3.10.7)

File Edit Format Run Options Window Help

```
1  #PB sep-2022
2  #Presentar un menú en pantalla con opciones
3  print("Dame una opcion")
4  print("Pulsa 1 para ejecutar bloque 1")
5  print("Pulsa 2 para ejecutar bloque 2")
6  print("Pulsa 3 para salir")
7  num=int(input())
8
9  if num==1:
10     print("Se hace el bloque 1")
11 elif num==2:
12     print("Se hace el bloque 2")
13 elif num==3:
14     print("Gracias por usar este programa")
15 else:
16     print("Opción incorrecta")
```