

Apellido y Nombre:

Fecha:/...../.....

CONCEPTOS TEÓRICOS A TENER EN CUENTA

Las computadoras están especialmente diseñadas para todas aquellas aplicaciones en las que una operación o conjunto de ellas deben repetirse muchas veces.

La cantidad de iteraciones o repeticiones puede ser fija (previamente determinada por el programador) o variable (depende de algún dato o evento del programa).

Las sentencias que permiten especificar la repetición de un conjunto de instrucciones son:

- **Para** (PARA-FIN_PARA)
- **Mientras** (MIENTRAS-FIN_MIENTRAS)
- **Repetir** (REPETIR-HASTA_QUE)

PROBLEMAS REPETITIVOS O CÍCLICOS: son aquellos que para resolverse aplican un conjunto de acciones repetidamente (un número definido o indefinido de veces).

BUCLÉS: estructuras de control que permiten repetir una secuencia de instrucciones.

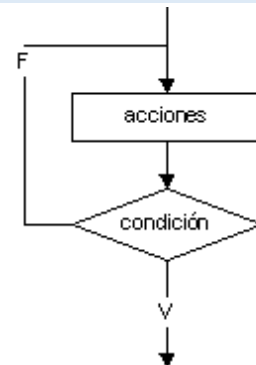
ITERACIÓN: la iteración o repetición se refiere a la ejecución de un conjunto de acciones.

ESTRUCTURAS DE CONTROL REPETITIVAS

DIAGRAMA DE FLUJO Y SINTAXIS EN PSEUDOCÓDIGO: REPETIR, MIENTRAS Y PARA.

REPETIR

REPETIR
 acciones
HASTA_QUE condición

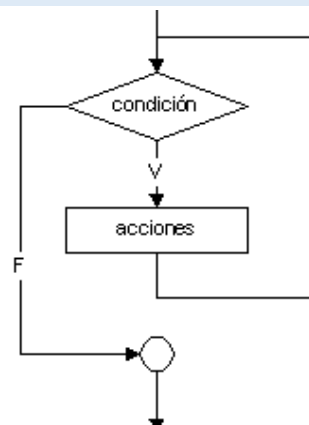


Funcionamiento:

1. Se ejecuta el bloque de ACCIONES incluido en el bucle.
2. Se evalúa la CONDICIÓN.
3. Si la CONDICIÓN toma el valor Falso, se retorna al paso 1, si toma valor Verdadero, se sigue al paso 4.
4. Sale del bucle.

MIENTRAS

MIENTRAS condición **HACER**
 acciones
FIN_MIENTRAS

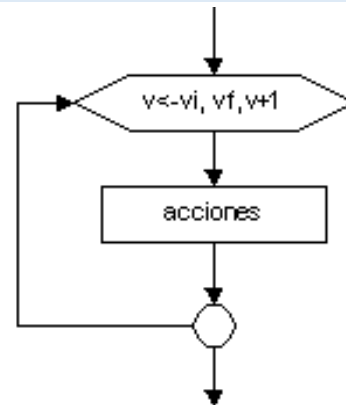


Funcionamiento:

1. Se evalúa la CONDICIÓN.
2. Si la CONDICIÓN toma el valor Verdadero, se sigue con el paso 3, si la CONDICIÓN toma valor FALSO, se sigue al paso 4.
3. Se ejecuta el bloque de ACCIONES incluido en el bucle y se retorna al paso 1.
4. Sale del bucle.

PARA

PARA v **DESDE** vi **HASTA** vf **CON PASO** n **HACER**
 acciones
FIN_PARA



Funcionamiento:

1. Se asigna a la variable de control o índice el *valor inicial* (vi)
2. Si la CONDICIÓN ($vi \leq vf$) toma el valor Verdadero, se sigue con el paso 3, si la CONDICIÓN ($vi \leq vf$) toma valor FALSO, se sigue al paso 5.
3. Por defecto, se incrementa ($v+1$) o decrementa ($v-1$) el índice en una unidad, aunque puede especificarse otro valor.
4. Se ejecuta el bloque de ACCIONES incluido en el bucle, se retorna al paso 2.
5. Sale del bucle.

EJERCICIOS RESUELTOS

1. Diseñe un algoritmo (diagrama de flujo y pseudocódigo) que sume 30 valores ingresados por el usuario.

¿Cuál es el problema que debo resolver?

Calcular la suma de 30 valores 👍

¿Qué datos necesito para resolver el problema?

30 valores generados por el usuario 👍

¿Cuál es el resultado que debo obtener?

La suma de 30 valores 👍

¿Cuáles son los pasos que debo realizar para obtener la solución?

- 1) Obtener los 30 valores 👍
- 2) Sumar cada uno de los 30 valores ingresados 👍
- 3) Mostrar el resultado calculado 👍



1

¿Cómo realizo 30 lecturas y 30 sumas? Tendré que escribir 60 instrucciones?



2

Ya sé, escribo una lectura y una suma, con un bucle se repetirán 30 veces



3

Pero, qué tipo de bucle usaré?



4

Mmmmm Repetir, Mientras o Para?



5

Tengo 30 valores de entrada, y debo calcular 30 sumas



6

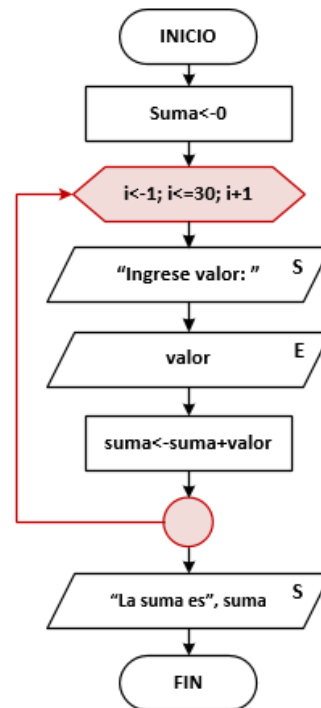
Ya sé, cuando la cantidad de veces que debo aplicar una operación es fija, uso PARA!!!!

```

PROGRAMA Ejemplo_1
VARIABLES
    suma, i, valor: ENTERO
INICIO
    suma ← 0
    PARA i DESDE 1 HASTA 30 HACER
        ESCRIBIR "Ingrese valor:"
        LEER valor
        suma ← suma + valor
    FIN PARA
    ESCRIBIR "la suma es:" suma
FIN

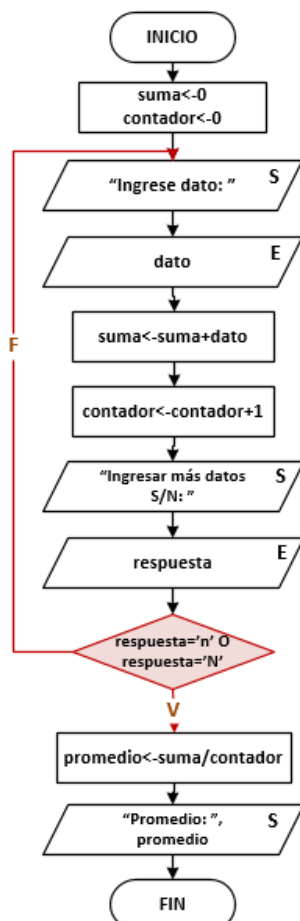
```

ACUMULADOR: es una variable cuya misión es almacenar cantidades variables resultantes de sumas o productos sucesivos.



El algoritmo tiene por objetivo **sumar 30 valores** ingresados por el usuario, lo que implica realizar 30 operaciones de lectura y 30 operaciones de suma. Este conjunto de operaciones puede resumirse aplicando estructuras repetitivas, en este caso, utilizando la estructura **PARA** ya que se conoce la **cantidad de repeticiones** a realizar. En cada iteración (repetición), la variable *valor* se carga con el dato ingresado por el usuario e inmediatamente se adiciona a la variable *suma*. La variable *suma*, cuyo valor inicial es cero (neutro de la operación de adición), funciona como un **acumulador** de los valores.

2. Diseñe un algoritmo (diagrama de flujo y pseudocódigo) que calcule el promedio de los valores ingresados por el usuario. Considere que el ingreso finaliza a pedido del usuario.



PROGRAMA Ejemplo_2

VARIABLES

```

    suma, contador: ENTERO
    promedio, dato: REAL
    respuesta: CARACTER

```

INICIO

```

    suma ← 0
    contador ← 0

```

REPETIR

```

    ESCRIBIR "Ingrese dato:"
    LEER dato
    suma ← suma + dato
    contador ← contador + 1
    ESCRIBIR "Ingresar más datos s/n:"
    LEER respuesta

```

```

HASTA QUE respuesta = 'n' O respuesta = 'N'
    promedio ← suma / contador
    ESCRIBIR "Promedio:" promedio

```

FIN

CONTADOR: es una variable cuyo valor se incrementa o decrementa en una cantidad constante en cada iteración. El contador puede ser positivo o negativo.

CENTINELA: es aquel valor especial que, al generarse en el bucle, permite finalizar las iteraciones.

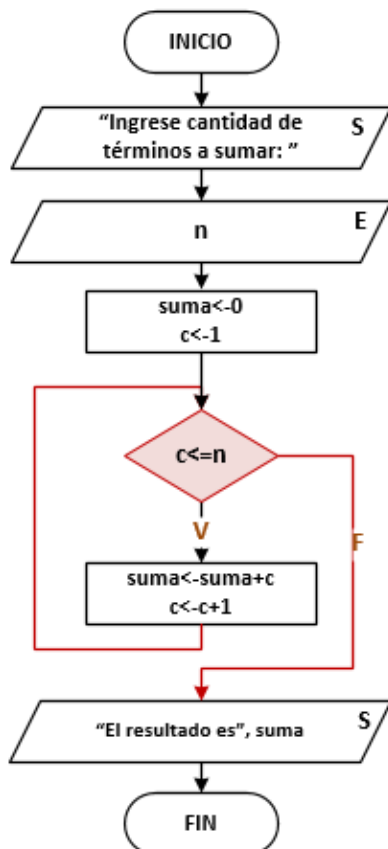
El algoritmo tiene por objetivo **calcular el promedio de un conjunto valores** ingresados por el usuario, lo que implicará ejecutar varias operaciones de lectura y suma para obtener el promedio. En este caso, la cantidad de valores a promediar es desconocida ya que depende de la respuesta del usuario (el ingreso finaliza a petición del usuario). Al desconocer *a priori* la cantidad de datos será necesario contar los valores ingresados mediante la variable *contador*. La estructura REPETIR ejecutará la lectura y suma de valores en tanto la variable *respuesta* sea distinta de 'n' o 'N'; finalizando el bucle cuando el usuario ingrese 'n' o 'N'. Un bucle que utiliza un valor específico para su condición de salida se denomina **bucle controlado por centinela**.

3. Diseñe un algoritmo (diagrama de flujo y pseudocódigo) que resuelva la siguiente expresión:

$$\sum_{c=1}^n c = 1 + 2 + 3 + \dots + n$$

SUMATORIA: Σ es una notación matemática utilizada para resumir la suma de *n* o infinitos términos.

La variable *c* asumirá los valores del rango especificado por los límites inferior (1) y superior (*n*) de la sumatoria. Como puede observarse cada término de la suma se corresponde con los valores que tomará la variable *c*.



PROGRAMA Sumatoria

VARIABLES

n, *c*, *suma*: ENTERO

INICIO

ESCRIBIR "Ingrese la cantidad de términos a sumar:"

LEER *n*

suma ← 0

c ← 1

MIENTRAS *c* ≤ *n* **HACER**

suma ← *suma* + *c*

c ← *c* + 1

FIN MIENTRAS

ESCRIBIR 'El resultado es:' *suma*

FIN

El algoritmo tiene por objetivo **sumar los *n* términos** de la sumatoria especificada. La variable *c* funciona como un **contador** que controla la cantidad de términos que se sumarán (repeticiones a realizar). En este caso, una estructura repetitiva que utiliza un contador para evaluar su condición de salida se denomina **bucle controlado por contador**. La finalización del bucle sucede cuando una variable contador (*c*) alcanza el valor final (*n*). Para almacenar los términos de la sumatoria se utiliza la variable *suma* (acumulador), cuyo valor inicial será cero (neutro de la operación de suma).

4. Dado el siguiente algoritmo escrito en lenguaje natural:

- realice el diagrama de flujo equivalente
- realice la prueba de escritorio para los valores: *m*=6 y *n*=3, *m*=4 y *n*=4,
- determine el objetivo del algoritmo y
- escriba el pseudocódigo equivalente.

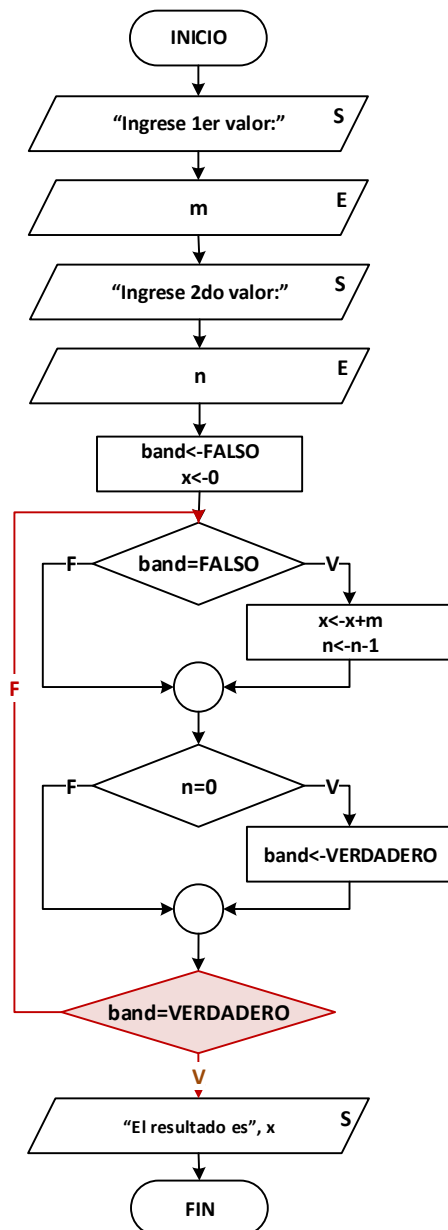
ALGORITMO

a)

Secuencia de acciones

- 1) Solicitar al usuario 2 valores numéricos
- 2) Guardar en una variable el valor lógico FALSO
- 3) Inicializar una variable que cumpla la función de acumulador
- 4) Evaluar la variable lógica y en función del resultado realizar lo siguiente: a) si la variable lógica es FALSA, debe adicionarse el primer valor de entrada a la variable acumulador y descontar una unidad al segundo valor de entrada; b) si la variable lógica es VERDADERA, no se realizarán acciones.
- 5) Evaluar el segundo valor de entrada y realizar lo siguiente: a) si el valor es cero debe guardarse VERDADERO en la variable lógica; b) si el valor es distinto de cero, no se realizarán acciones.
- 6) Evaluar la variable lógica y realizar lo siguiente: a) si la variable es VERDADERA, debe mostrarse el valor del acumulador y finalizar; b) si la variable es FALSA se debe volver al paso 4.

BANDERA: variable lógica utilizada para indicar la ocurrencia de un evento. El cambio de valor de una bandera puede usarse para finalizar las iteraciones de un bucle.



b)

Pasos	m	n	band	X
01	6	3	F	0
02	6	2	F	6
03	6	1	F	12
04	6	0	V	18
Resultado 18				
01	4	4	F	0
02	4	3	F	4
03	4	2	F	8
04	4	1	F	12
05	4	0	V	16
Resultado 16				

- d) El objetivo del algoritmo es determinar el producto de dos números ingresados por el usuario.

d)

```

PROGRAMA ejemplo4
VARIABLES
    m, n, x: ENTERO
    band: LOGICO

INICIO
    ESCRIBIR "Ingrese primer valor:"
    LEER m
    ESCRIBIR "Ingrese segundo valor:"
    LEER n
    band<-FALSO
    x<-0
    REPETIR
        SI band=FALSO ENTONCES
            x<-x+m
            n<-n-1
        FINSI
        SI n=0 ENTONCES
            band<-VERDADERO
        FIN_SI
    HASTA QUE band=VERDADERO
    ESCRIBIR "Resultado: ", x
FIN
  
```

El algoritmo calcula el producto, mediante sumas sucesivas, de 2 números ingresados por el usuario. Con este método el producto se obtiene sumando el primer valor tantas veces como lo indique el segundo. Así, suponiendo $m=2$ y $n=3$ se realizará la suma $2+2+2=6$. Esto implica repetir la operación de suma varias veces, lo que puede implementarse mediante estructuras PARA, MIENTRAS o REPETIR. En este caso, la solución se planteó con estructuras REPETIR utilizando como criterio de finalización del bucle el concepto de bandera. La variable *band* se utiliza para detectar en qué momento el segundo valor (n) se hace 0, y entonces modificar la variable *band* para hacer VERDADERA la condición de salida de REPETIR.

EJERCICIOS A RESOLVER

1. Modifique el ejercicio 1 del TP3 de modo que muestre el mensaje N veces. Diseñe 3 versiones del algoritmo utilizando las distintas estructuras repetitivas.
2. En el TP1 ayudaste a este pequeño a determinar cuál era el siguiente número de una serie de valores. Ahora, podrías diseñar un algoritmo (diagrama de flujo y pseudocódigo) que le permita calcular cualquier valor de la serie. Por ejemplo, si desea obtener el 4° término, el algoritmo debería generar como resultado 10.

¿Cuál es el siguiente valor de la serie?

1, 3, 6, 10, 15, 21, ?



3. Diseñe un algoritmo (diagrama de flujo y pseudocódigo) que permita ingresar números y determinar cuál de ellos es el mínimo. Considere que el ingreso se realiza en tanto el usuario no introduzca un CERO. Luego, diseñe una segunda versión que además del mínimo obtenga el máximo.
4. Considerando que la potencia de un número entero positivo a elevado a un número entero positivo b , puede expresarse como el producto sucesivo de a , b veces, diseñe un algoritmo (diagrama de flujo y pseudocódigo) que realice el cálculo de potencia mediante productos sucesivos. Controle la finalización del bucle aplicando el concepto de bandera.
5. Diseñe un algoritmo (diagrama de flujo y pseudocódigo) que calcule la siguiente expresión:

$$\sum_{i=1}^n 2 \times i - 1$$

Tenga en cuenta que el cálculo sólo debe realizarse si n es positivo, caso contrario, debe presentarse un mensaje de error.

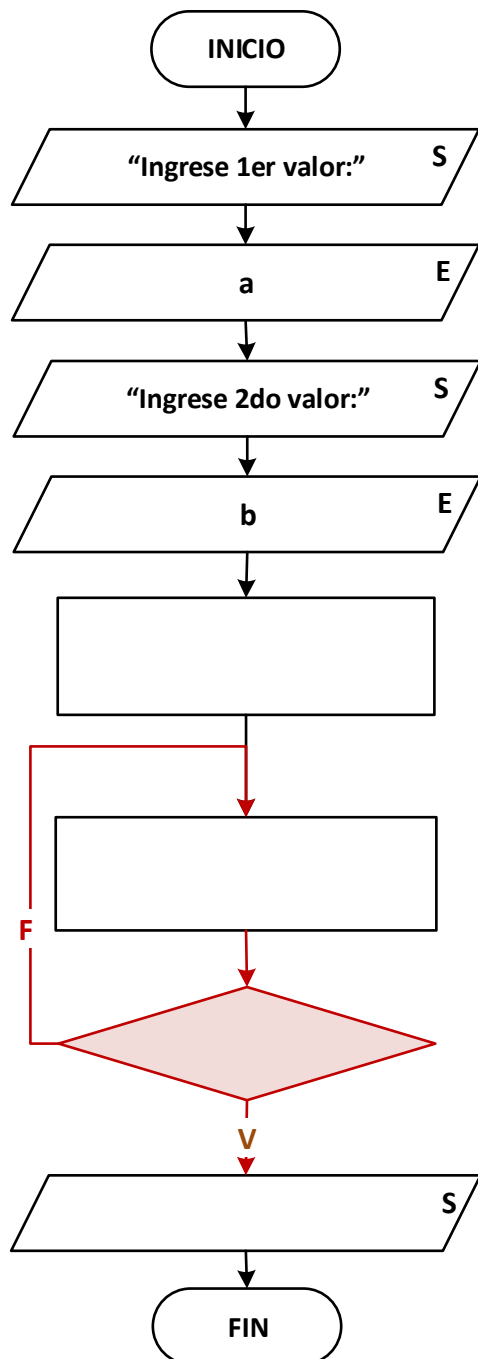
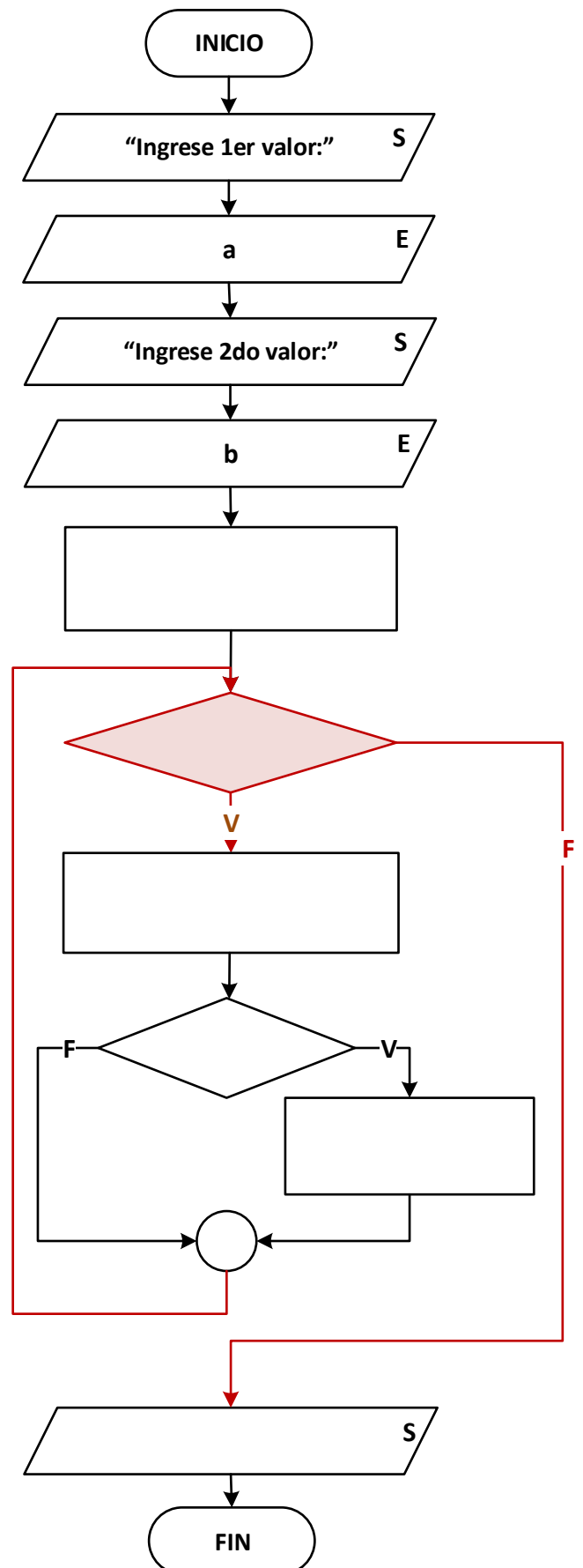
Además, considere que el bucle de cálculo se implementa con estructuras MIENTRAS controlado por centinela.

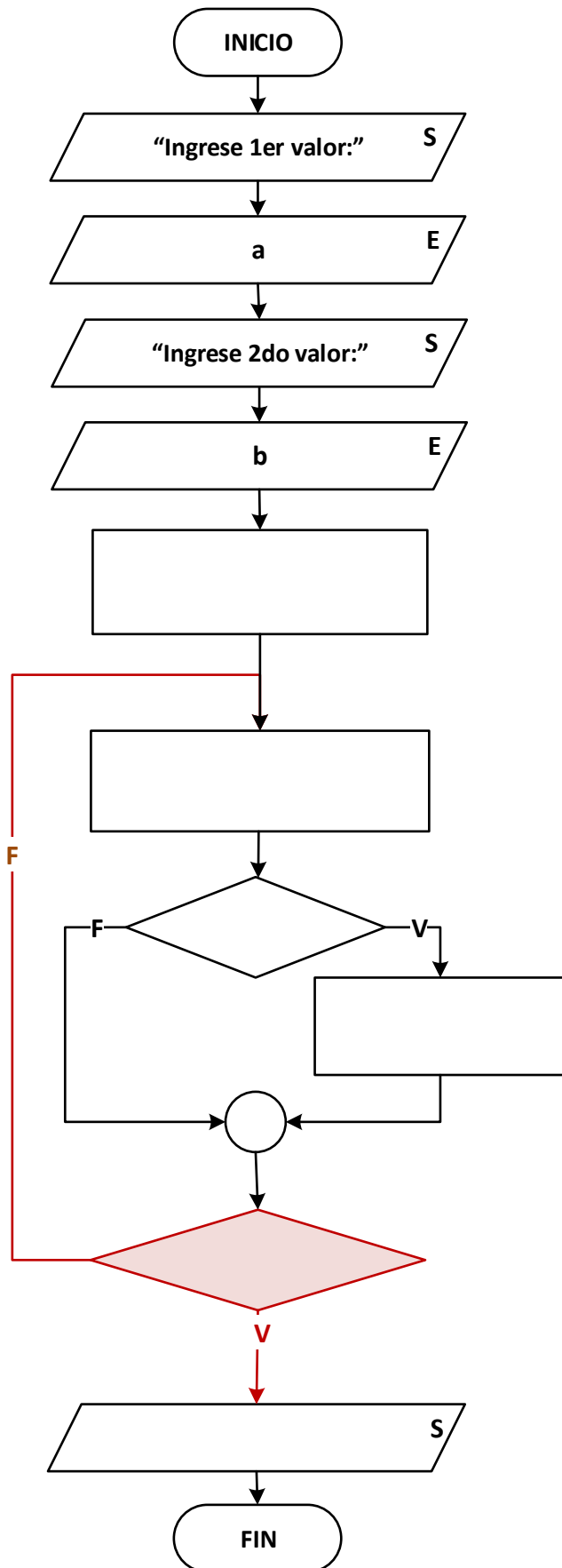
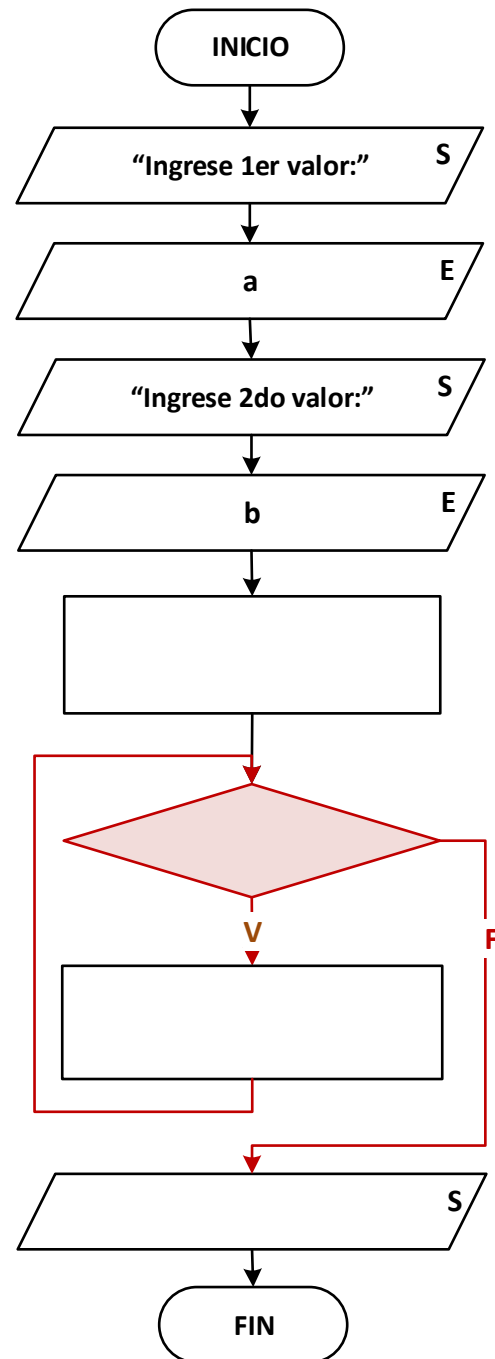
Además realice la **prueba de escritorio** del algoritmo diseñado para $n=3$ y $n=5$.

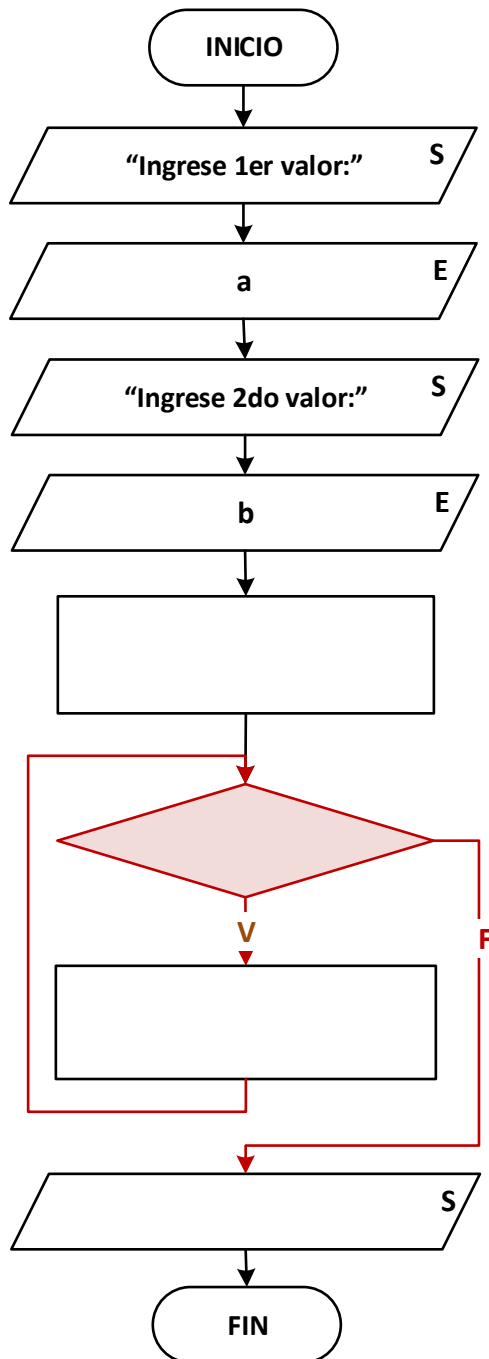
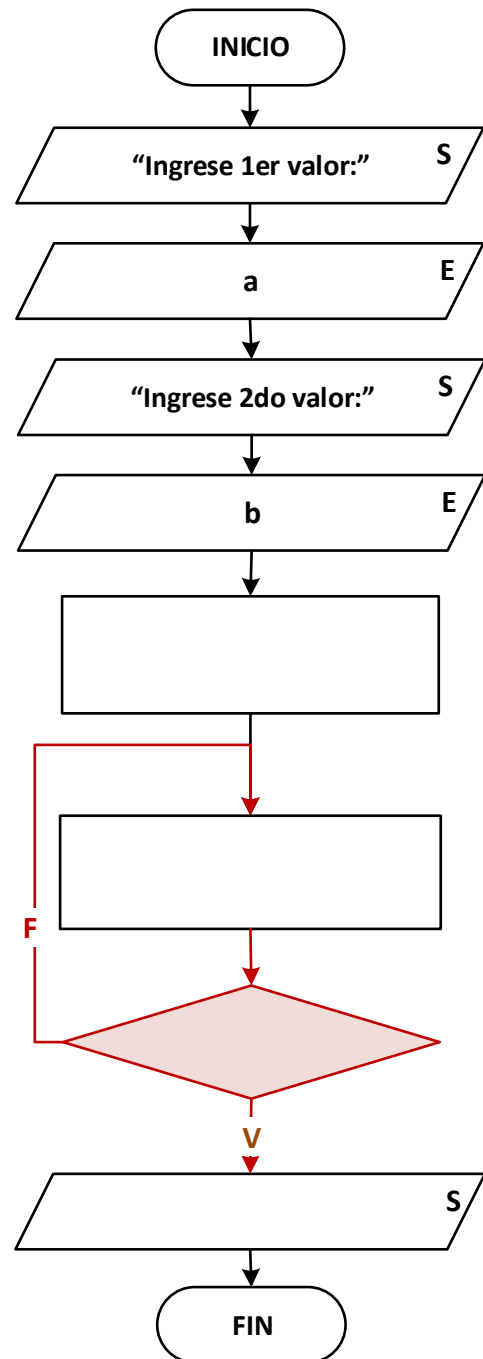
6. Considerando que la división entera de dos números enteros positivos, a y b , puede expresarse como la resta sucesiva entre a y b (siempre que a sea mayor o igual que b), diseñe un algoritmo (diagrama de flujo y pseudocódigo) que calcule el cociente y el resto de la división mediante restas sucesivas. Escriba una versión con estructuras MIENTRAS y otra con estructuras REPETIR.
Nota: la *cantidad* de veces que pueda realizar la resta de a y b ($a \geq b$) indica el cociente de la división.
7. Considerando que el factorial de un número entero positivo n , denotado como $n!$, se calcula como el producto de los n primeros números naturales, diseñe un algoritmo (diagrama de flujo y pseudocódigo) que realice este cálculo. Desarrolle una versión para cada estructura repetitiva estudiada.

Nota: El factorial de n es $n!=n \times (n-1) \times (n-2) \times \dots \times 3 \times 2 \times 1$. Por ejemplo, el factorial de 5 es $5!=5 \times 4 \times 3 \times 2 \times 1 \rightarrow 5!=120$. En particular, el factorial de 0 es $0! = 1$.

8. Considerando que el producto de un número entero positivo a por otro entero positivo b , puede expresarse como la suma sucesiva de a , b veces, complete los siguientes diagramas de flujo para realizar este cálculo, teniendo en cuenta la estructura repetitiva utilizada y el criterio de finalización de bucle aplicado.

FINALIZACIÓN POR CONTADOR**FINALIZACIÓN POR BANDERA**

FINALIZACIÓN POR BANDERA**FINALIZACIÓN POR CENTINELA**

FINALIZACIÓN POR CONTADOR**FINALIZACIÓN POR CENTINELA**

9. Dado el siguiente algoritmo (pseudocódigo):

```

PROGRAMA misterio
VARIABLES
  a, b, c, x: ENTERO
INICIO
  REPETIR
    SI (a>b) ENTONCES
      x←a
      a←b
      b←x
    FIN_SI
    SI (b>c) ENTONCES
      x←b
      b←c
      c←x
    FIN_SI
  HASTA_QUE (a<=b Y b<=c)
  ESCRIBIR "Salida: ", a, b, c
FIN
  
```

- Realice la prueba de escritorio para los valores $a=4$, $b=1$, $c=2$ y $a=4$, $b=7$, $c=1$.
- Determine el propósito del algoritmo.
- Dibuje el diagrama de flujo equivalente.

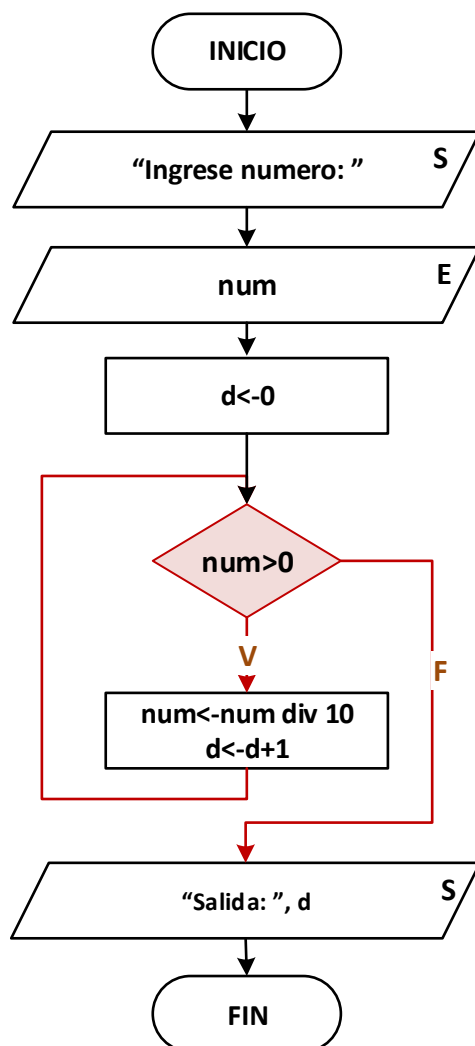
10. Dado el siguiente algoritmo (pseudocódigo):

```

PROGRAMA enigma
VARIABLES
  q, w, x: ENTERO
  band: LOGICO
INICIO
  x ← 1
  band ← VERDADERO
  ESCRIBIR "Ingrese dos números enteros y positivos"
  LEER q, w
  REPETIR
    SI (w=0) ENTONCES
      band ← FALSO
    SINO
      x ← x*q
      w ← w-1
    FIN_SI
  HASTA_QUE (band=FALSO)
  ESCRIBIR "Resultado", x
FIN
  
```

- Realice la prueba de escritorio para los valores $q=4$ y $w=2$, $q=3$ y $w=3$.
- Determine el propósito del algoritmo.
- Dibuje el diagrama de flujo equivalente.

11. Dado el siguiente algoritmo (diagrama de flujo):



- Realice la prueba de escritorio para $num=216$ y $num=1973$.
- Determine el propósito del algoritmo.
- Escriba el pseudocódigo equivalente.

12. Teniendo en cuenta que en el TP1 resolviste cómo se forma la siguiente serie de igualdades, serás capaz de diseñar un algoritmo que dado un número de n dígitos calcule el valor al que se iguala?

8806=6 7111=0 2172=0 6666=4 1111=0 7662=2 9312=1 0000=4 2222=0 3333=0

5555=0 8193=3 8096=5 7777=0 9999=4 7756=1 6855=3 9881=5 5531=0 2581= ¿?