

Introducción a la algoritmia



Lic. Julia Monasterio



Clase N°4

TEMAS

- Estructura Iterativa
- Instrucción While
- Contador
- Acumulador
-



¿Cómo lo resolvemos?

Con los conocimientos actuales como resolverías lo siguiente:

**Mostrar por pantalla los números del
1 al 5**

Y ahora...

Como imprimirías del 1 al 100

¿Que piensan de la estrategia de
solución?

100

Estructuras Iterativas

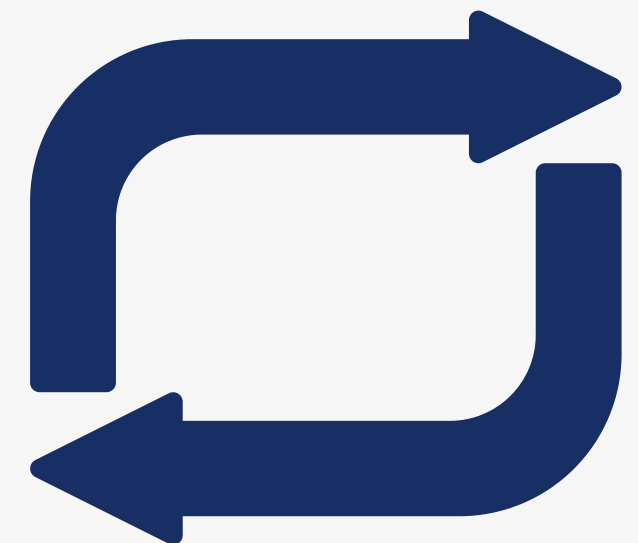
Estructura Iterativa

Las estructuras que **repiten** una secuencia de instrucciones un **número determinado de veces** se denominan **bucles** y se denomina **iteración** al hecho de repetir la ejecución de una secuencia de acciones



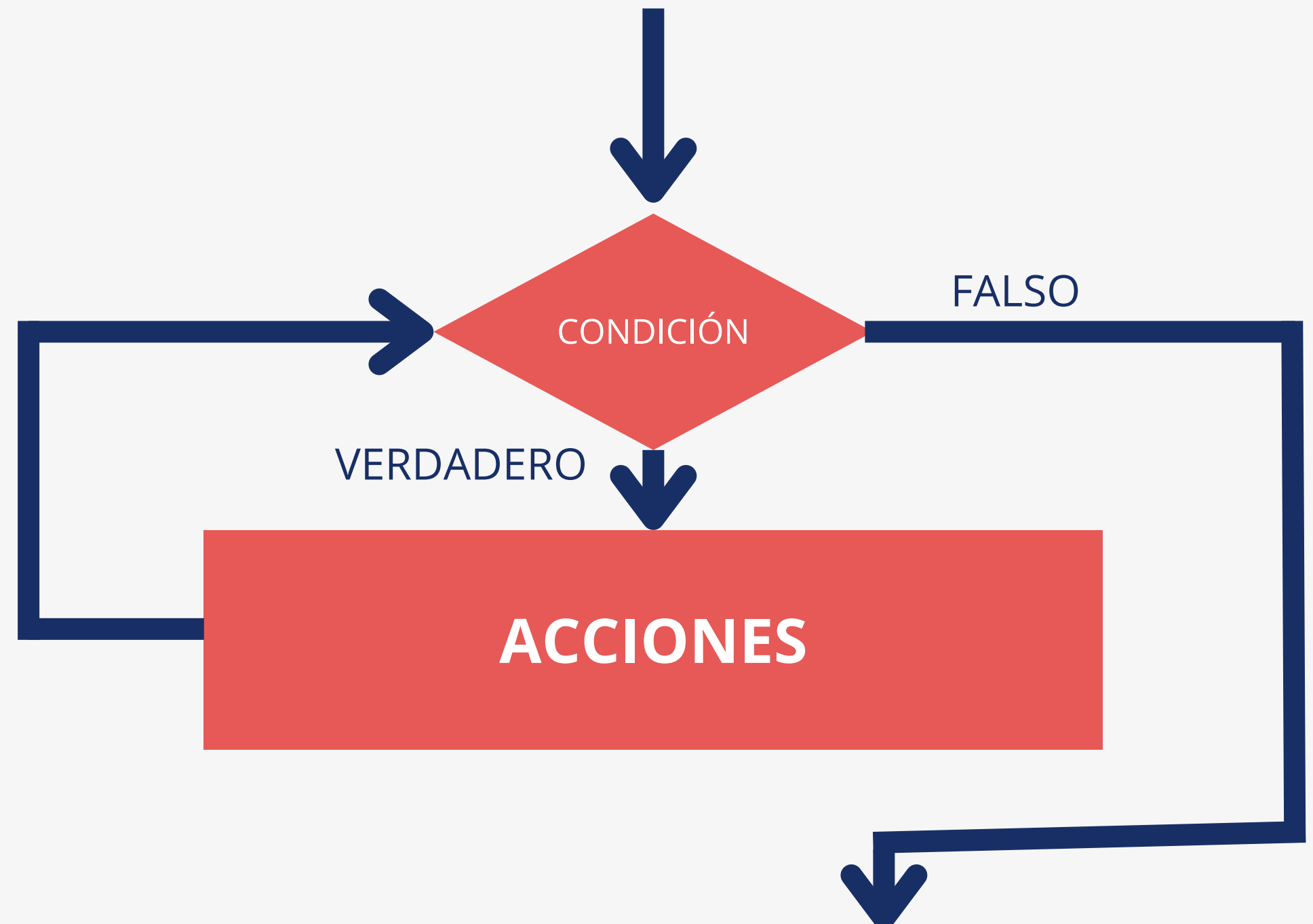
Estructura Iterativa

En esta estructura el programa **repetirá una porción de su código una cierta cantidad de veces, y luego seguirá adelante.**



WHILE (Mientras)

Diagrama de Flujo



WHILE (Mientras)

En Código

while <condicion>:

.....

.....

.....

WHILE

- Cuando se ejecuta la instrucción **while** sucede lo siguiente:
 1. Se evalúa la condición (una expresión booleana).
 - a. Si se evalúa falsa, no se toma ninguna acción y el programa continua en la siguiente instrucción del bucle.
 - b. Si la expresión booleana es verdadera, entonces se ejecuta el cuerpo del bucle, después de lo cual se evalúa de nuevo la expresión booleana.
 1. Este proceso se repite una y otra vez mientras la expresión booleana (condición) sea verdadera.

Reglas de sintaxis

- La condición va seguida de dos puntos
- La indentación es lo que establece el alcance del while



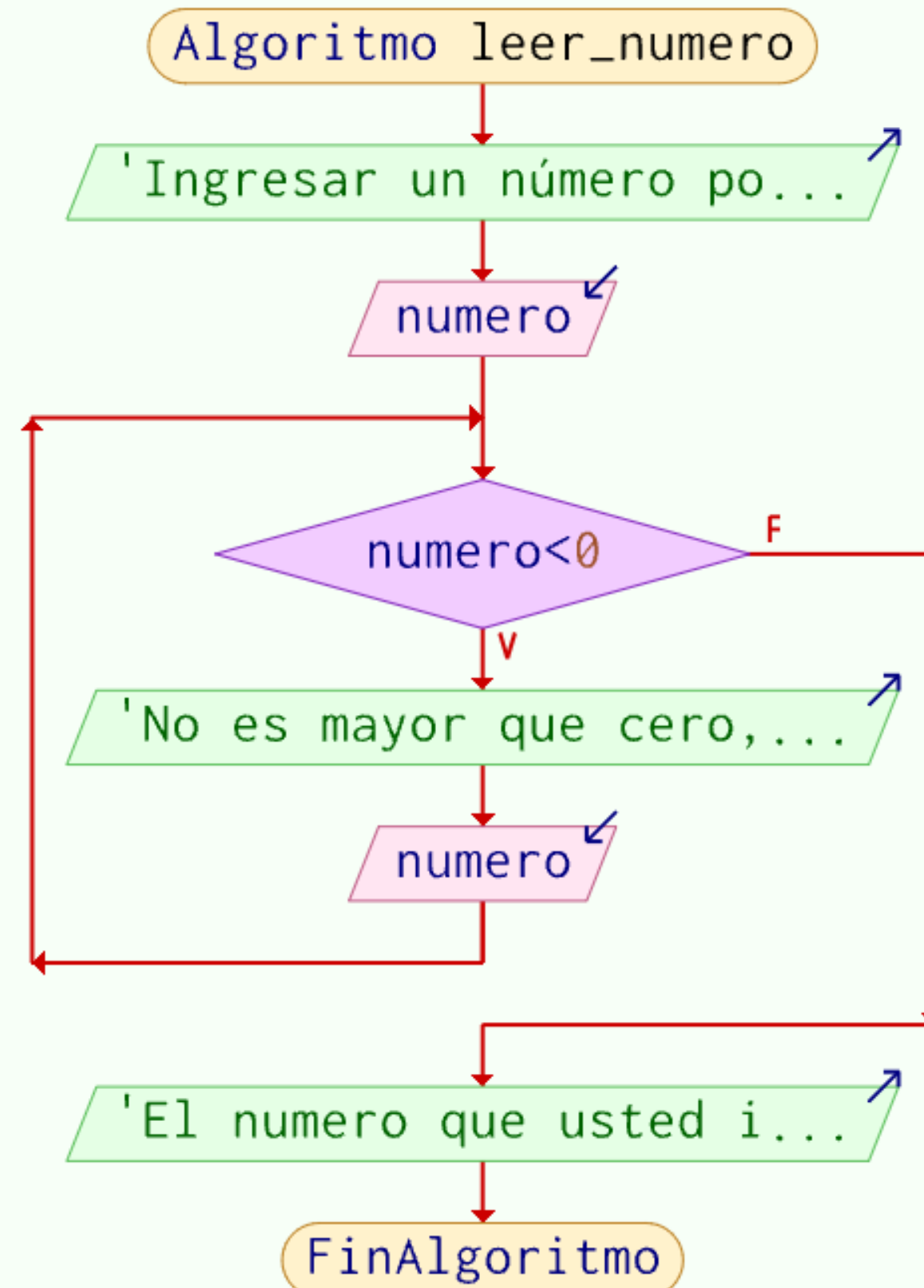
Práctica en clase

Realizar en PSEINT y en Python

Leer desde el teclado un número entero, asegurándose que sea positivo.



PSEINT



Código Python

```
numero= int(input("Ingrese un numero"))

while numero<0:
    print("Número incorrecto. Reingrese")
    numero=int(input("Ingrese un número correcto\n"))

print("El numero ingresado es: ",numero)
```

CONTADOR

Cuando una variable es modificada en una cantidad fija respecto de su valor anterior, se la denomina **contador**.

Práctica en clase

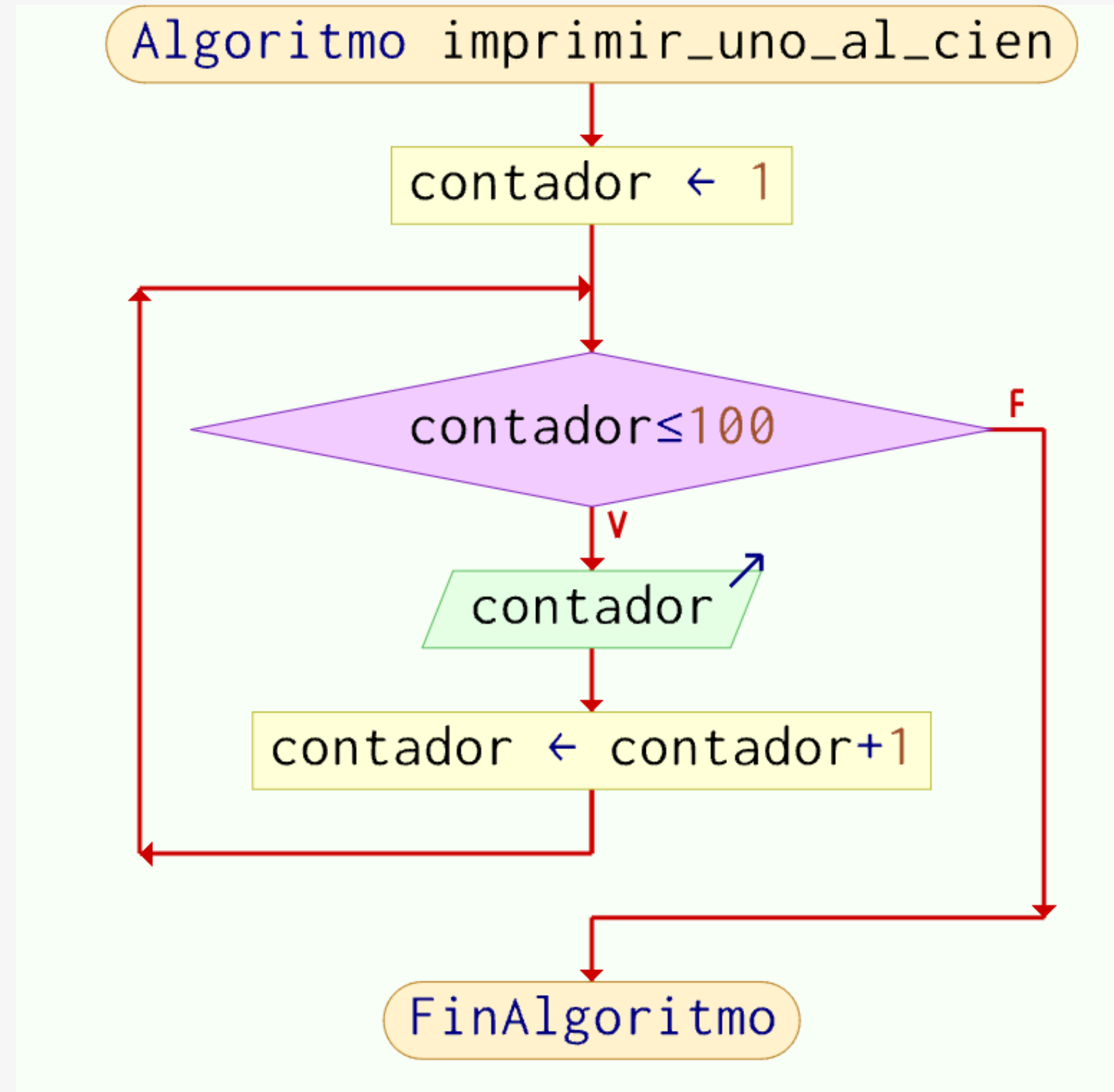
Realizar en PSEINT y en Python

Imprimir los números enteros del 1 al
100

**¿Qué datos debemos solicitarle al
usuario?**



PSEINT



Código Python

```
contador=1

while contador<=100:
    print(contador)
    contador+=1
```

Eso se conoce como **CICLO INFINITO**

Es uno de los errores más comunes y de los peores que puede cometer un programador



**Sucede lo mismo si la condición
nunca devuelve false**

```
a=1
```

```
while a>1:
```

```
    print(a)
```

```
    a+=1
```



**La condición siempre
esta devolviendo True,
es decir siempre es
positiva**

EJECUCIÓN DE BUCLE CERO VECES

En una estructura **mientras o while** lo primero que sucede es la **evaluación de la expresión booleana**; si se evalúa falsa en ese punto, entonces del **cuerpo del bucle nunca se ejecuta**.

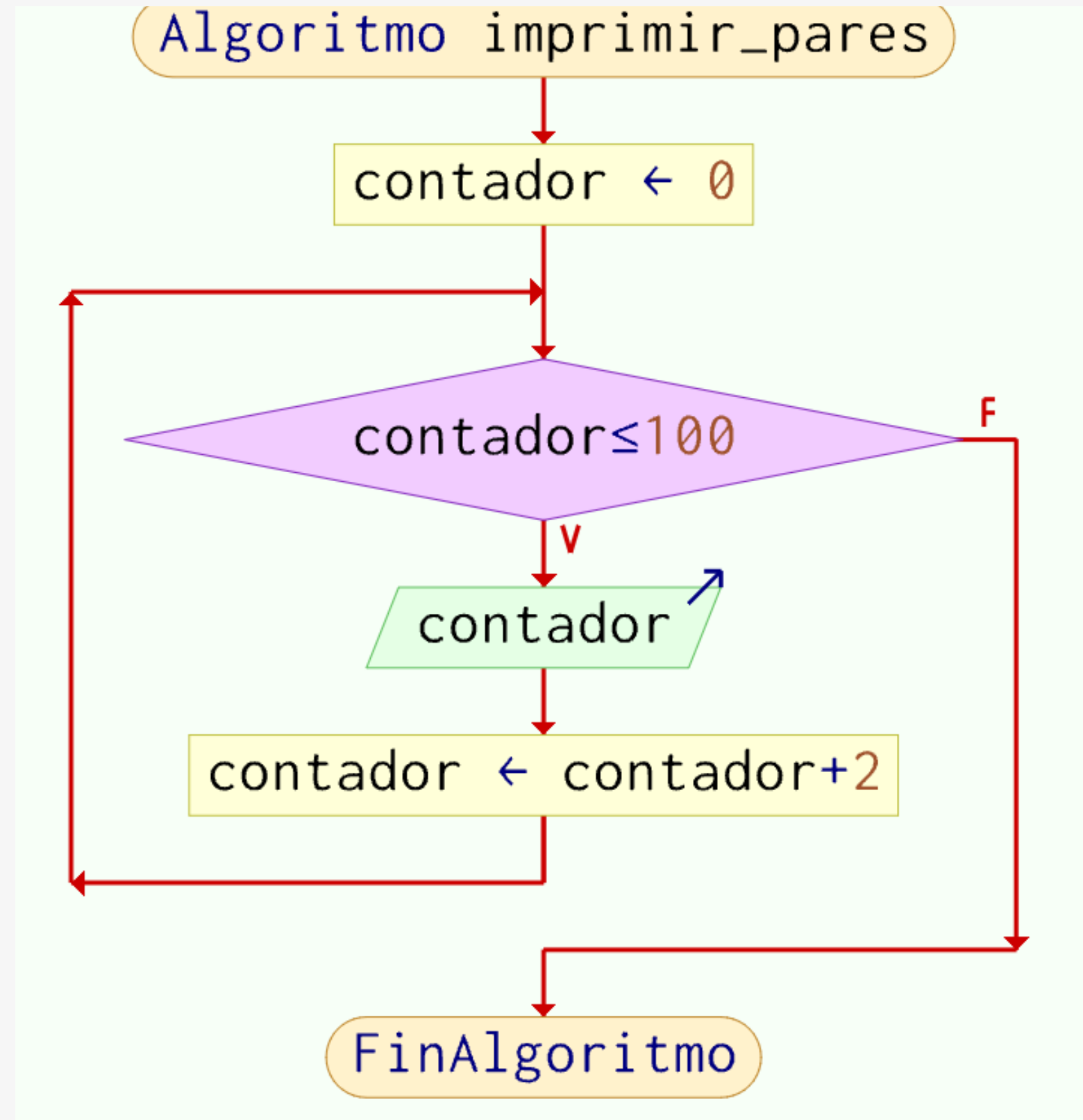
Práctica en clase

Realizar en PSEINT y en Python

Imprimir los números pares del 0 al
100



PSEINT - Solución 1



PSEUDO CÓDIGO - SOLUCIÓN 1

```
Algoritmo imprimir_pares
```

```
    contador ← 0;
```

```
    Mientras contador ≤ 100 Hacer
```

```
        Escribir contador;
```

```
        contador ← contador + 2;
```

```
    Fin Mientras
```

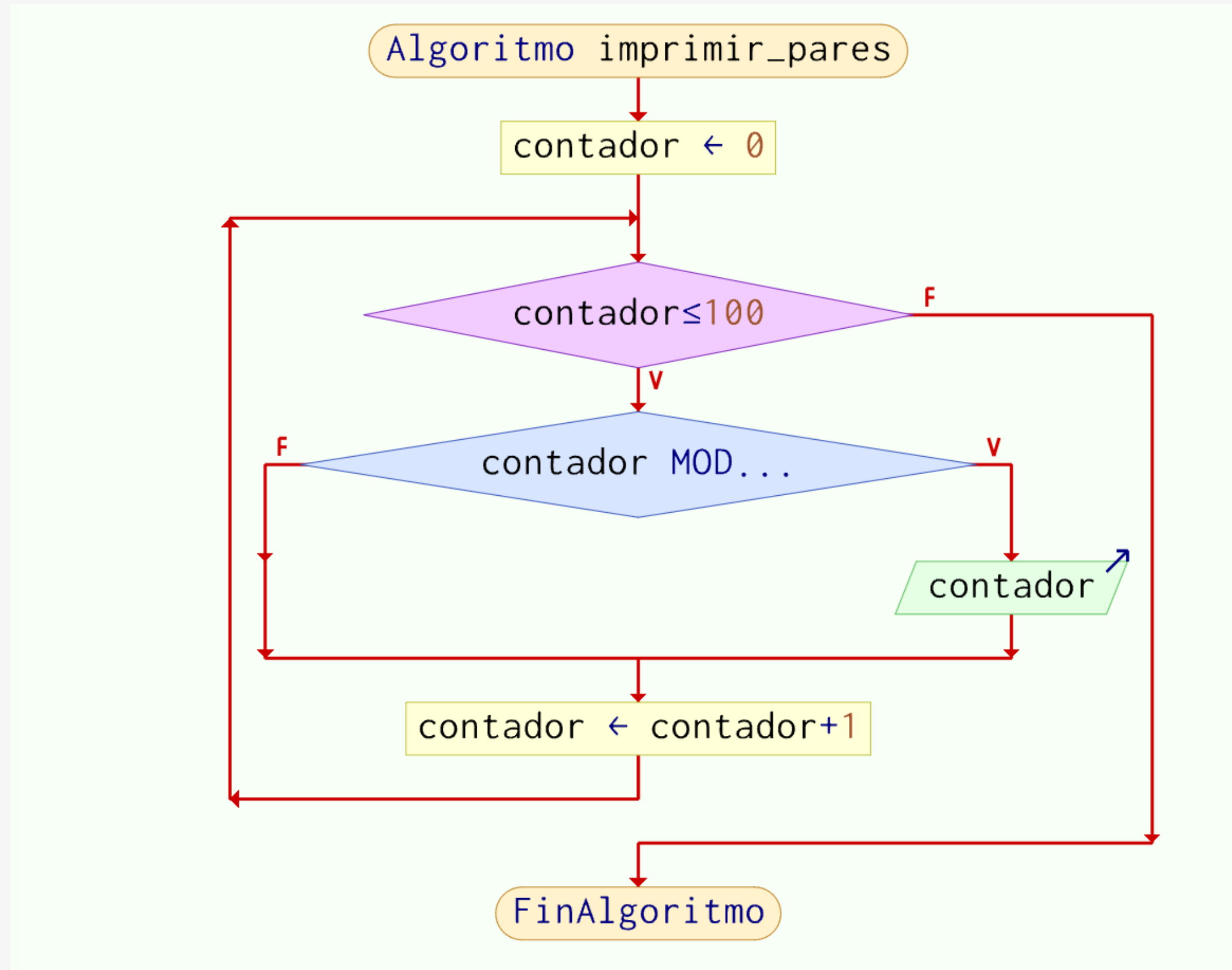
```
FinAlgoritmo
```


Código Python - Solución 1

```
contador=0

while contador<=100:
    print(contador)
    contador+=2
```

PSEINT - Solución 2



PSEUDO CÓDIGO - SOLUCIÓN 2

```

Algoritmo imprimir_pares
    contador ← 0;

    Mientras contador ≤ 100 Hacer
        Si contador % 2 == 0 Entonces
            Escribir contador;
        Fin Si
        contador ← contador + 1;
    Fin Mientras
FinAlgoritmo
    
```

Código Python - Solución 1

```
contador=0

while contador<=100:
    if contador%2==0:
        print(contador)
    contador+=2
```

ACUMULADOR

Cuando una variable es modificada en una cantidad cambiante respecto de su valor anterior, se la denomina **acumulador**.

Práctica en clase

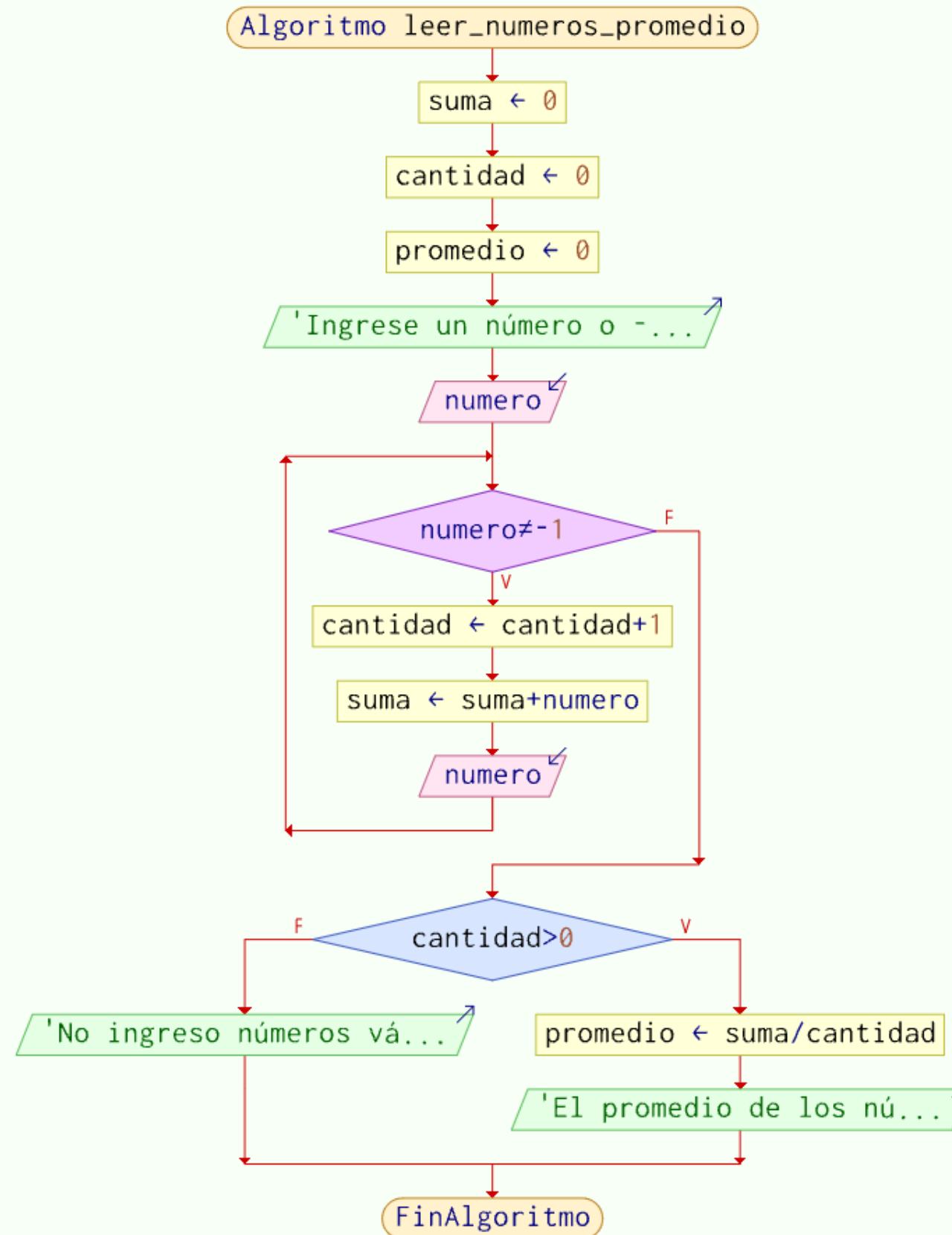
Realizar en PSEINT y en Python

Leer un conjunto de números enteros e imprimir su promedio. El fin de los datos se indica ingresando el valor 1.

¿Qué datos debemos hacer con los datos ingresados?



PSEINT - Solución



PSEUDO CÓDIGO - SOLUCIÓN

```

Algoritmo leer_numeros_promedio
    suma←0;
    cantidad←0;
    promedio←0;

    Escribir "Ingrese un número o -1 para terminar";
    Leer numero;

    Mientras numero≠-1 Hacer
        cantidad←cantidad+1;
        suma←suma+numero;
        Leer numero;
    Fin Mientras

    Si cantidad>0 Entonces
        promedio←suma/cantidad;
        Escribir "El promedio de los números es: ",promedio;
    SiNo
        Escribir "No ingreso números válidos";
    Fin Si
FinAlgoritmo
    
```


Código Python - Solución

```
suma=0
cantidad=0
promedio=0

numero=int(input("Ingrese un número o -1 para terminar\n"))

while numero!=-1:
    cantidad+=1
    suma+=numero
    numero=int(input("Ingrese otro número\n"))

if cantidad>0:
    promedio=suma/cantidad
    print("El promedio es: ",promedio)
else:
    print("No ingreso valores válidos")
```

Aspectos importantes

- Es importante evitar **dividir por cero**
- Los promedios siempre se deben informar con decimales

BANDERA

- Una variable de tipo booleana o bandera, es una variable que solo puede tener dos posibles valores: **verdadero o falso**.
- Estas variables son útiles para representar estados binarios o para controlar la ejecución de un programa mediante una condición simple.
- En Python, las variables de tipo bandera se definen utilizando los valores **True o False**, que son palabras clave del lenguaje

Aspectos importantes

- La mayoría de las veces, tienen 2 estados, por eso se las conoce como booleanas.
- El concepto de bandera significa que la variable solo puede tomar un determinado **tipo de valor fijo**, como mínimo 2. Lo que el programa necesite.
- Estos valores fijos se los denomina **ESTADOS**, por lo que, en tiempo de ejecución, el programa toma decisiones en función de estos valores predeterminados al momento de la codificación.

EJEMPLO 1

```
bandera=True
if bandera:
    #Este código se ejecutara si la bandera se encuentra en True
    print("La bandera esta activada")
else:
    #Este código se ejecutara si la bandera se encuentra en False
    print("La bandera no esta activada")
```

EJEMPLO 2

```
bandera=True
while bandera:
    #Código a ejecutar mientras la bandera sea verdadera
    respuesta = input("¿Desea continuar? (s/n): ")
    if respuesta == 'n':
        bandera = False
```

Práctica en clase

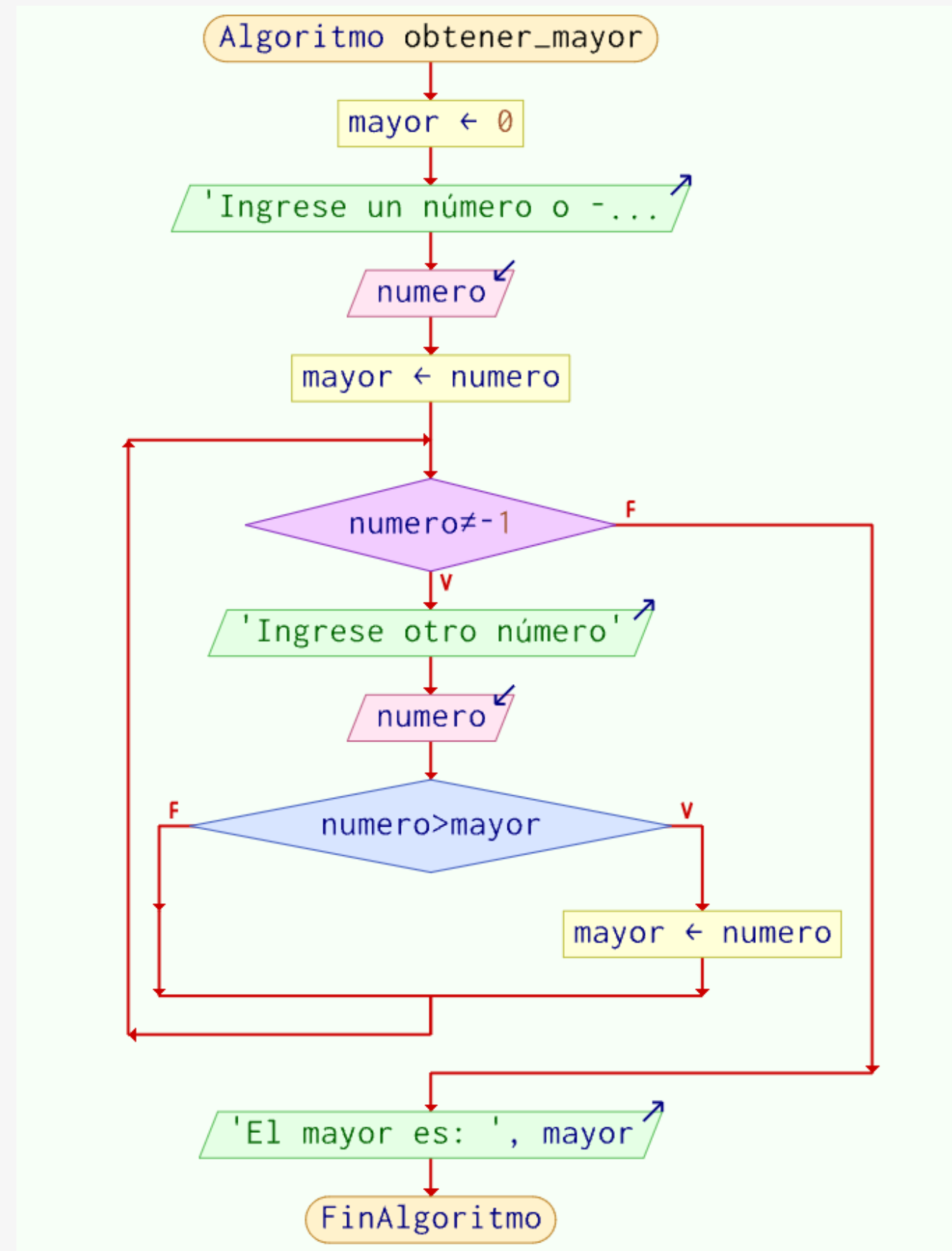
Realizar en PSEINT y en Python

Leer un conjunto de números enteros e
imprimir el mayor.

El fin de los datos se indica con -1.



PSEINT - Solución



PSEUDO CÓDIGO - SOLUCIÓN

Algoritmo obtener_mayor

```

mayor←0;
Escribir "Ingrese un número o -1 para terminar";
Leer numero;
mayor←numero;
Mientras numero≠-1 Hacer
    Escribir "Ingrese otro número";
    Leer numero;
    Si numero>mayor Entonces
        mayor←numero;
    Fin Si
Fin Mientras

Escribir "El mayor es: ",mayor;
FinAlgoritmo
    
```

Código Python - Solución

```
#Obtener el mayor

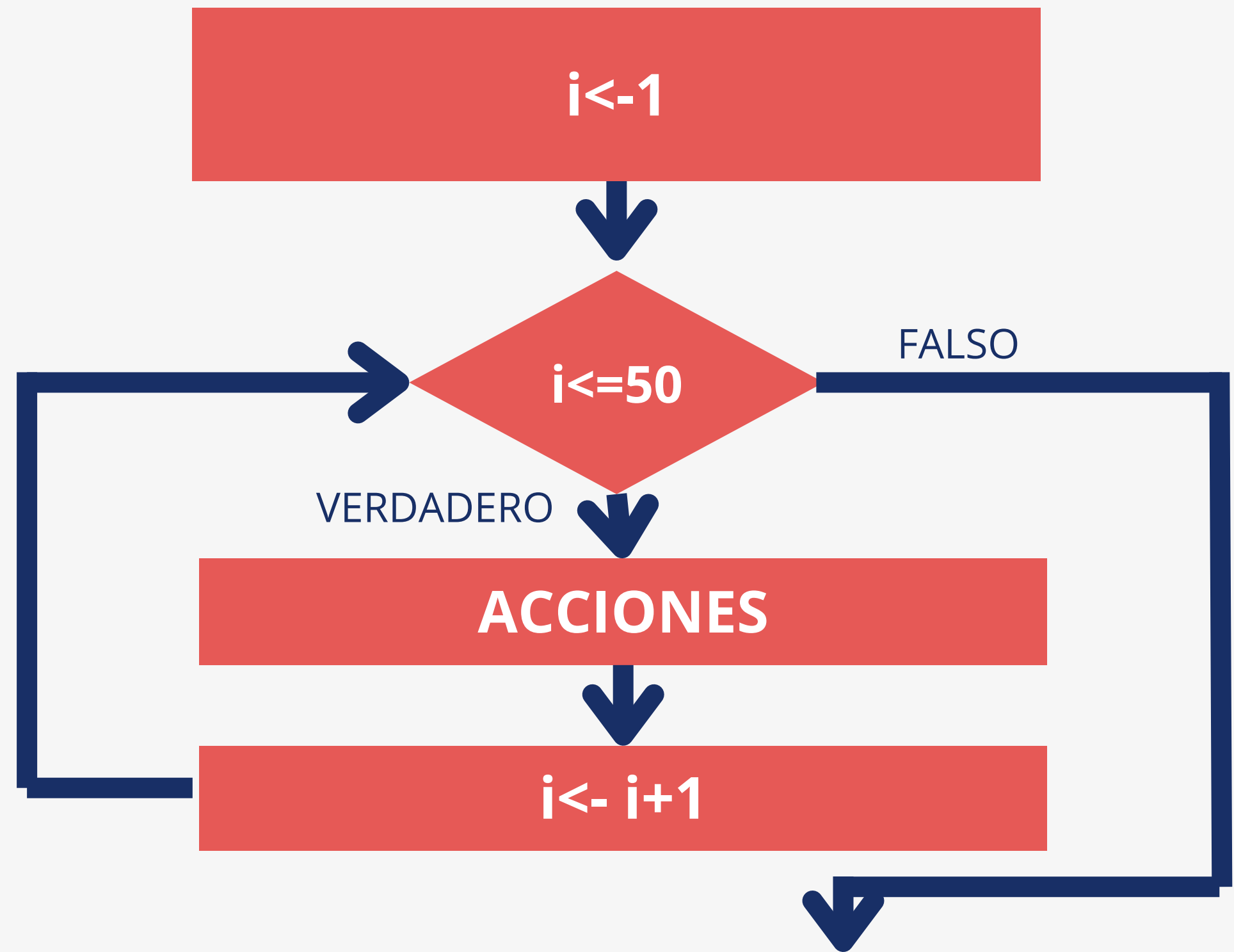
numeroEvaluar=int(input("Ingrese un número\n"))
mayor=numeroEvaluar

while numeroEvaluar!=-1:
    numeroEvaluar=int(input("Ingrese un número\n"))
    if numeroEvaluar>mayor:
        mayor=numeroEvaluar

print("El mayor es: ", mayor)
```

FOR (desde/para)

Diagrama de Flujo



FOR

En muchas ocasiones se conoce de **antemano el número de veces que se desean ejecutar las acciones de un bucle.**

En estos casos, en el que el número de iteraciones es fijo, se debe usar la estructura **desde o para (for, en inglés).**

for i in range (x,y):

.....

.....

.....

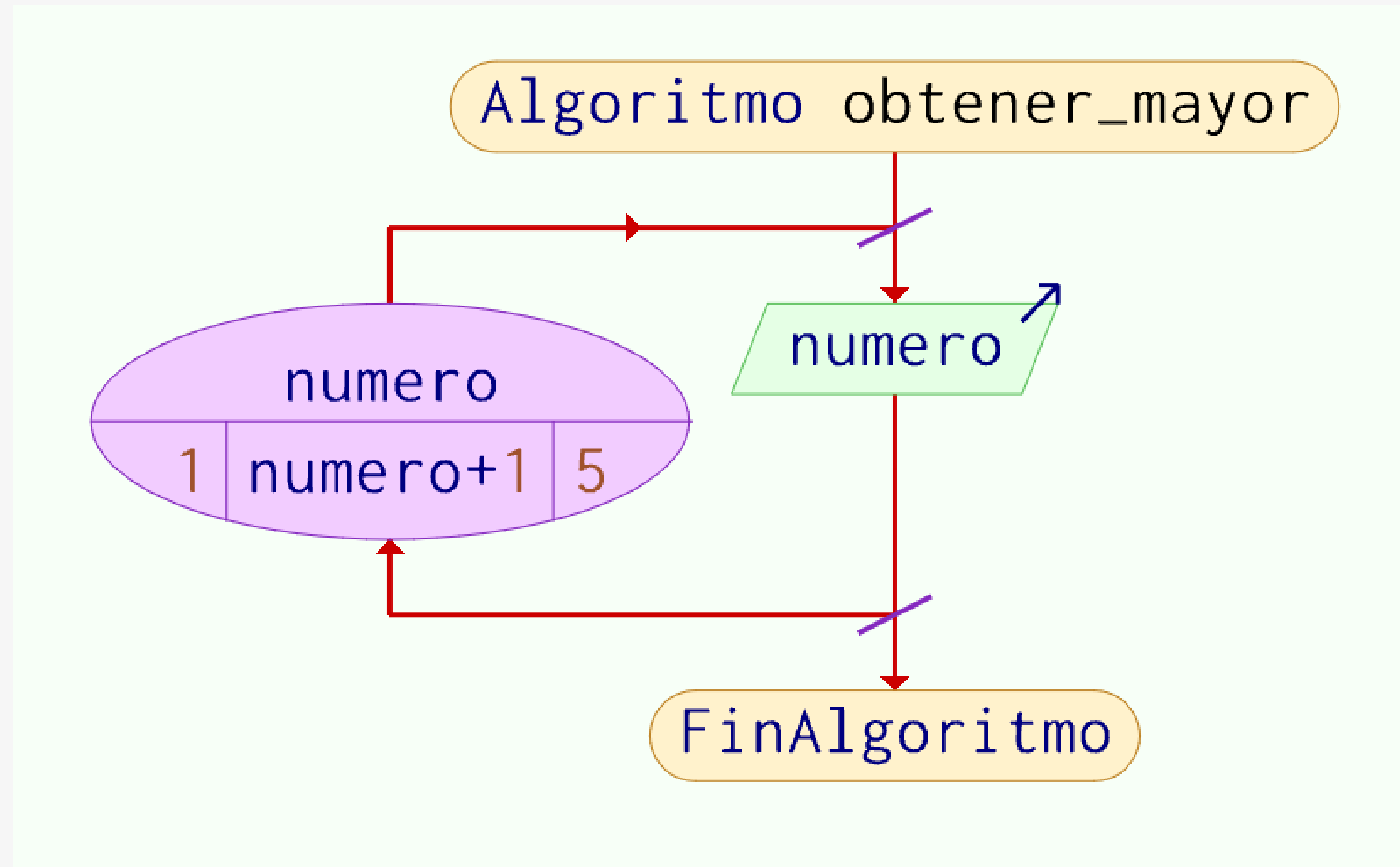
Práctica en clase

Realizar en PSEINT y en Python

Imprimir los números del 1 al 5



PSEINT - Solución



PSEUDO CÓDIGO - SOLUCIÓN

```

Algoritmo obtener_mayor
    Para numero<-1 Hasta 5 Con Paso numero+1 Hacer
        |   Escribir numero;
    Fin Para
FinAlgoritmo
    
```

Código Python - Solución

```
for i in range(1,6):  
    print(i)
```


Resumen de la clase

- Estructuras iterativas
- Instrucción while en Python
- Variable Contador
- Variable de tipo Acumulador
- Variable tipo Bandera
- Instrucción FOR en Python



EJERCITACIÓN

Objetivos

- Crear algoritmos en diagrama de flujo y codificar combinando estructuras secuencial, condicional e iterativa

Ejercitación

- **Ejercicio 1:** Realizar un programa para ingresar desde el teclado un conjunto de números. Al finalizar mostrar por pantalla el primer y último valor ingresado. Finalizar la lectura con -1.
- **Ejercicio 2:** Realizar un programa para ingresar desde el teclado un conjunto de números e informar si la cantidad de elementos es impar o par, sin utilizar contadores. Finalizar la lectura de datos con -1.
- **Ejercicio 3:** Realizar un programa para ingresar desde el teclado un conjunto de números y mostrar por pantalla el menor y el mayor de ellos. Finalizar la lectura de datos con un valor -1.



Ejercitación

- **Ejercicio 4:** Desarrollar un programa que imprima la suma de los números impares comprendidos entre 42 y 176.
- **Ejercicio 5:** Desarrollar un programa que imprima los números naturales comprendidos entre 1 y N. El valor de N se ingresa desde el teclado.
- **Ejercicio 6:** Mostrar la tabla de multiplicar (entre 1 y 12) del número 4. ¿Cómo cambiaría el algoritmo para que el usuario pueda decidir la tabla de multiplicar a mostrar?
- **Ejercicio 7:** Leer 10 números enteros e imprimir su promedio, el mayor valor leído y en qué posición se encontraba. Si se ingresó más de una vez sólo debe informar la primera.



Ejercitación

- **Ejercicio 8:** Ingresar números, hasta que la suma de los números pares supere 100. Mostrar cuántos números se ingresaron en total.
- **Ejercicio 9:** Se desea analizar cuántos autos circulan con patente con numeración par y cuántos con numeración impar en un día. Escribir un programa que permita ingresar la terminación de la patente (entre 0 y 9) hasta ingresar -1 e informe cuántos vehículos pasaron con numeración par y cuántos con numeración impar.
- **Ejercicio 10:** El factorial de un número entero N mayor que cero se define como el producto de todos los enteros X tales que $0 < X \leq N$. Desarrollar un programa para calcular el factorial de un número dado. Deberán rechazarse las entradas inválidas (menores que 0).



Ejercitación

- **Ejercicio 11:** Realizar un programa que lea un número natural H e imprima un mensaje indicando si H es primo o no. Se dice que un número es primo cuando sólo es divisible por sí mismo y por la unidad.
- **Ejercicio 12:** La sucesión de Fibonacci es una sucesión de números enteros donde cada término se obtiene como suma de los dos anteriores, siendo los dos primeros 0 y 1. Por lo tanto, $Fib=0, 1, 1, 2, 3, 5, 8, 13, 21, \dots$. Realizar un programa que lea N e imprima los N primeros términos de esta sucesión, como así también la suma de los mismos



Muchas gracias!

Consultas?

