**TAREA 4**

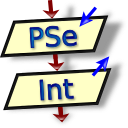
**Fecha de entrega: 21-08-2025**

**PRACTICA 2**

**RESOLUCIÓN DE ALGORITMOS UTILIZANDO ESTRUCTURAS DE CONTROL**

**Grupo N°4**

**Integrantes**

1. Harold Jimmy Arias Pérez
2. Juan Carlos Hidalgo Sosa
3. Juan Jose Rivas Ramirez
4. Luciano Franco Vargas Rojas
5. Nestor Fabian Loras Vega (Líder)

Cochabamba-Bolivia

#### FECHA DE ENTREGA: 21-08-2025

**PRACTICA 2**

1.- Crear un programa que escriba en pantalla los números del 1 al 20 que sean múltiplos de 3.

### E-P-S

#### ENTRADA:

1-Números del 1 al 20 Tipo entero

#### PROCESO:

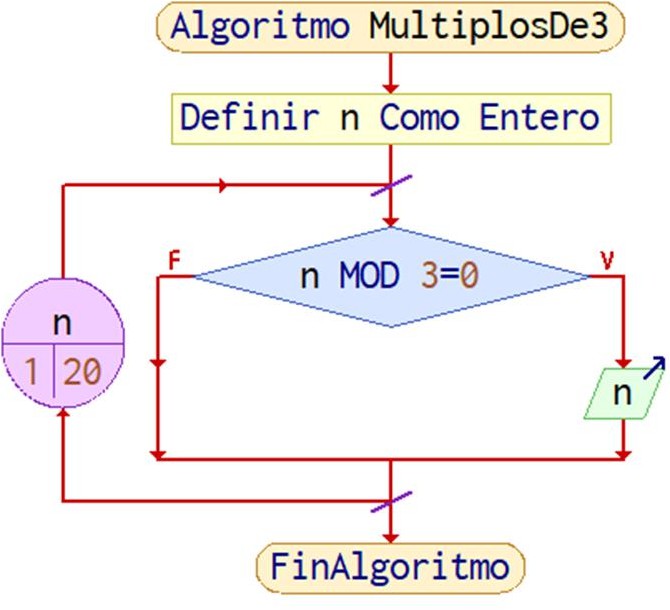
1-Recorrer los números del 1 al 20.

2-Verificar si el número es divisible entre 3 (n mod 3 = 0).

#### SALIDA:

1-Números entre 1 y 20 que son múltiplos de 3.

# **DIAGRAMA DE FLUJO**



2.- Realice un algoritmo para generar e imprimir los números que son primos que se encuentran entre 0 y 100

### E-P-S

#### ENTRADA:

1-Números del 0 al 100 tipo entero

#### PROCESO:

1-Recorrer los números del 2 al 100.

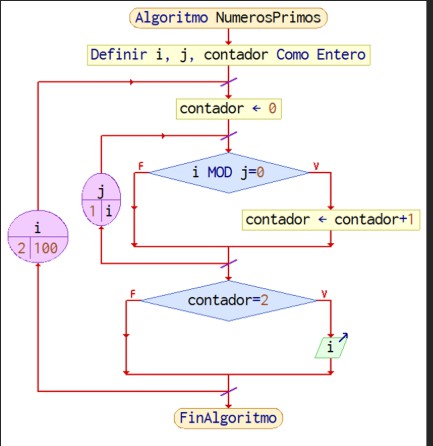
2-Para cada número, contar cuántos divisores tiene.

3-Si tiene exactamente 2 divisores(1 y él mismo), es primo.

#### SALIDA:

1-Lista de números primos entre 0 y 100.

# **DIAGRAMA DE FLUJO**



3.-Realizar un programa que permita ingresar un valor entero positivo por teclado y nos muestre la tabla de multiplicar de dicho valor.

### E-P-S

#### ENTRADA:

1-num: número positivo para hacer la tabla TIPO ENTERO

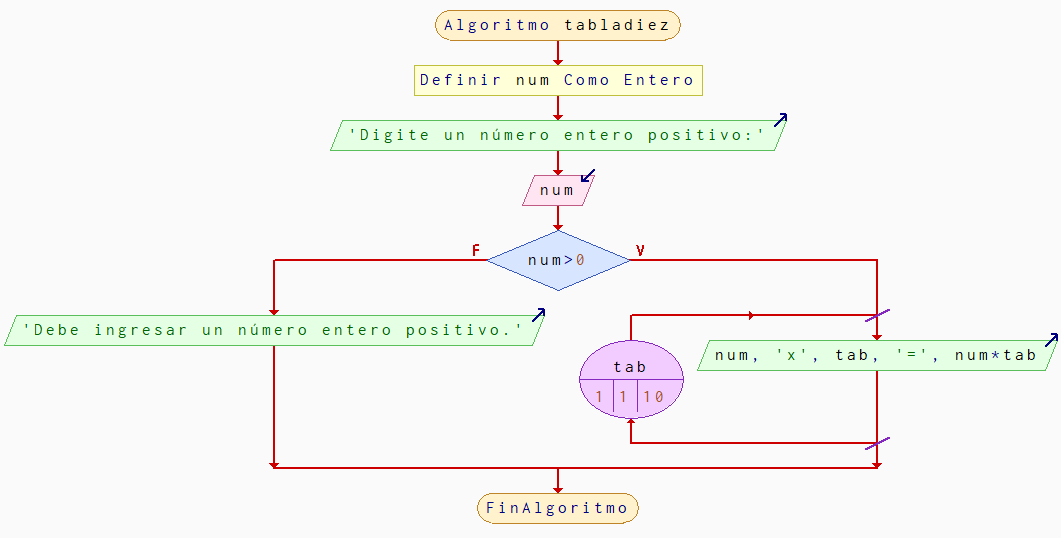
#### PROCESO:

1-Si “num” es mayor que 0 Repetir desde “tab” = 1 hasta 10 con paso 1 2-Si no escribir que debe ingresar un numero entero positivo

#### SALIDA:

3-Mostrar la tabla de “num” del 1 al 10

# **DIAGRAMA DE FLUJO**

****

4.-Realizar un programa que permita ingresar n números enteros por el teclado y se desea conocer:

1. La cantidad de números positivos.
2. La cantidad de números negativos.
3. La cantidad de números pares.
4. La cantidad de números impares

### E-P-S

#### ENTRADA:

**1-Cant:**la cantidad de números a procesar TIPO ENTERO

**2-n:**el numero para verificar si es positivo, negativo, par, o impar TIPO ENTERO

#### PROCESO:

1-Si el número es mayor a 0 se incrementa el contador de positivos, si es menor a 0 se incrementa el de negativos, si el número es divisible entre 2 se incrementa el contador de pares, de lo contrario se incrementa el de impares.

#### SALIDA:

1-Mostrar la cantidad total de números positivos, negativos, pares e impares

# **DIAGRAMA DE FLUJO**

5.- Realizar un algoritmo que pida números hasta que se introduzca un cero. debe imprimir la suma y el promedio de todos los números introducidos

### E-P-S

### ENTRADA

1-Números enteros introducidos por el usuario TIPO REAL

2-El número 0 como condición de finalización TIPO REAL

## PROCESO

1-Inicializar variables: suma = 0, contador = 0

2-Solicitar números repetidamente

3-Para cada número introducido:

4-Si el número es diferente de 0:

5-Sumar el número a la suma total

6-Incrementar el contador en 1 si el número es 0:

7-Calcular el promedio (suma / contador)

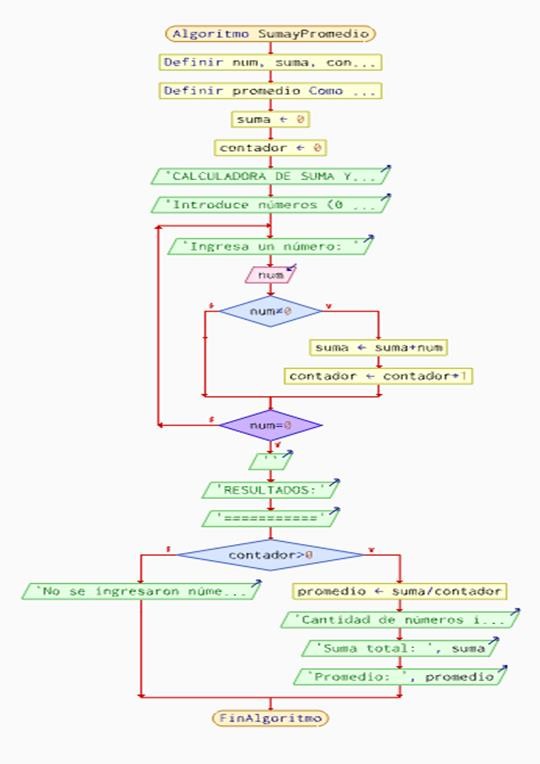
8-Terminar el ciclo

## SALIDA

1-La suma total de todos los números

2-El promedio de los números introducidos

**DIAGRAMA DE FLUJO**



6.-Un número entero es un “número perfecto” si sus factores, incluyendo al 1 (pero excluyendo en el número mismo), suman igual que el número. Ejemplo: 6 es un numero perfecto porque 6= 1+2+3. Escriba un programa que muestre los números perfecto de los primeros 50 números.

### E-P-S

## ENTRADA

1-Rango de números a evaluar: del 1 al 50 tipo entero   
2-No requiere entrada del usuario   
PROCESO

1-Para cada número del 1 al 50:

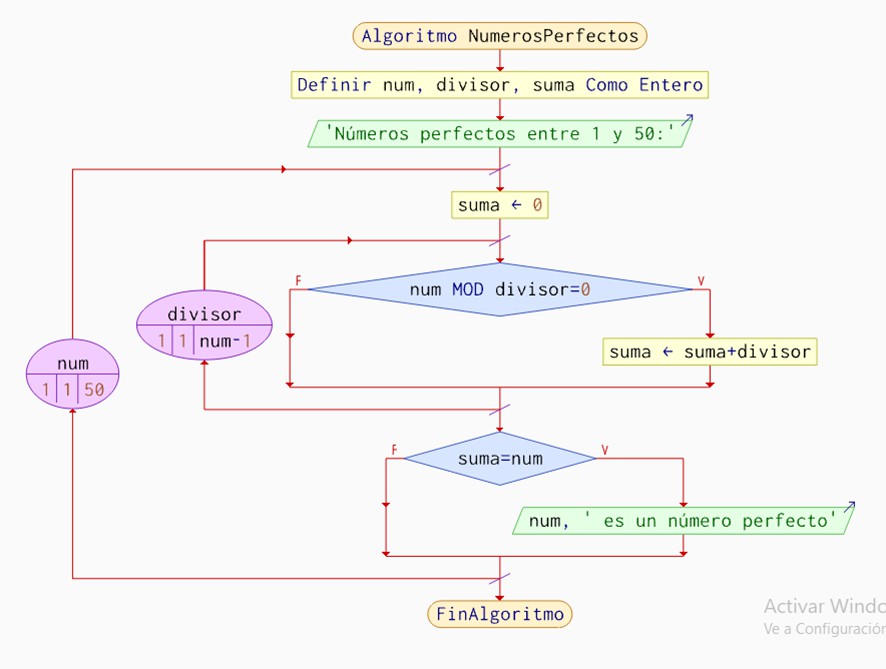
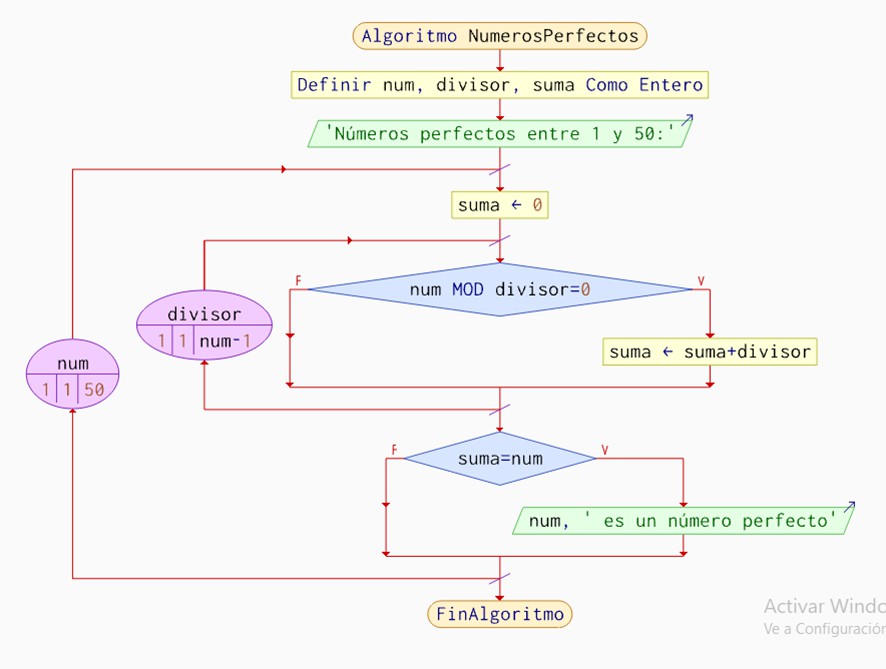
2-Inicializar suma de factores en O

3-Encontrar todos los factores del número (excluyendo el número MISMO)   
4-Sumarlos factores encontrados

5-Comparar si la suma de factores es igual al número original 6-Si son iguales, es un número perfecto

## SALIDA

Lista de números perfectos encontrados en el rango 1-50

**DIAGRAMA DE FLUJO**

7.- Considere siguiente proceso repetitivo para un número entero dado: si el número es 1, el proceso termina. De lo contrario, si es par se divide entre 2, y si es impar se multiplica por 3 y se le suma 1. Si empezamos con 6, por ejemplo, obtendremos la sucesión de números 6, 3, 10, 5, 16, 8, 4, 2, 1. La conjetura de Collatz dice que, a partir de cualquier número inicial, la sucesión obtenida siempre termina en 1. Escriba un programa que permita verificar la conjetura de Collatz para cualquier entero dado, y que imprima la secuencia correspondiente.

### E-P-S

## ENTRADA

## 1-n: el número para verificar si se cumple la conjetura de Collatz. De tipo entero

## PROCESO

1-Si n es uno. Se termina el proceso.

2-Si n es par, se divide entre 2.

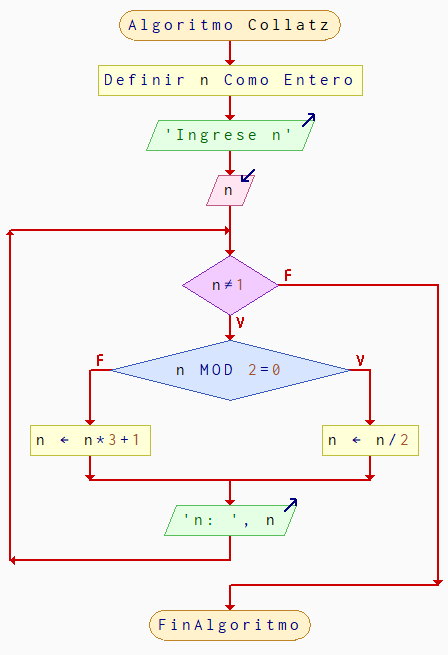
3-Si n es impar, se multiplica por 3 y se le suma 1.

## 4-Para cada n, escribir su valor.

## SALIDA

1-n: para cada alteración al valor de n, se escribe n. Tipo entero.

**DIAGRAMA DE FLUJO**



8.- Escribe un algoritmo que le pida al usuario dos enteros, de manera que el primero sea menor que el segundo, sino se cumple esta condición los volveremos a pedir hasta que se cumpla. Una vez introducidos correctamente mostraremos la suma de todos los enteros comprendidos entre ambos números incluidos ellos. Por ejemplo, para los números 3 y 7, la suma seria 25.

### E-P-S

## ENTRADA

1-Dos números enteros a y b. TIPO ENTERO

## PROCESO

## 1-Pedir al usuario que ingrese los números a y b.

## 2-Mientras a no sea menor que b (es decir, mientras a ≥ b):

## Mostrar mensaje de error indicando que el primer número debe ser menor que el segundo.

## Volver a pedir ambos números.

## 3-Una vez que se cumple la condición (a < b):

## Inicializar suma ← 0.

## Para cada entero i desde a hasta b (inclusive):

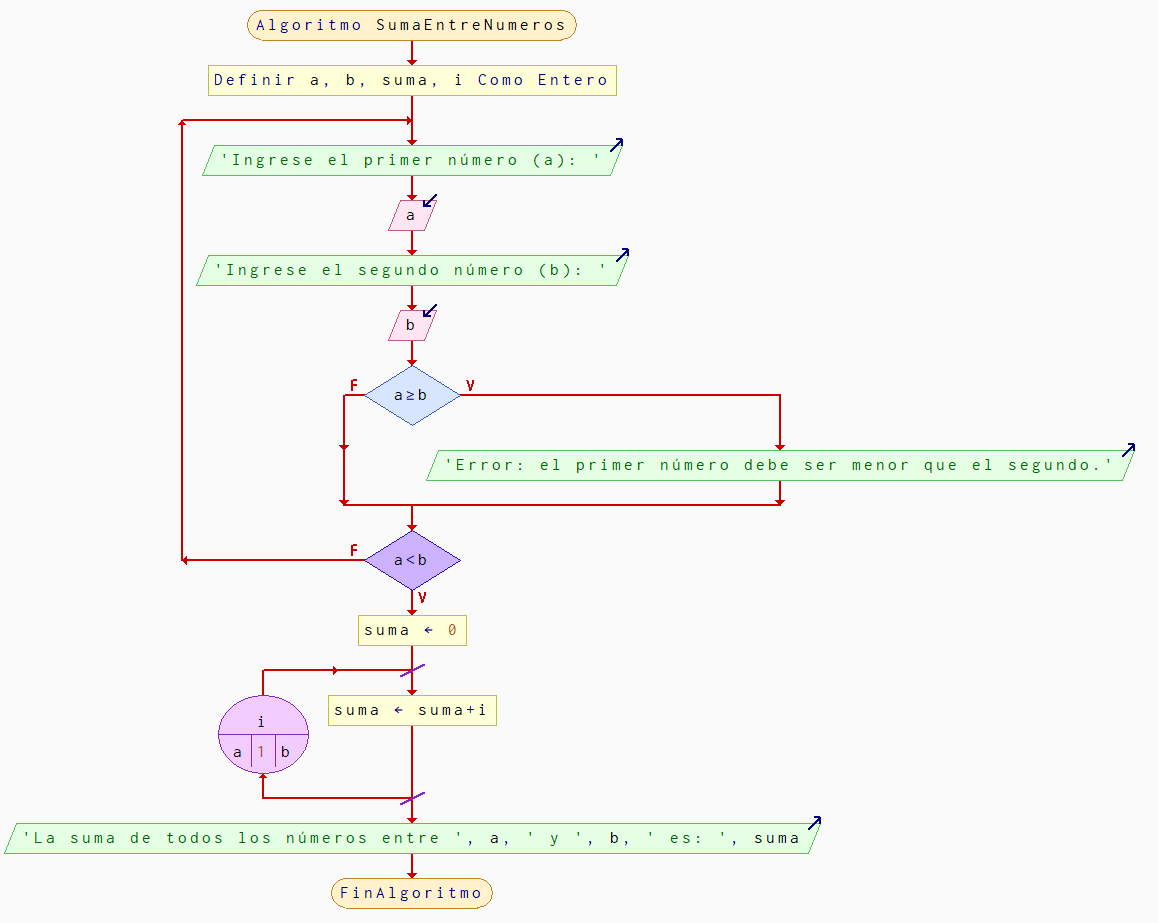
## suma ← suma + i.

## 4-Guardar el resultado de la suma.

## SALIDA

1-Mostrar el resultado de la suma de todos los números comprendidos entre a y b

**DIAGRAMA DE FLUJO**



9.- En una empresa trabajan n empleados cuyos sueldos oscilan entre $100 y $500, realizar un algoritmo que lea los sueldos que cobra cada empleado e informe cuántos empleados cobran entre $100 y $300 y cuántos cobran más de $300. Además, el programa deberá informar el importe que gasta la empresa en sueldos al personal.

### E-P-S

## ENTRADA

1-Numero empleados: la cantidad de empleados que van a realizar el cobro. De tipo entero

2-Sueldo\_empleado\_actual: el sueldo que el n-ésimo empleado está cobrando. De tipo real

## PROCESO

1-Mediante un bucle con cantidad de iteraciones igual al número de empleados, preguntar por el sueldo que cobra el empleado actual.

2-En la n-ésima iteración del bucle mencionado en el anterior paso, preguntar por un número entre los valores de 100 y 500, sea cual sea el valor, acumularlo en importe\_empresa, si el número está entre 100 y 300, aumentar el contador de cobros\_entre\_100\_y\_300 en 1, si es mayor a 300, aumentar el contador de cobros\_mas\_de\_300 en 1.

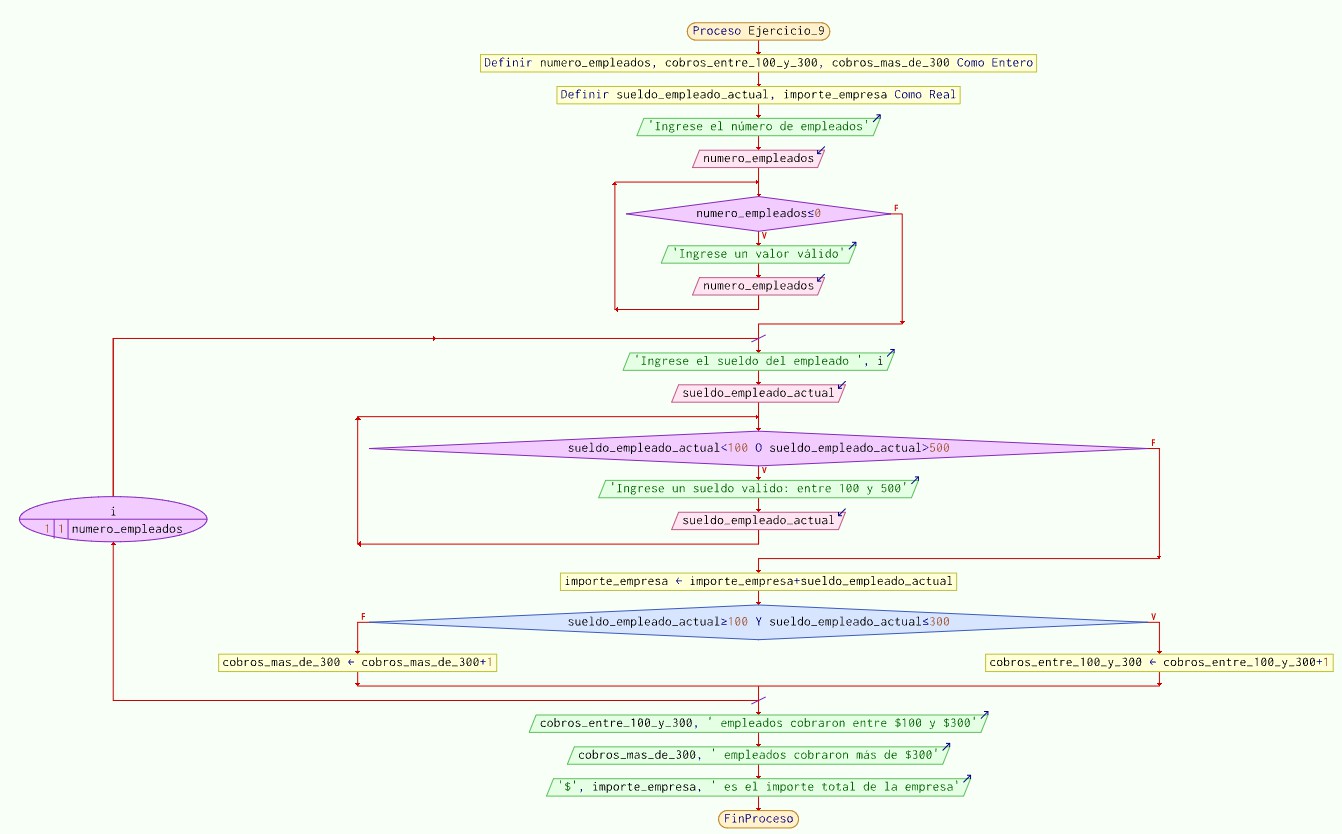
## SALIDA

1-importe\_empresa: el importe que gastará la empresa en los cobros de todos los empleados. De tipo real

2-cobros\_entre\_100\_y\_300: la cantidad de empleados que cobran sueldos entre $100 y $300. De tipo entero

3-cobros\_mas\_de\_300: la cantidad de empleados que cobran sueldos mayores a

$300. De tipo entero

**DIAGRAMA DE FLUJO**

10.- Elaborar un programa que calcule la suma de los números múltiplos de 3 a partir del número 9 y finaliza en el número 45, no debe incluirse en la suma los números comprendidos entre 21 y 27.

### E-P-S

## ENTRADA

1-numero\_inicio: el número desde donde se empieza a resolver el problema. Valor: 9 numero\_fin: el número donde finaliza la resolución del problema. Valor: 45 TIPO ENTERO

2-restriccion\_inicio: el número desde donde empieza la exclusión de números para incluir en la suma. Valor: 21. TIPO ENTERO

3-restriccion\_fin: el número donde termina la exclusión de números para incluír en la suma. Valor: 27. TIPO ENTERO

## PROCESO

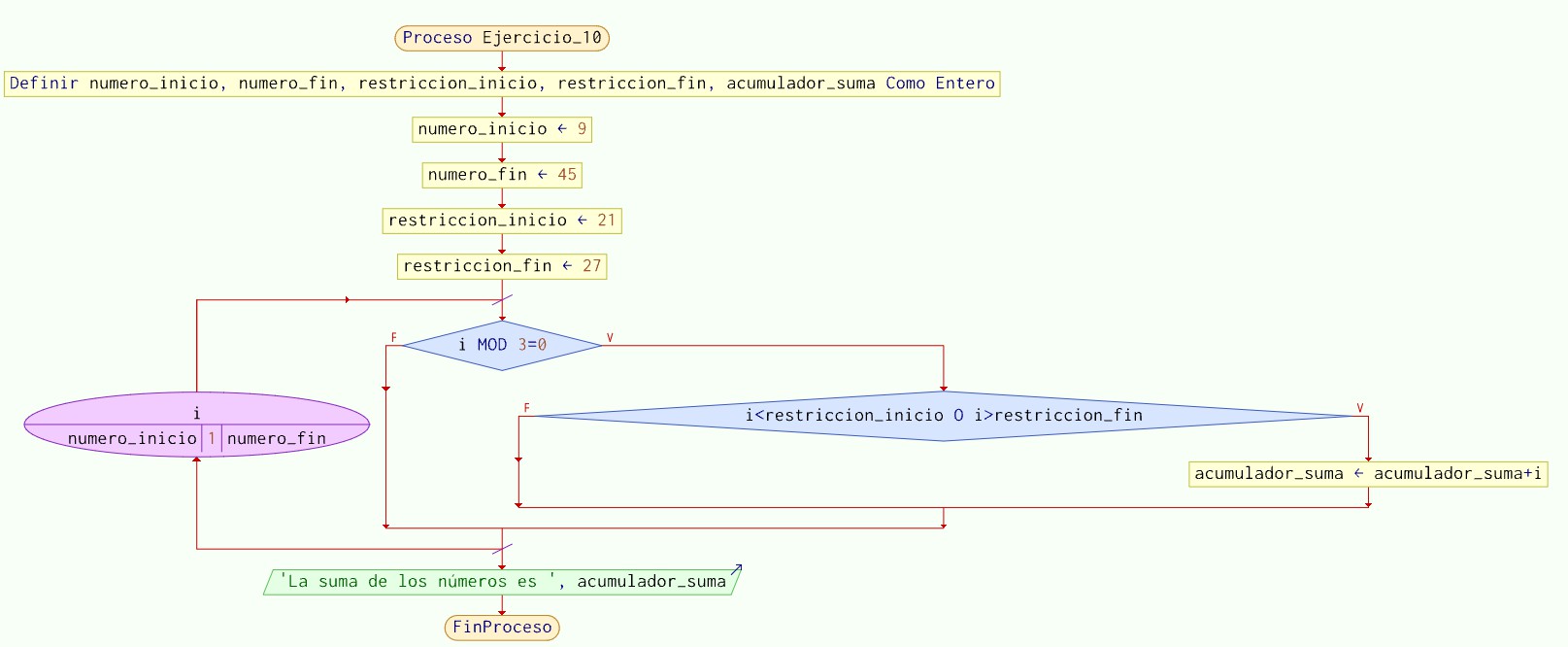
1-Definir un acumulador donde se guardará la suma de los números, llamado acumulador\_suma.

2-Iniciar un bucle que inicie en numero\_inicio y termine en numero\_fin, el bucle sóloiterará sobre los números múltiplos de 3.

3-Si el número en la iteración actual es menor a restriccion\_inicio o mayor a restriccion\_fin, su valor se suma al acumulador\_suma.

## SALIDA

1-Acumulador\_suma: el acumulador resultante de la suma. Tipo: enter

**DIAGRAMA DE FLUJO**