

TEMA 2: Sentencias de control de flujo: Condicionales, Bucles e Iteraciones

Dpto. de Matemática Aplicada a la Ingeniería Aeroespacial

Informática, 1^{er} semestre
2022 – 2023

Profesor: Juan Antonio Hernández Ramos (juanantonio.hernandez@upm.es)

Coordinador: Javier de Vicente Buendía (fj.devicente@upm.es)

Colaborador: Víctor Javier Llorente Lázaro (victorjavier.llorente@upm.es)

- SENTENCIAS CONDICIONALES

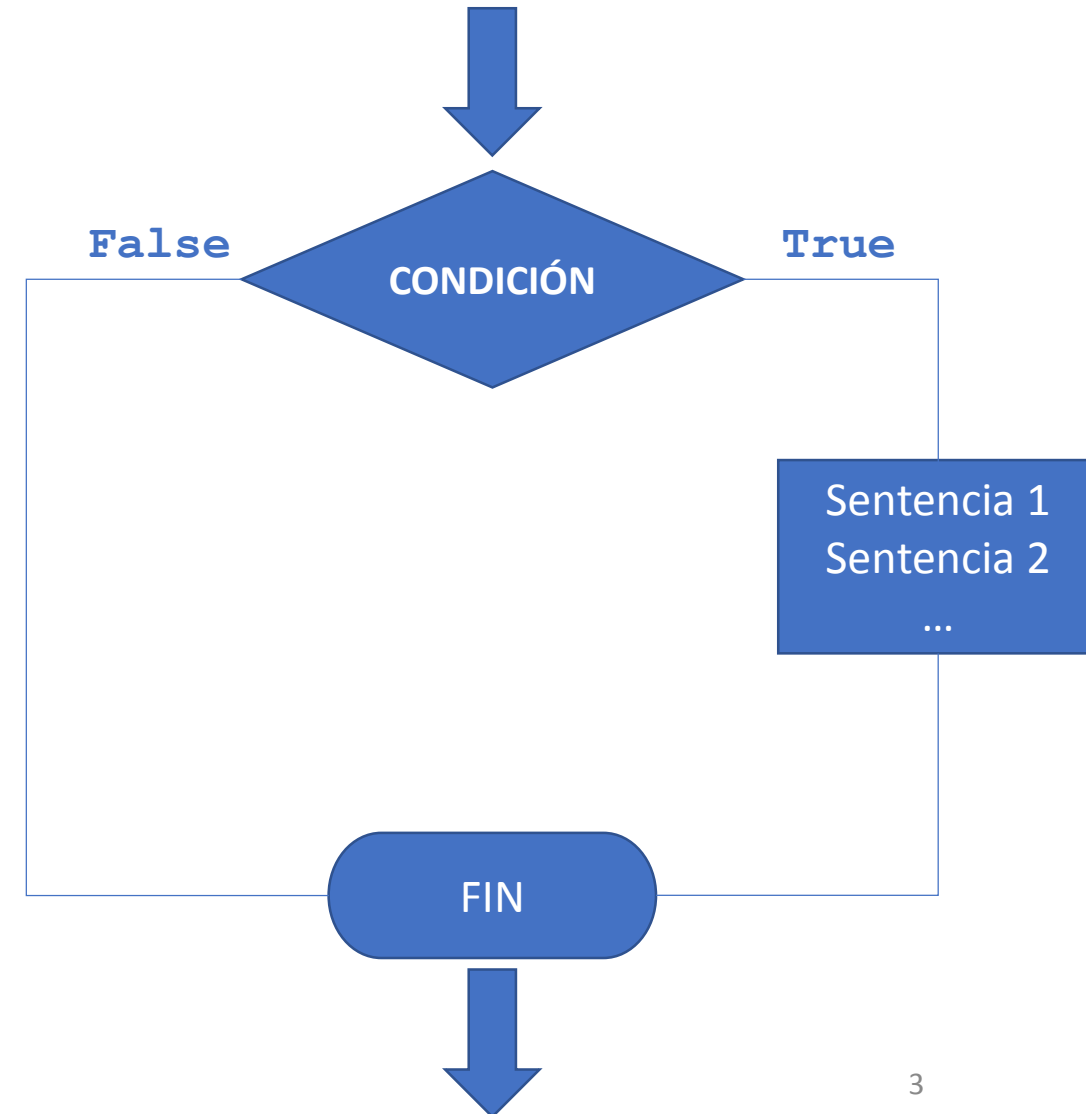
➤ `if <condición>:`
 `<sentencia1>`
 `<sentencia2>`
 ...

- El programa ejecutará las sentencias en el caso de que la condición evaluada sea `True`
- El sangrado, de 4 espacios, es automático cada vez que usamos `:`

- SENTENCIAS CONDICIONALES

```
➤ if <condición>:  
    <sentencia1>  
    <sentencia2>  
    ...
```

- El programa ejecutará las sentencias en el caso de que la condición evaluada sea `True`
- El sangrado, de 4 espacios, es automático cada vez que usamos `:`



- SENTENCIAS CONDICIONALES

➤ **if** <condición>:
 <sentencia1>
 <sentencia2>
 ...

```
# Expresiones condicionales  
  
# if  
numero = int(input("Escribe un número:"))  
  
if (numero > 5):  
    print("el número ", numero, " es mayor que cinco")
```

```
Escribe un número:23  
El número 23 es mayor que cinco
```

- SENTENCIAS CONDICIONALES

➤ **if** <condición>:
 <sentencia1>
 <sentencia2>
 ...

```
# Expresiones condicionales

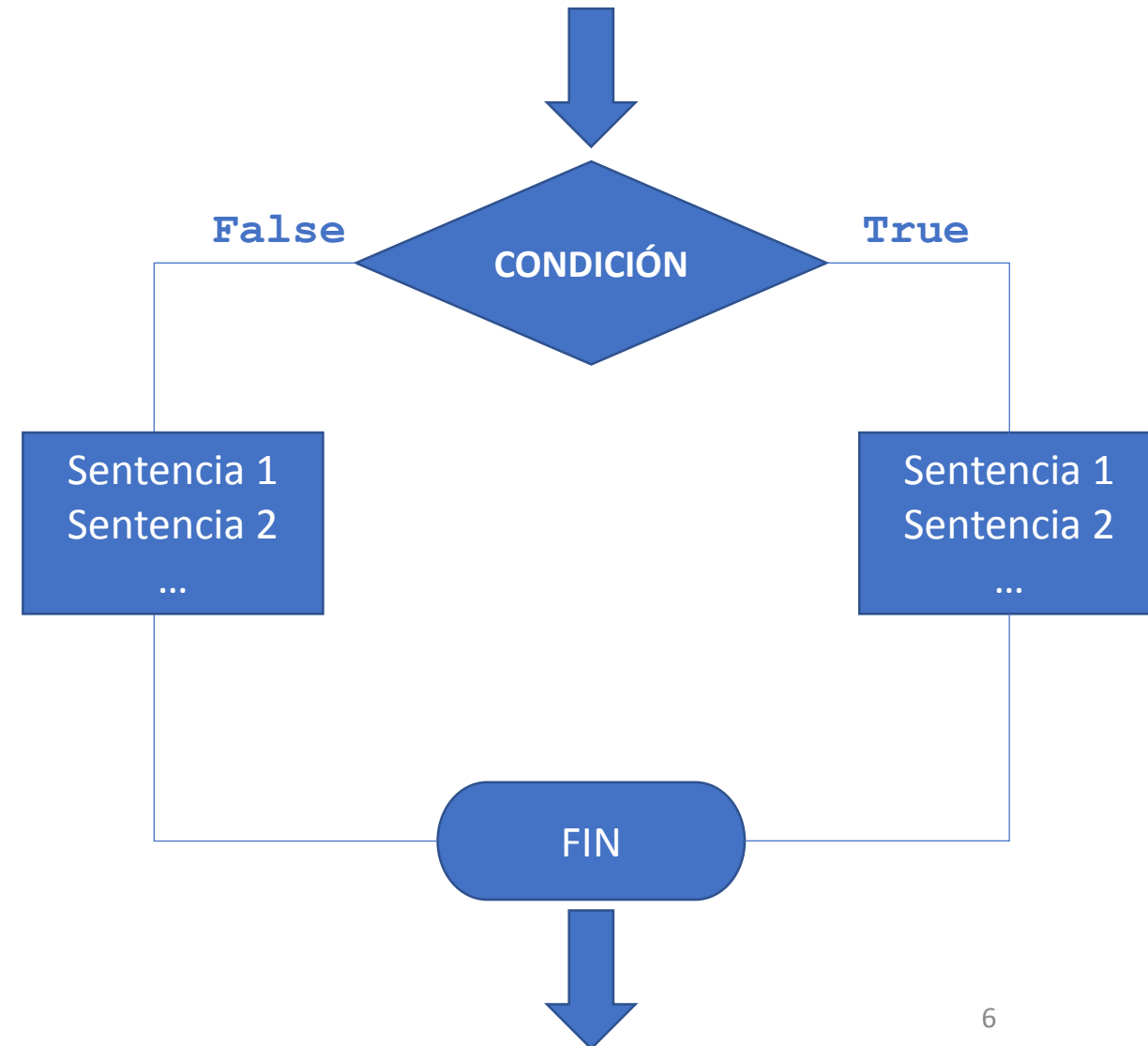
# if versión 2
nombre = input("Dime tu nombre: ")
numero = int(input("Escribe un número: "))

if (numero % 2 == 0):
    print("Hola ", nombre)
    print("el número ", numero, " es par")
```

```
Dime tu nombre: Juan
Escribe un número: 18
Hola Juan
El número 18 es par
```

- SENTENCIAS CONDICIONALES

```
➤ if <condición>:  
    <sentencia1>  
    <sentencia2>  
    ...  
else:  
    <sentencia1>  
    <sentencia2>  
    ...
```



- SENTENCIAS CONDICIONALES

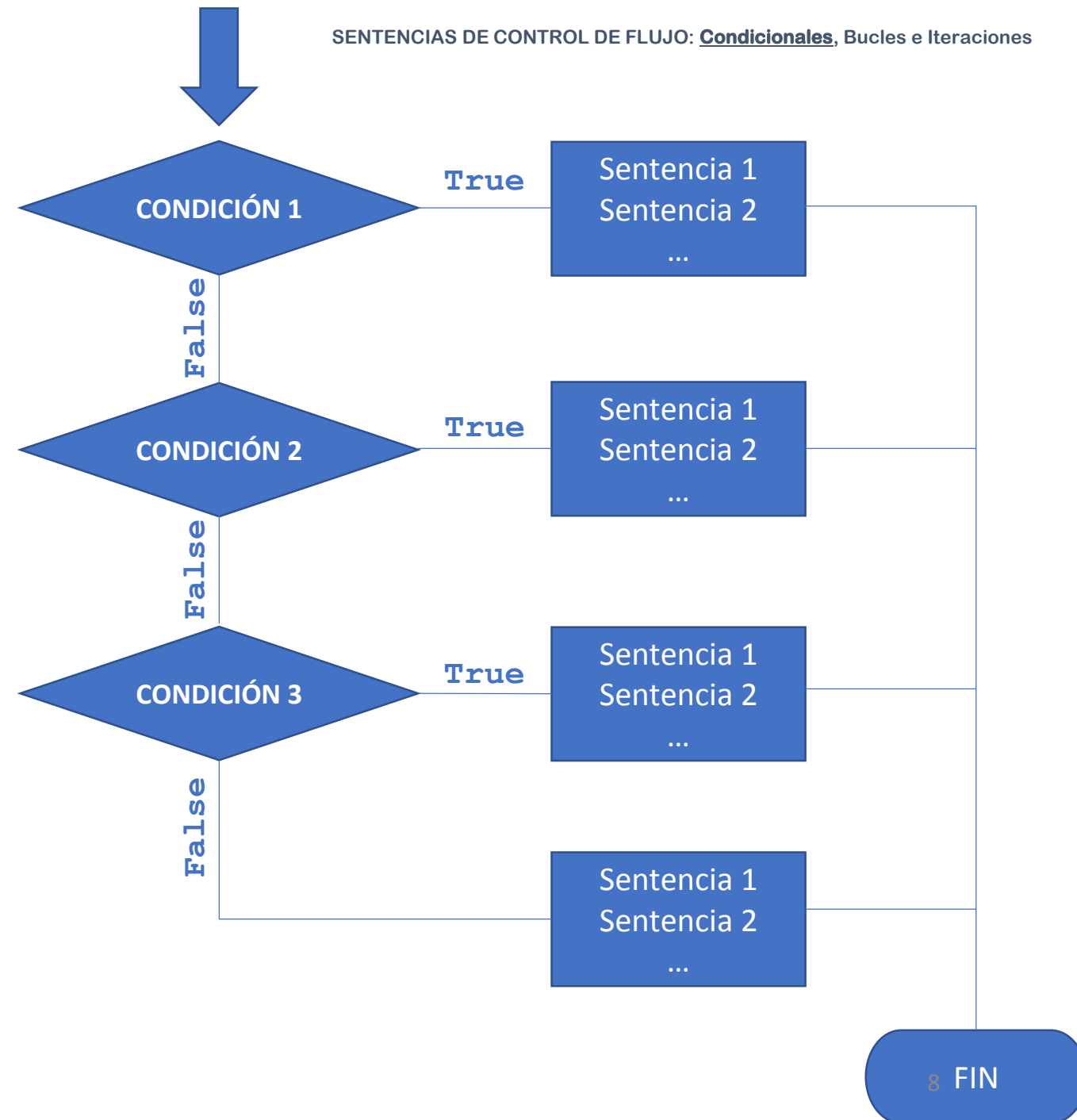
```
➤ if <condición>:  
    <sentencia1>  
    <sentencia2>  
    ...  
else:  
    <sentencia1>  
    <sentencia2>  
    ...
```

```
# Expresiones condicionales  
  
# if con else  
numero = int(input("Escribe un número: "))  
  
if (numero > 5):  
    print("el número ", numero, " es mayor que cinco")  
else:  
    print("el número ", numero, " es menor que cinco")
```

```
Escribe un número: 4  
El número 4 es menor que cinco
```

- SENTENCIAS CONDICIONALES

```
➤ if <condición>:  
    <sentencia1>  
    ...  
elif <condición>:  
    <sentencia1>  
    ...  
elif <condición>:  
    <sentencia1>  
    ...  
...  
else:  
    <sentencia1>  
    <sentencia2>  
    ...
```



• SENTENCIAS CONDICIONALES

```

➤ if <condición>:
    <sentencia1>
    ...
elif <condición>:
    <sentencia1>
    ...
elif <condición>:
    <sentencia1>
    ...
else:
    <sentencia1>
    <sentencia2>
    ...

```

```
# Expresiones condicionales
```

```
# if con elif
```

```
numero = int(input("Escribe un número: "))
```

```

if (numero % 2 == 0):
    print("el número ", numero, " es múltiplo de 2")
elif (numero % 3 == 0):
    print("el número ", numero, " es impar y múltiplo de 3")
elif (numero % 5 == 0):
    print("el número ", numero, " es impar, "
          "no es múltiplo de 3 pero si múltiplo de 5")
else:
    print("el número ", numero, " no es múltiplo "
          "de 2, de 3, ni de 5")

```

```
Escribe un número: 30
```

```
El número 30 es múltiplo de 2
```

```
Escribe un número: 15
```

```
El número 15 es impar y múltiplo de 3
```

```
Escribe un número: 49
```

```
El número 49 no es múltiplo de 2, de 3, ni de 5
```

```
Escribe un número: 35
```

```
El número 35 es impar, no es múltiplo de 3 pero si múltiplo de 5
```

- SENTENCIAS CONDICIONALES

➤ Ejemplo de `if` anidado

```
# Expresiones condicionales

# if anidado
numero = int(input("Escribe un número: "))

if (numero % 2 == 0):
    print("el número ", numero, " es par")
    if (numero % 3 == 0):
        print("además, el número ", numero, " es múltiplo de 3")
    elif (numero % 5 == 0):
        print("además, el número ", numero, " es múltiplo de 5")
else:
    print("el número ", numero, " es impar")
    if (numero % 3 == 0):
        print("además, el número ", numero, " es múltiplo de 3")
    elif (numero % 5 == 0):
        print("además, el número ", numero, " es múltiplo de 5")
```

Escribe un número: 5
el número 5 es impar
además, el número 5 es múltiplo de 5

Escribe un número: 24
el número 24 es par
además, el número 24 es múltiplo de 3

EJERCICIOS PROPUESTOS

1. Escribir un programa que permita convertir entre representaciones de ángulos en radianes y en grados ($\alpha_{\text{grad}} = 180\alpha_{\text{rad}} / \pi$).
NOTA: π se puede obtener del módulo `math` directamente o mediante la función intrínseca `arccos(-1)`
2. Escribir un programa que permita convertir entre escalas de temperatura Celsius, Kelvin y Fahrenheit ($T_C = T_K - 273.15$, $T_F = 9/5T_K - 459.67$).
3. Escribir un programa que pida un número natural inferior a 1000. Si el número introducido no cumple ese requisito el programa terminará con este mensaje por pantalla El número no es correcto. Si se cumple la condición el programa devolverá la descomposición del número en centenas, decenas y unidades.
4. Se considera la función $f : \mathbb{R} \rightarrow \mathbb{R}$ definida del modo siguiente:

$$f(x) = \begin{cases} \exp(-x) \sin(x^2) & \text{si } x < 0 \\ \log(1 + x^3) \cos(x) & \text{si } 0 \leq x \leq \frac{\pi}{2} \\ \frac{x - 5}{1 + x^2} & \text{si } x > \frac{\pi}{2} \end{cases}$$

Dado un valor de entrada x , devuelva el valor de la función en dicho punto.

TEMA 2: Sentencias de control de flujo: Condicionales, Bucles e Iteraciones

Dpto. de Matemática Aplicada a la Ingeniería Aeroespacial

Informática, 1^{er} semestre
2022 – 2023

Profesor: Juan Antonio Hernández Ramos (juanantonio.hernandez@upm.es)

Coordinador: Javier de Vicente Buendía (fj.devicente@upm.es)

Colaborador: Víctor Javier Llorente Lázaro (victorjavier.llorente@upm.es)

- SENTENCIAS ITERATIVAS (BUCLE *WHILE*)

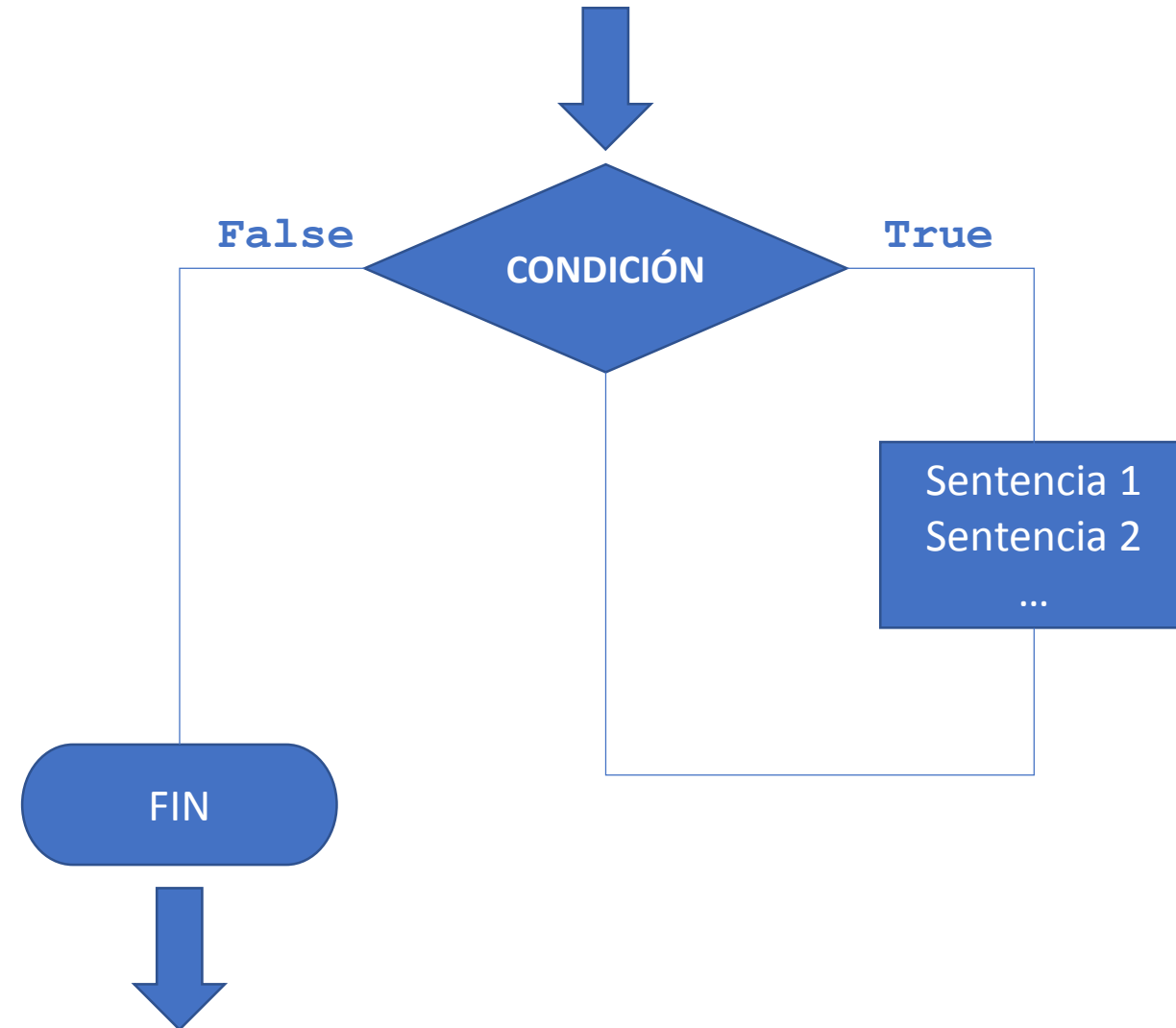
```
➤ while <condición>:  
    <sentencia 1>  
    <sentencia 2>  
    ...
```

- El programa ejecutará las sentencias mientras la condición sea `True`
- El sangrado, de 4 espacios, es automático cada vez que usamos `:`
- **PRECAUCIÓN:** si las sentencias no hacen que la condición sea `False`, el bucle *while* no saldrá

- SENTENCIAS ITERATIVAS (BUCLE *WHILE*)

```
➤ while <condición>:  
    <sentencia 1>  
    <sentencia 2>  
    ...
```

- El programa ejecutará las sentencias mientras la condición sea `True`
- El sangrado, de 4 espacios, es automático cada vez que usamos `:`
- **PRECAUCIÓN:** si las sentencias no hacen que la condición sea `False`, el bucle *while* no saldrá



- SENTENCIAS ITERATIVAS (BUCLE *WHILE*)

➤ **while** <condición>:
 <sentencia 1>
 <sentencia 2>
 ...

```
# Expresiones iterativas  
  
# Bucles tipo while  
numero = 1  
while (numero < 10):  
    print(numero)  
    numero += 1
```

1
2
3
4
5
6
7
8
9

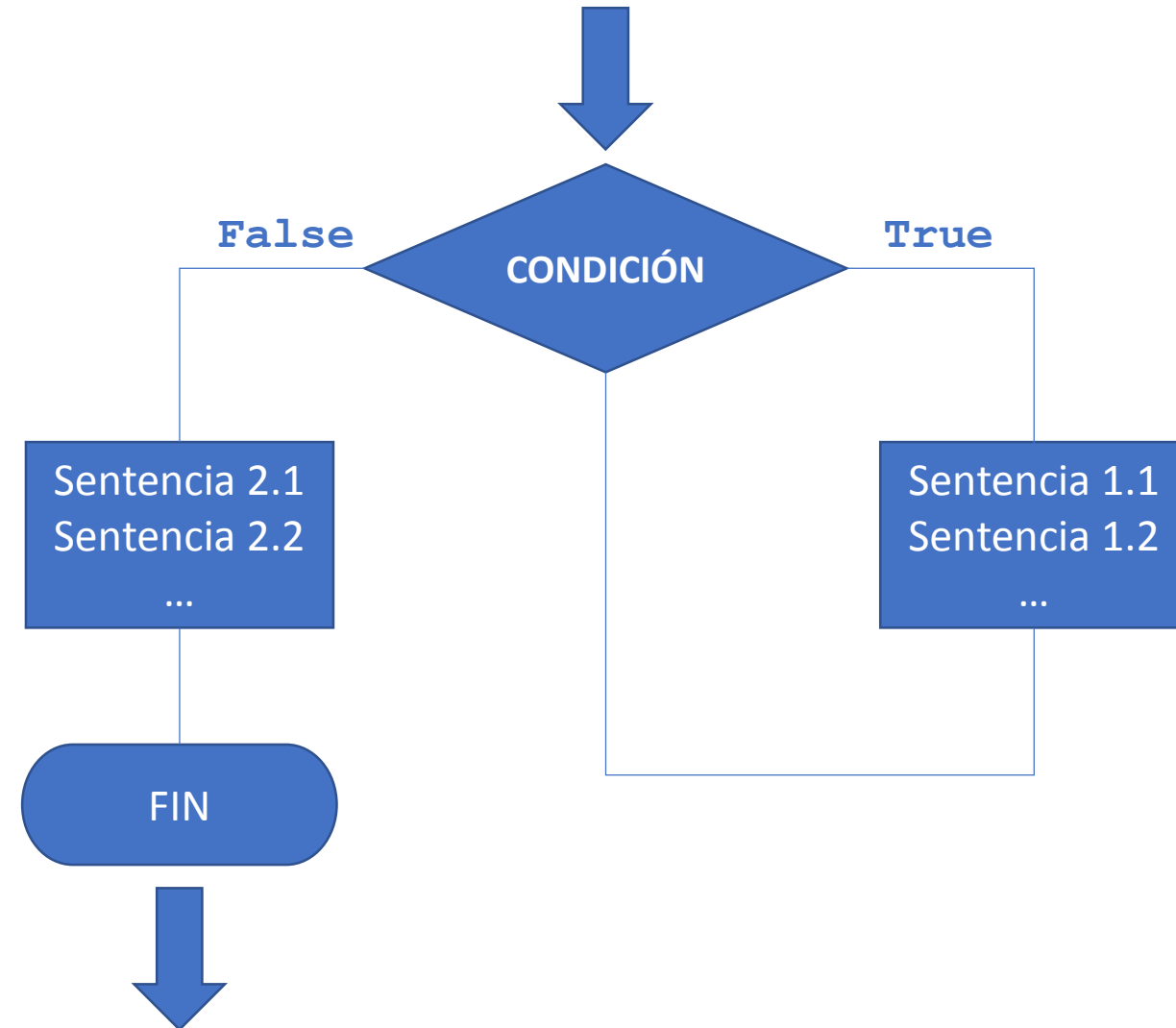
```
# Expresiones iterativas  
  
# Bucles tipo while  
print("Con palabras")  
palabra = "aeropuerto"  
letra = 0  
while letra < len(palabra):  
    print(palabra[letra])  
    letra += 1
```

Con palabras
a
e
r
o
p
u
e
r
t
o

- SENTENCIAS ITERATIVAS (BUCLE *WHILE*)

```
➤ while <condición>:  
    <sentencia 1.1>  
    <sentencia 1.2>  
    ...  
else:  
    <sentencia 2.1>  
    <sentencia 2.2>  
    ...
```

- La parte **else** es ejecutada si la condición en el bucle *while* la evalúa como **False**



- SENTENCIAS ITERATIVAS (BUCLE *WHILE*)

```
➤ while <condición>:  
    <sentencia 1.1>  
    <sentencia 1.2>  
    ...  
else:  
    <sentencia 2.1>  
    <sentencia 2.2>  
    ...
```

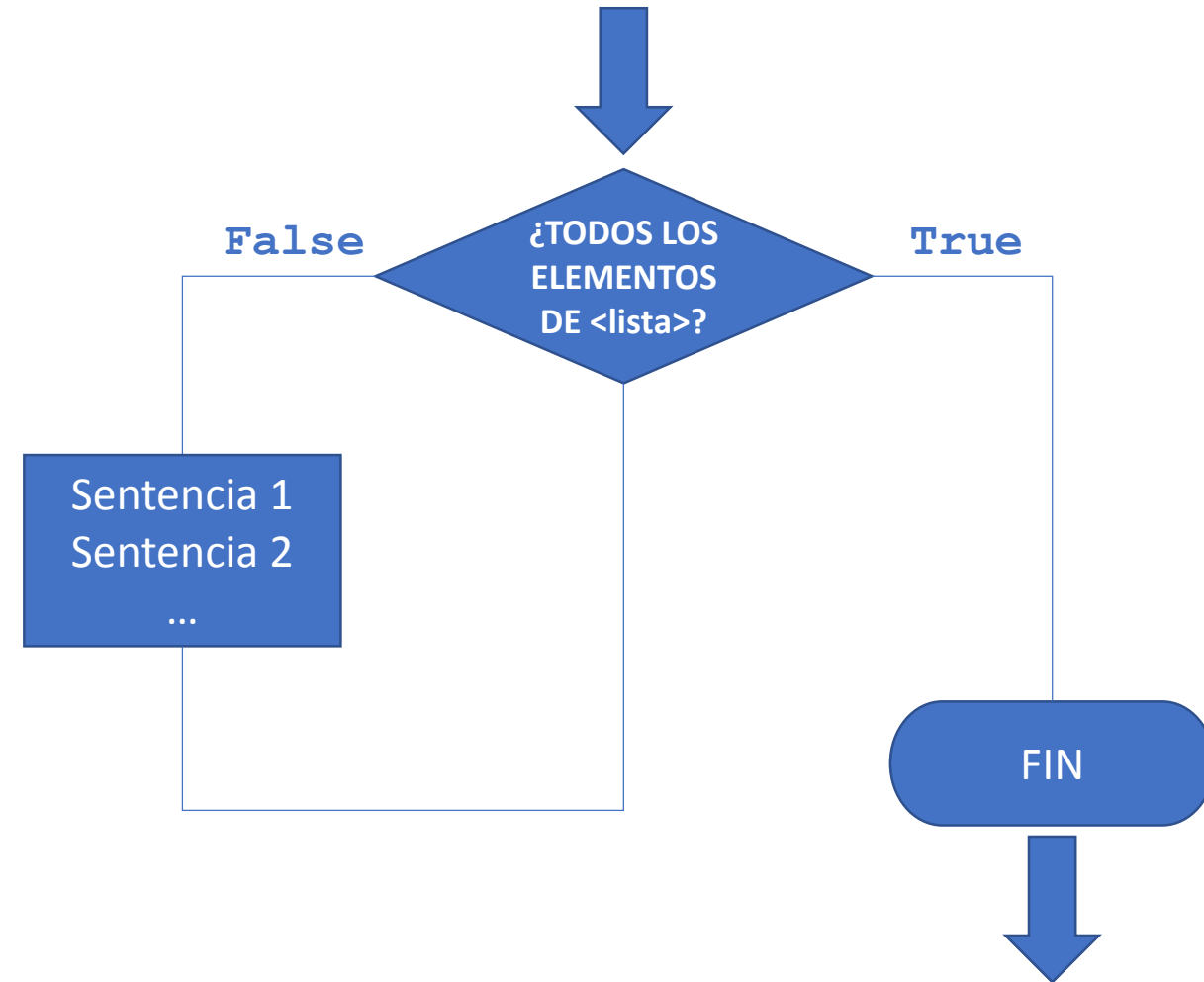
```
# Expresiones iterativas  
  
# Bucles tipo while con else  
print("Dos condiciones con operadores lógicos")  
numero = 1  
while (numero < 10 or numero%2 == 0):  
    print(numero)  
    numero += 1  
else:  
    print(numero, "es un impar mayor de 9")
```

```
Dos condiciones con operadores lógicos  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11 es un impar mayor de 9
```

- SENTENCIAS ITERATIVAS (BUCLE *FOR*)

```
➤ for <variable> in <lista>:  
    <sentencia 1>  
    <sentencia 2>  
    ...
```

- El programa vincula la variable al primer elemento de la lista y se ejecuta las sentencias. A continuación la variable pasa al segundo elemento y se vuelve a ejecutar.
- El bucle continua hasta que ha completado todos los elementos de la lista.
- El sangrado, de 4 espacios, es automático cada vez que usamos :



- SENTENCIAS ITERATIVAS (BUCLE *FOR*)

➤ `for <variable> in <lista>:`
 `<sentencia 1>`
 `<sentencia 2>`
 ...

```
# Expresiones iterativas

# Bucles tipo for
numeros = [6, 5, 3, 8, 4, 2, 5, 4, 11]
suma = 0
for valor in numeros:
    suma = suma + valor
print("La suma es", suma)
```

La suma es 48

```
# Expresiones iterativas

# Bucles tipo for
genero = ['pop', 'rock', 'jazz']
for i in range(len(genero)):
    print("A mi me gusta el", genero[i])
```

A mi me gusta el pop
A mi me gusta el rock
A mi me gusta el jazz

- SENTENCIAS ITERATIVAS (BUCLE *FOR*)

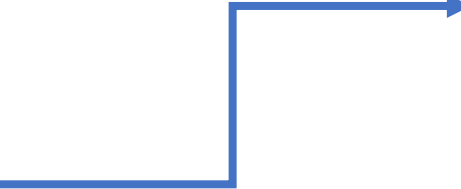
➤ Ejemplo de **range**

```
print("Bucle for con valor final")  
for i in range(10):  
    print(i)
```

```
print("Bucle for con valor inicial-final")  
for i in range(5,10):  
    print(i)
```

```
print("Bucle for con valor inicial-final-salto")  
for i in range(1,10,2):  
    print(i)
```

```
print("Bucle for con valor inicial-final-salto(negativo)")  
for i in range(10,1,-1):  
    print(i)
```



```
Bucle for con valor final  
0  
1  
2  
3  
4  
5  
6  
7  
8  
9
```

- SENTENCIAS ITERATIVAS (BUCLE *FOR*)

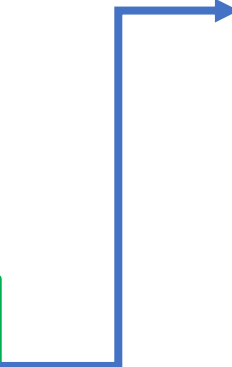
➤ Ejemplo de **range**

```
print("Bucle for con valor final")
for i in range(10):
    print(i)
```

```
print("Bucle for con valor inicial-final")
for i in range(5,10):
    print(i)
```

```
print("Bucle for con valor inicial-final-salto")
for i in range(1,10,2):
    print(i)
```

```
print("Bucle for con valor inicial-final-salto(negativo)")
for i in range(10,1,-1):
    print(i)
```



```
Bucle for con valor inicial-final
5
6
7
8
9
```

- SENTENCIAS ITERATIVAS (BUCLE *FOR*)

➤ Ejemplo de **range**

```
print("Bucle for con valor final")
for i in range(10):
    print(i)
```

```
print("Bucle for con valor inicial-final")
for i in range(5,10):
    print(i)
```

```
print("Bucle for con valor inicial-final-salto")
for i in range(1,10,2):
    print(i)
```

```
print("Bucle for con valor inicial-final-salto(negativo)")
for i in range(10,1,-1):
    print(i)
```

```
Bucle for con valor inicial-final-salto
1
3
5
7
9
```

- SENTENCIAS ITERATIVAS (BUCLE *FOR*)

➤ Ejemplo de **range**

```
print("Bucle for con valor final")
for i in range(10):
    print(i)
```


```
print("Bucle for con valor inicial-final")
for i in range(5,10):
    print(i)
```

```
print("Bucle for con valor inicial-final-salto")
for i in range(1,10,2):
    print(i)
```

```
print("Bucle for con valor inicial-final-salto(negativo)")
for i in range(10,1,-1):
    print(i)
```

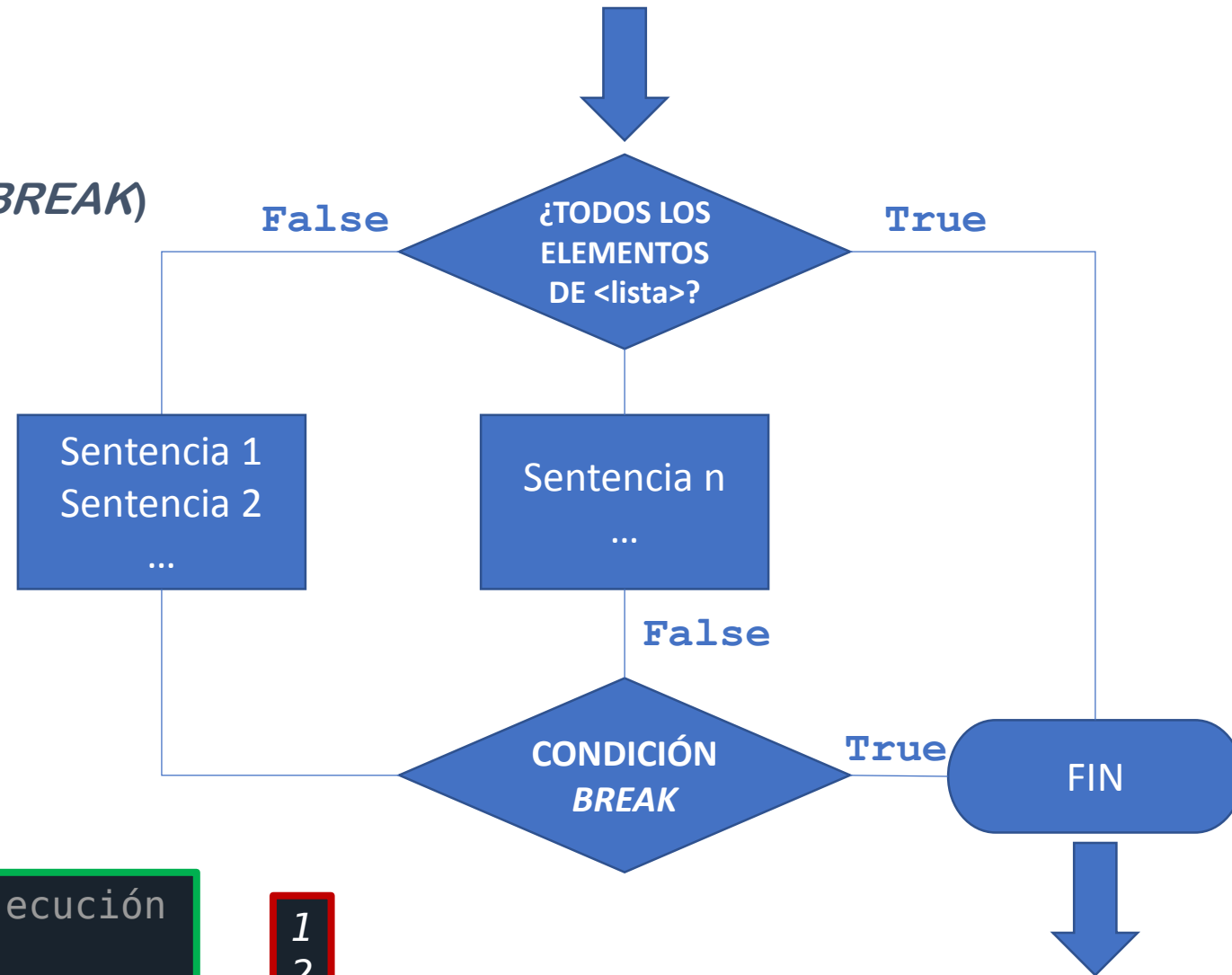
*Bucle for regresivo con valor
final-inicial-salto(negativo)*

```
10
9
8
7
6
5
4
3
2
```



- SENTENCIAS ITERATIVAS (BUCLE *FOR*+*BREAK*)

```
➤ for <variable> in <lista>:  
    <sentencia 1>  
    <sentencia 2>  
    ...  
    if <condición>:  
        break  
    <sentencia n>  
    ...
```

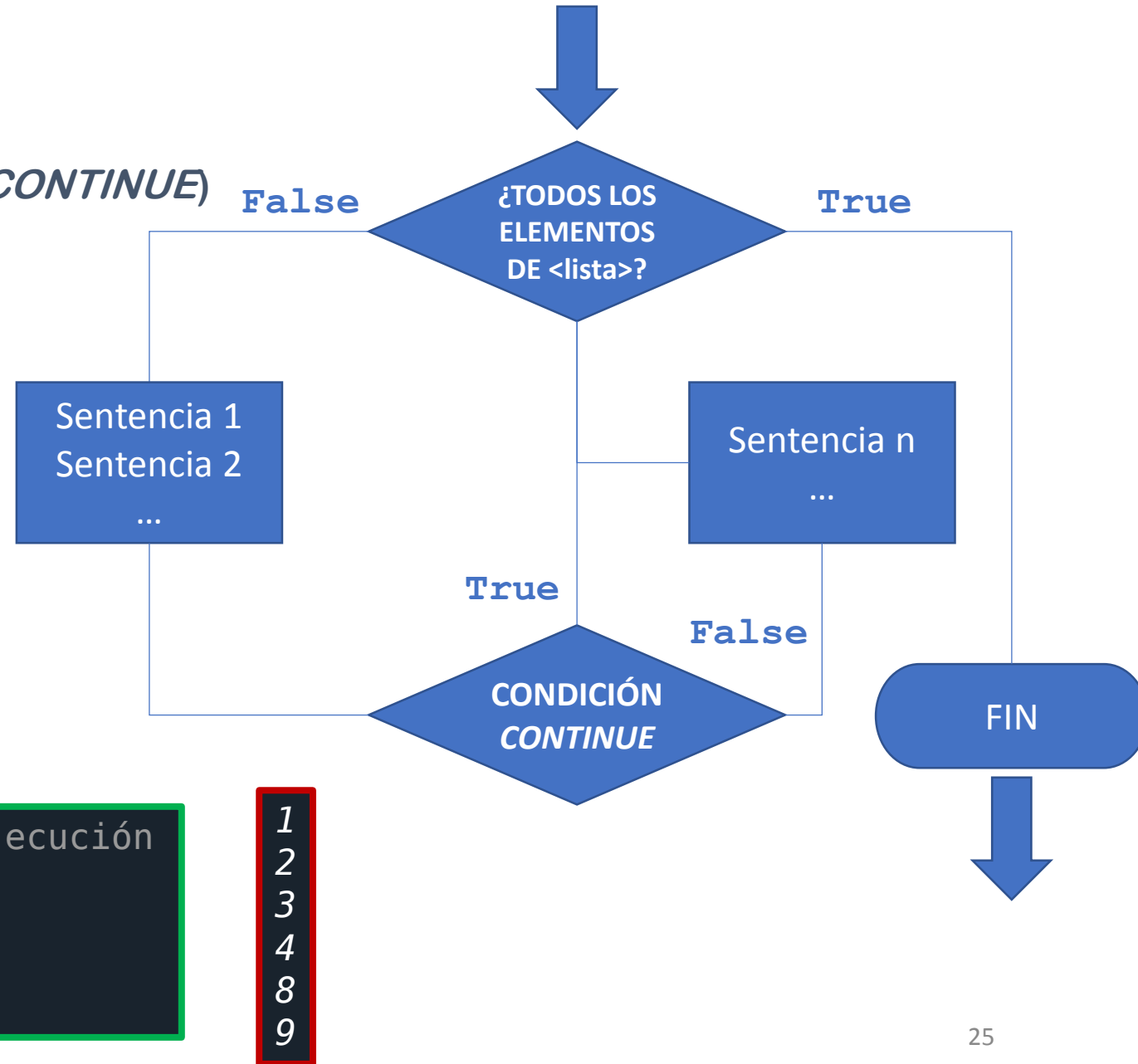


```
# Bucle for con sentencias de control de ejecución  
for i in range(1,10):  
    if (i == 5):  
        break  
    print(i)
```

1
2
3
4

- SENTENCIAS ITERATIVAS (BUCLE *FOR*+*CONTINUE*)

```
➤ for <variable> in <lista>:
    <sentencia 1>
    <sentencia 2>
    ...
    if <condición>:
        continue
    <sentencia n>
    ...
```



```
# Bucle for con sentencias de control de ejecución
for i in range(1,10):
    if (i >= 5 and i < 8):
        continue
    print(i)
```

1
2
3
4
8
9

- SENTENCIAS ITERATIVAS (BUCLE *FOR* COMO BUCLE *WHILE*)

```
for <variable> in <lista>:  
    <sentencia 1>  
    <sentencia 2>  
    ...
```

```
indice = 0  
while indice < len(<lista>):  
    <variable> = <lista>[indice]  
    <sentencia 1>  
    <sentencia 2>  
    ...  
    indice += 1
```

EJERCICIOS PROPUESTOS

1. Escribir un programa que pida (por teclado) un número natural y muestre por pantalla si se trata de un número primo o no.
2. Escribir un programa que pida (por teclado) un número natural y muestre por pantalla el número de dígitos que contiene.
3. Escribir un programa que pida (por teclado) un número natural y muestre por pantalla el primer y el último dígito de este número.
4. Determine la suma finita de las siguientes sucesiones:

$$S_5 = \sum_{n=1}^5 \frac{2n-1}{n(n+1)} \quad S_{20} = \sum_{n=0}^{20} \sum_{k=0}^{n+1} \frac{1}{k!}$$

5. Determine la suma infinita de las siguientes sucesiones:

$$S = \sum_{n=0}^{\infty} \frac{1}{n!} \quad S = \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{2^n}$$

NOTA: para salir del bucle deberá cumplirse que el termino de la suma sea menor que precisión de maquina