

Tabla 2.22

NUMERO DE CORRIDA	DATOS			RESULTADO
	TIPOENF	EDAD	DIAS	
1	3	35	4	80
				96
2	2	20	6	105.6
				100
3	3	40	5	256
				281.6
4	4	16	8	

: Expresa valores que se imprimen

Programa 2.29

## HOSPITAL

{El programa dados como datos la edad del paciente, el tipo de enfermedad padecida y el número de días hospitalizados, calcula el costo total por internación}

{TIPOENF, EDAD y DIAS son variables de tipo entero. COSTOT es una variable de tipo real}

1. Leer TIPOENF, EDAD y DIAS
2. Si TIPOENF igual
  - 1: Hacer COSTOT ← DIAS \* 25
  - 2: Hacer COSTOT ← DIAS \* 16
  - 3: Hacer COSTOT ← DIAS \* 20
  - 4: Hacer COSTOT ← DIAS \* 32
3. {Fin del condicional del paso 2}
4. Si EDAD  $\geq$  14 y EDAD  $\leq$  22 entonces
 

Hacer COSTOT ← COSTOT \* 1.10
5. {Fin del condicional del paso 4}
6. Escribir "Costo Total:", COSTOT

## 3

## Estructuras algorítmicas repetitivas

Metodología de la Programación  
Osvaldo Cairo

### 3.1 Introducción

Es muy común encontrar en la práctica algoritmos cuyas operaciones se deben ejecutar un número repetido de veces. Si bien las instrucciones son las mismas, los datos sobre los que se opera varían. El conjunto de instrucciones que se ejecuta repetidamente se llama ciclo.

Todo ciclo debe terminar de ejecutarse luego de un número finito de veces, por lo que es necesario en cada iteración del mismo, evaluar las condiciones necesarias para decidir si se debe seguir ejecutando o si debe detenerse. En todo ciclo, siempre debe existir una condición de parada o fin de ciclo.

En algunos algoritmos podemos establecer a priori que el ciclo se repetirá un número definido de veces. Es decir, el número de repeticiones no dependerá de las proposiciones dentro del ciclo. Llamaremos *repetir* a la estructura algorítmica repetitiva que se ejecuta un número definido de veces.

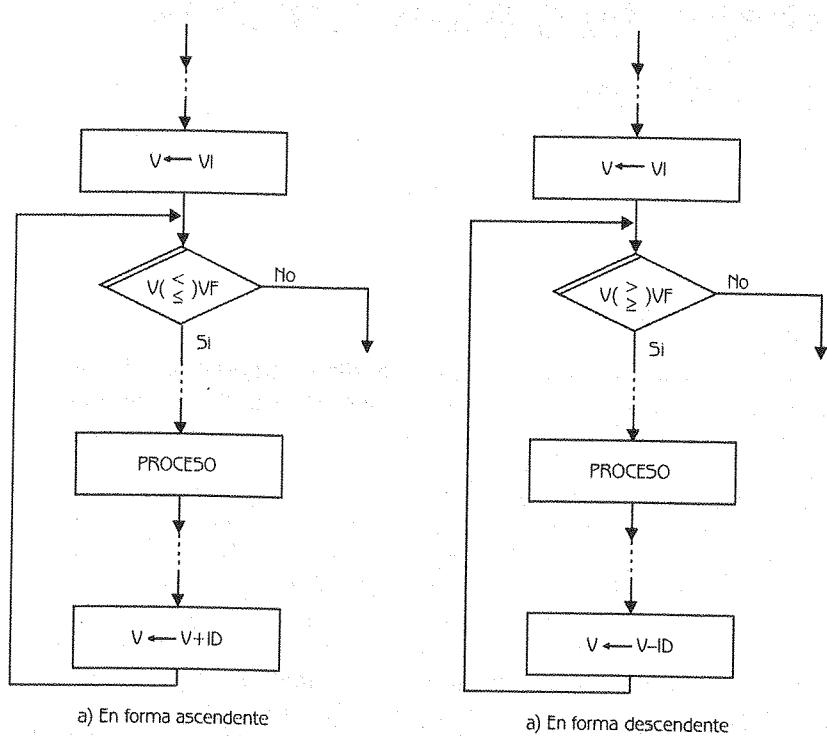
Por otra parte, en algunos algoritmos no podemos establecer a priori el número de veces que ha de ejecutarse el ciclo, sino que este número dependerá de las proposiciones dentro del mismo. Llamaremos *mientras* a la estructura algorítmica repetitiva que se ejecuta mientras la condición evaluada resulta verdadera.

### 3.2 La estructura repetitiva *repetir* (FOR)

La estructura *repetir* conocida comúnmente como FOR, es la estructura algorítmica adecuada para utilizar en un ciclo que se ejecutará un número definido de veces. Este tipo de estructura está presente en todos los lenguajes de programa-

ción, ya sean estructurados u orientados a objetos. Por ejemplo cuando necesitamos calcular la nómina total de la empresa, tenemos que sumar los sueldos de los  $N$  empleados de la misma. Cuando necesitamos obtener el promedio de calificaciones de un curso, debemos sumar las  $N$  calificaciones de los alumnos y dividir esa suma entre  $N$ . Es decir, sabemos de antemano cuántas veces tenemos que repetir una determinada operación, acción o tarea. El número de repeticiones no depende de las proposiciones dentro del ciclo. El número de veces se obtiene del planteamiento del problema o de una lectura que indica que el número de iteraciones se debe realizar para  $N$  ocurrencias.

El diagrama de flujo de la estructura algorítmica repetir es el siguiente:



### Diagrama de Flujo 3.1

Donde:

V es la variable de control del ciclo.

VI es el valor inicial.

**VF** es el valor final.

**ID**: es el incremento o decremento, según sea la estructura *repetir ascendente o descendente*.

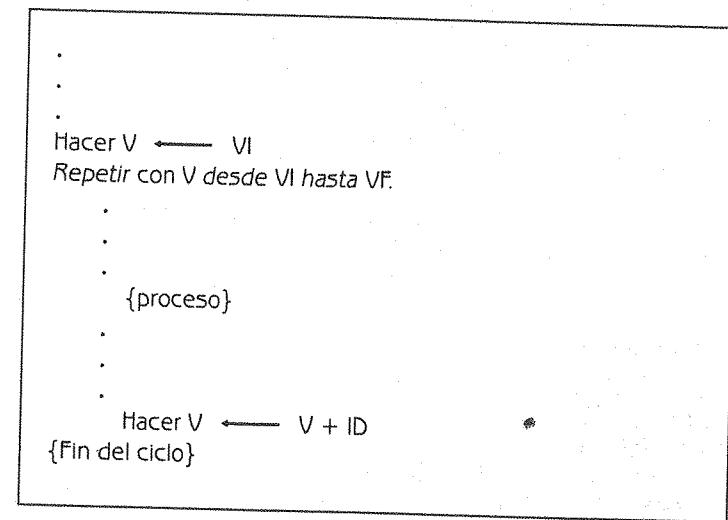
### 3.2 La estructura repetitiva *repetir* (FOR)

V ( contador del ciclo, generalmente representado por las letras I, J, K, V) toma un valor inicial y se compara con VF (valor final). El ciclo se ejecuta mientras V es menor, menor igual, mayor o mayor igual al valor de V. El valor de V se incrementa o decrementa en cada iteración. Cuando V supera el valor de V. entonces el ciclo se detiene.

Observe el lector que la modificación introducida al simbolo de la decisión –doble linea en la parte superior izquierda– en la estructura *repetir*, le será de gran utilidad tanto en la construcción como en la interpretación del diagrama de flujo. Indudablemente, este cambio sencillo en la estructura *repetir* le permitirá diferenciarlo de la otra estructura repetitiva; *mientras*.

En lenguaje algorítmico, al Diagrama de Flujo 3.1(a), lo expresamos de esta forma:

Programa 3.1



Veamos a continuación el siguiente ejemplo:

### Ejemplo 3.1

Construya un diagrama de flujo tal que dado como datos los sueldos de los 10 trabajadores de una empresa, obtenga el total de nómina de la misma. Considere además que no puede utilizar estructuras algorítmicas repetitivas en la solución del problema.

Datos: SUE1, SUE2, SUE3, SUE4, SUE5, SUE6, SUE7, SUE8, SUE9, SUE10

Donde:

$SUE_1, SUE_2, \dots, SUE_{10}$  son variables de tipo real que representan los sueldos de los 10 trabajadores.

A continuación presentamos la solución correspondiente del problema:

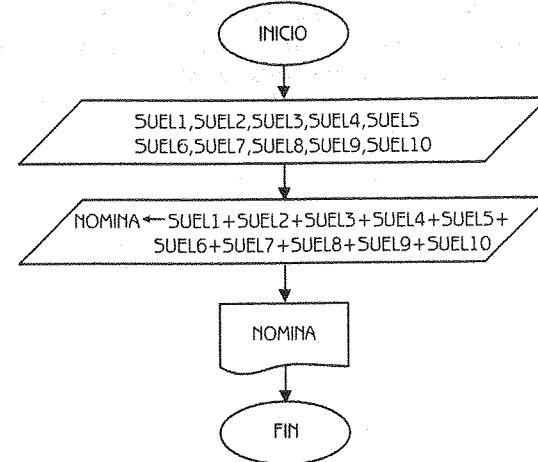


Diagrama de Flujo 3.2

### Explicación de las variables

$SUE_1, SUE_2, SUE_3, SUE_4, SUE_5, SUE_6, SUE_7, SUE_8, SUE_9$  y

$SUE_{10}$ : Variables de tipo real.

$NOMINA$ : Variable de tipo real. Almacena la suma de los sueldos de todos los trabajadores.

Consideré el lector qué sucedería si en lugar de tener 10 empleados, la empresa tuviera 100 o 1000. En realidad el problema es sencillo y se puede resolver con una estructura algorítmica repetitiva, de tal forma que el ciclo se ejecute tantas veces como empleados tenga la empresa. Observemos el siguiente ejemplo:

### Ejemplo 3.2

A continuación presentamos la solución del problema anterior, utilizando una estructura algorítmica repetitiva

Datos:  $SUE_1, SUE_2, \dots, SUE_{10}$

Donde:

$SUE_i$  es una variable de tipo real que representa el sueldo del trabajador ( $1 \leq i \leq 10$ ).

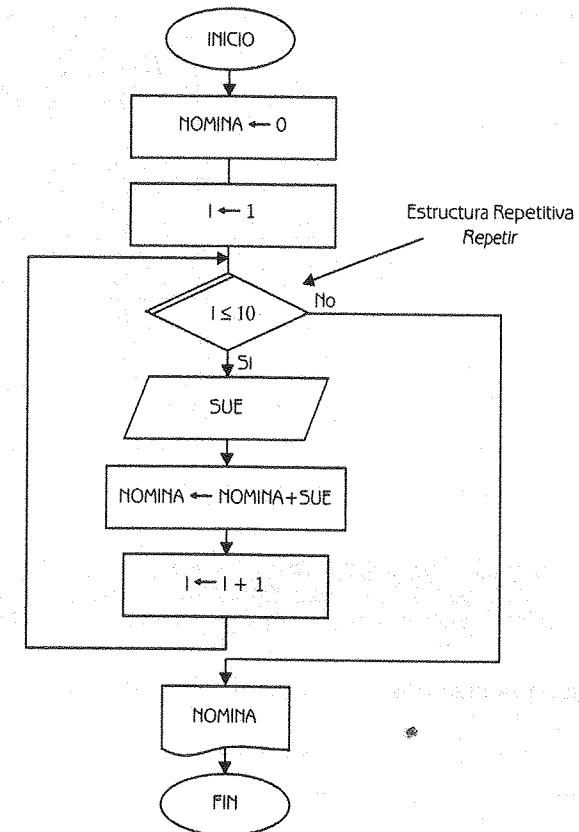


Diagrama de Flujo 3.3

### Explicación de las variables

$I$ : Es una variable de tipo entero que representa la variable de control del ciclo. Contabiliza el número de veces que ha de repetirse una determinada acción. El contador toma un valor inicial (generalmente 0 o 1) y se incrementa en la mayoría de los casos en una unidad en cada vuelta del ciclo.

$NOMINA$ : Es una variable de tipo real que representa un acumulador. Este se utiliza cuando debemos obtener el total acumulado de un conjunto de cantidades. Generalmente se inicializa en cero.

**SUE:** Es una variable de tipo real. Representa el sueldo del trabajador.

En la siguiente tabla podemos observar el seguimiento del algoritmo para los siguientes datos (SUE): \$1 500, \$890, \$700, \$950, \$2 300, \$1 650, \$1 800, \$1 400, \$760, \$900.

Inicia el Ciclo →

I	SUE	NOMINA
1		0
2	1500	1500
3	890	2390
4	700	3090
5	950	4040
6	2300	6340
7	1650	7990
8	1800	9790
9	1400	11190
10	760	11950
11	900	12850

Finaliza el Ciclo →

: Expresa la cantidad que se imprime.



### Nota:

Observe el lector que cuando inicia el ciclo se lee el sueldo (SUE), se actualiza la nómina (NOMINA) y se incrementa la variable I (contador del ciclo). Este es el criterio que utilizaremos en todo el capítulo para la construcción de tablas.

A continuación presentamos la solución del problema, en lenguaje algorítmico.

### Programa 3.2

NOMINA

{El programa calcula el total de la nómina de un grupo de 10 empleados}

{I es una variable de tipo entero. SUE y NOMINA son variables de tipo real}

1. Hacer NOMINA ← 0 e I ← 1

2. Repetir con I desde 1 hasta 10

    Leer SUE

    Hacer NOMINA ← NOMINA + SUE e I ← I + 1

3. {Fin del ciclo del paso 2}

4. Escribir NOMINA

### 3.2 La estructura repetitiva repetir (FOR)

Observemos a continuación otro ejemplo.

### Ejemplo 3.3

Escriba un diagrama de flujo tal que dado como datos N números enteros, obtenga el número de ceros que hay entre estos números.

Datos: N, NUM<sub>1</sub>, NUM<sub>2</sub>, ..., NUM<sub>N</sub>

Donde:

N es una variable de tipo entero que representa el número de datos que se ingresan.

NUM<sub>i</sub> es una variable de tipo entero que representa al número i ( $1 \leq i \leq N$ ).

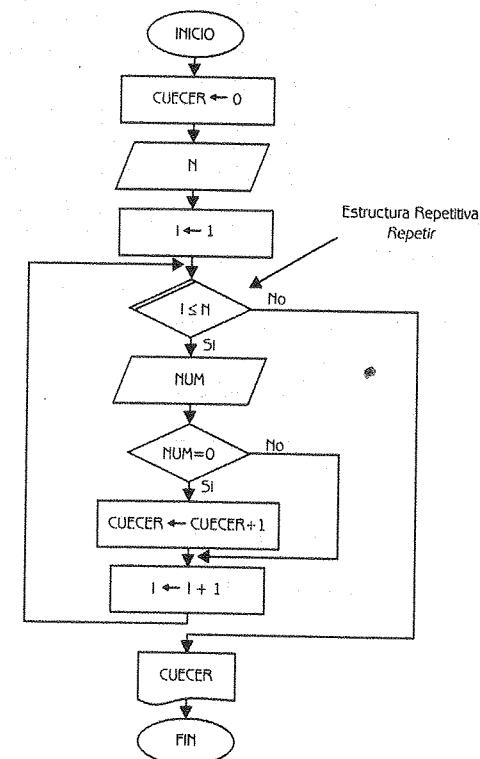


Diagrama de Flujo 3.4

**Explicación de las variables**

*I*: Variable de tipo entero. Representa al contador del ciclo.

*N* y *NUM*: Variables de tipo entero.

*CUECER*: Variable de tipo entero. Cuenta el número de ceros.

A continuación en la siguiente tabla podemos observar el seguimiento del algoritmo.

**Tabla 3.2**

<b>I</b>	<b>N</b>	<b>CUECER</b>	<b>NUM</b>
1	12	0	
2			18
3			23
4		1	0
5			17
6			22
7		2	0
8			37
9			43
10		3	0
11			27
12			41
13			53

Finaliza el Ciclo →

: Expresa el valor que se imprime.

**Programa 3.3****CUENTA\_CEROS**

{El programa cuenta cuántos ceros hay en un grupo de *N* números enteros}

{*I*, *N*, *NUM*, y *CUECER* son variables de tipo entero}

1. Hacer *CUECER* ← 0
2. Leer *N*
3. Hacer *I* ← 1
4. Repetir con *I* desde 1 hasta *N*  
    Leer *NUM*

**3.3 La estructura repetitiva *mientras* (WHILE)**

- 4.1 Si *NUM* = 0 entonces

    Hacer *CUECER* ← *CUECER* + 1

- 4.2 {Fin del condicional del paso 4.1}

    Hacer *I* ← *I* + 1

5. Fin del ciclo del paso 4

6. Escribir *CUECER*

**3.3 La estructura repetitiva *mientras* (WHILE)**

La estructura algorítmica *mientras*, comúnmente conocida como while, es la estructura adecuada para utilizar en un ciclo cuando no sabemos el número de veces que éste se ha de repetir. Dicho número depende de las proposiciones dentro del ciclo. Ejemplos en la vida cotidiana encontramos muchos. Por ejemplo, supongamos que tenemos que obtener el total de una serie de gastos, pero no sabemos exactamente cuántos son; o cuando tenemos que sacar el promedio de calificaciones de un examen, pero no sabemos precisamente cuántos alumnos lo aplicaron. Tenemos que sumar las calificaciones e ir contando el número de alumnos, esto con el fin de poder obtener posteriormente el promedio. El ciclo se repite mientras tengamos calificaciones de alumnos.

En la estructura *mientras* se distinguen dos partes:

- **Ciclo:** Conjunto de instrucciones que se ejecutarán repetidamente.
- **Condición de terminación:** La evaluación de esta condición permite decidir cuándo finalizará la ejecución del ciclo. La condición se evalúa al inicio del mismo.

El diagrama de flujo de la estructura algorítmica *mientras* es el siguiente:

Donde

*PI*: La proposición inicial, debe tener un valor verdadero inicialmente.  
Si el valor de *PI* es falso, entonces el ciclo no se ejecuta.

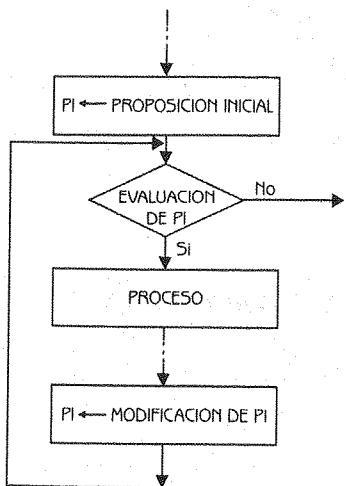


Diagrama de Flujo 3.5

Debe existir también un enunciado dentro del ciclo que afecte la condición, para evitar que el ciclo se ejecute indefinidamente.

En lenguaje algorítmico la estructura *mientras* la expresamos de esta forma:

Programa 3.4

```

    .
    .
    .
    Hacer PI ← Proposición inicial
    Mientras PI es verdadero repetir
    .
    .
    .
    {PROCESO}
    .
    .
    .
    Hacer PI ← modificación de PI
    {Fin del ciclo}
  
```

Veamos a continuación el siguiente ejemplo:

**Ejemplo 3.4**

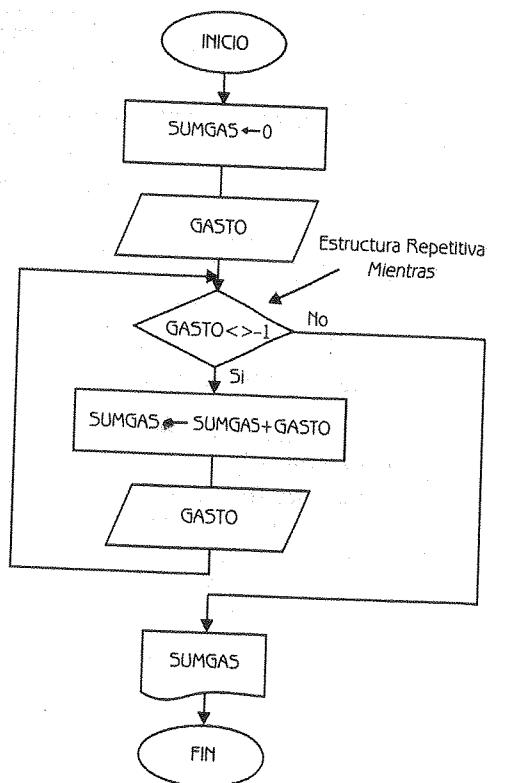
Supongamos que debemos obtener la suma de los gastos que hicimos en nuestro último viaje, pero no sabemos exactamente cuántos fueron. Los datos son expresados de forma:

Datos: GASTO<sub>1</sub>, GASTO<sub>2</sub>, ..., -1

Donde:

GASTO<sub>i</sub> es una variable de tipo real que representa el gasto número i.

A continuación presentamos la solución del problema.



## Explicación de las variables

**SUMGAS:** Es una variable de tipo real.  
Es un acumulador.  
Acumula los datos efectuados.

**GASTO:** Es una variable de tipo real. Su valor en la primera lectura debe ser verdadero, es decir distinto de -1. Su valor se modifica en cada vuelta del ciclo. Cuando gasto tome el valor de -1, entonces el ciclo se detendrá.

Diagrama de Flujo 3.6

**Nota:**

Observe el lector que en este tipo de problemas deberemos realizar dos lecturas, una antes de iniciar el ciclo y otra antes de finalizar el mismo.

En la siguiente tabla podemos observar el seguimiento del algoritmo para los siguientes datos (GASTO): \$2,528, \$3,500, \$1,600, \$1,850, \$150, -1

Inicia el Ciclo →

Tabla 3.3	
SUMGAS	GASTO
0	2528
2528	3500
6028	1600
7628	1850
9478	150
9628	-1

Finaliza el Ciclo →

: Expresa el valor que se imprime

**Programa 3.5****SUMA\_GASTOS**

{El programa obtiene el total de gastos de un viaje}

{Gasto y SUMGAS son variables de tipo real}

1. Hacer SUMGAS ← 0
2. Leer GASTO
3. Mientras GASTO < > -1 Repetir
  - Hacer SUMGAS ← SUMGAS + GASTO
  - Leer GASTO
4. {Fin del ciclo del paso 3}
5. Escribir SUMGAS

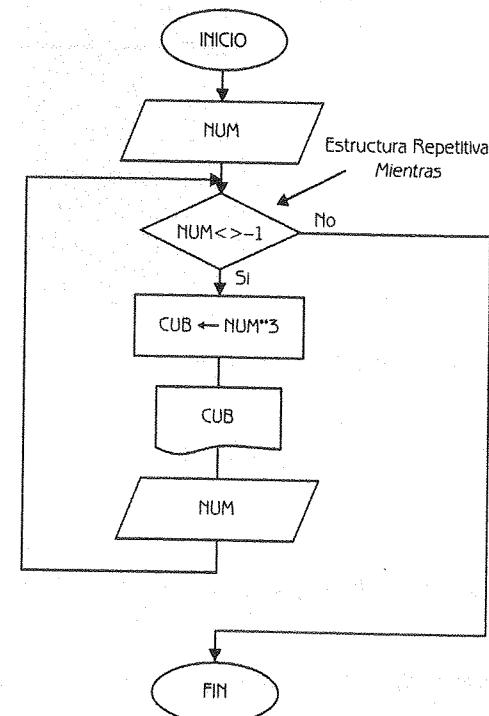
Observemos a continuación otro ejemplo.

**Ejemplo 3.5**

Escriba un diagrama de flujo tal que dado un grupo de números naturales positivos, calcule e imprima el cubo de estos números.

**3.3 La estructura repetitiva mientras (WHILE)**

**Datos:** NUM<sub>1</sub>, NUM<sub>2</sub>, NUM<sub>3</sub>, ..., -1 (NUM<sub>i</sub> es una variables de tipo entero que representa el número positivo i. El fin de datos está dado por -1).

**Explicación de las variables**

**NUM :** Variable de tipo entero.

**CUB :** Variable de tipo real. Almacena el cubo del número que se ingresa.

En la siguiente tabla podemos observar el seguimiento del diagrama de flujo.

Tabla 3.4

NUM	CUB
5	
13	125
7	2197
48	343
18	110592
27	5832
94	19683
62	830584
114	238328
-1	1481544

Inicia el Ciclo →

Finaliza el Ciclo →

: Expresa valores que se imprimen.

### Programa 3.6

#### CUBO

{El programa calcula el cubo de un grupo de números enteros positivos que se ingresan}

{NUM es una variable de tipo entero. CUB es una variable de tipo real}

1. Leer NUM
2. Mientras ( $NUM < -1$ ) Repetir
  - Hacer  $CUB \leftarrow NUM^{**} 3$
  - Escribir CUB
  - Leer NUM
3. {Fin del ciclo del paso 2}

A continuación presentamos una serie de problemas diseñados expresamente como elementos de ayuda para el análisis y la retención de los conceptos. Además se utilizan en muchos de ellos, tablas que permiten hacer un seguimiento del algoritmo.

## Problemas resueltos

### Problema 3.1

Escriba un diagrama de flujo tal que dado como datos 270 números enteros, obtenga la suma de los números impares y el promedio de los números pares.

Datos:  $NUM_1, NUM_2, \dots, NUM_{270}$

Donde:

$NUM_i$  es una variable de tipo entero que representa el número entero  $i$  que se ingresa ( $1 \leq i \leq 270$ ).

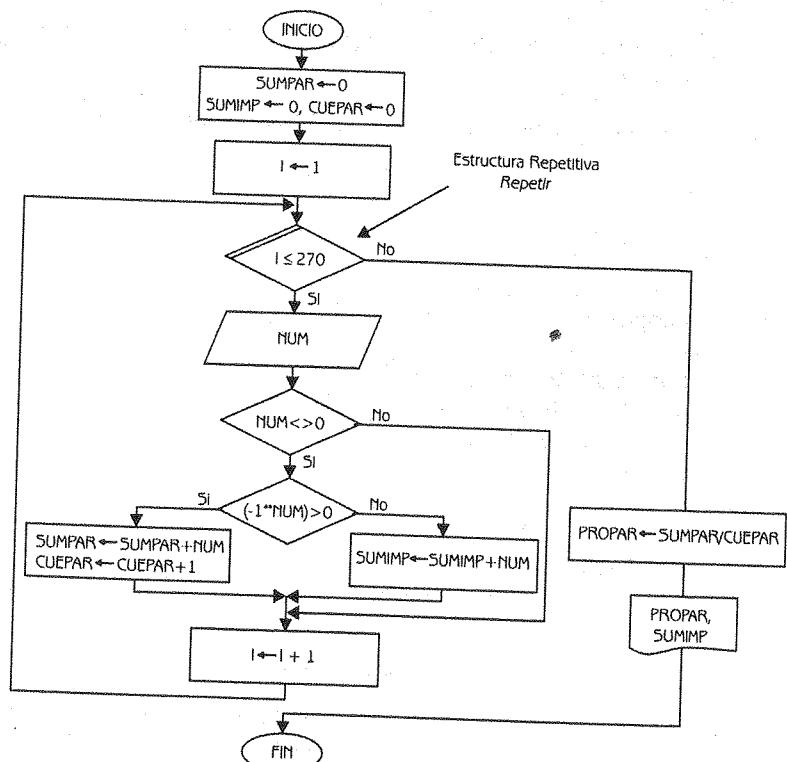


Diagrama de Flujo 3.8

**Explicación de las variables**

- I: Variable de tipo entero. Representa la variable de control del ciclo.
- NUM: Variable de tipo entero.
- SUMPAR y SUMIMP: Variables de tipo real. Acumulan los números pares e impares, respectivamente.
- CUEPAR: Variable de tipo entero. Cuenta el número de números pares.
- PROPAR: Variable de tipo real. Almacena el promedio de los números pares.

A continuación presentamos el diagrama de flujo en lenguaje algorítmico.

**Programa 3.7****PARES\_E\_IMPARES**

{El programa dadas 270 números enteros, obtiene la suma de los números impares y el promedio de los números pares}

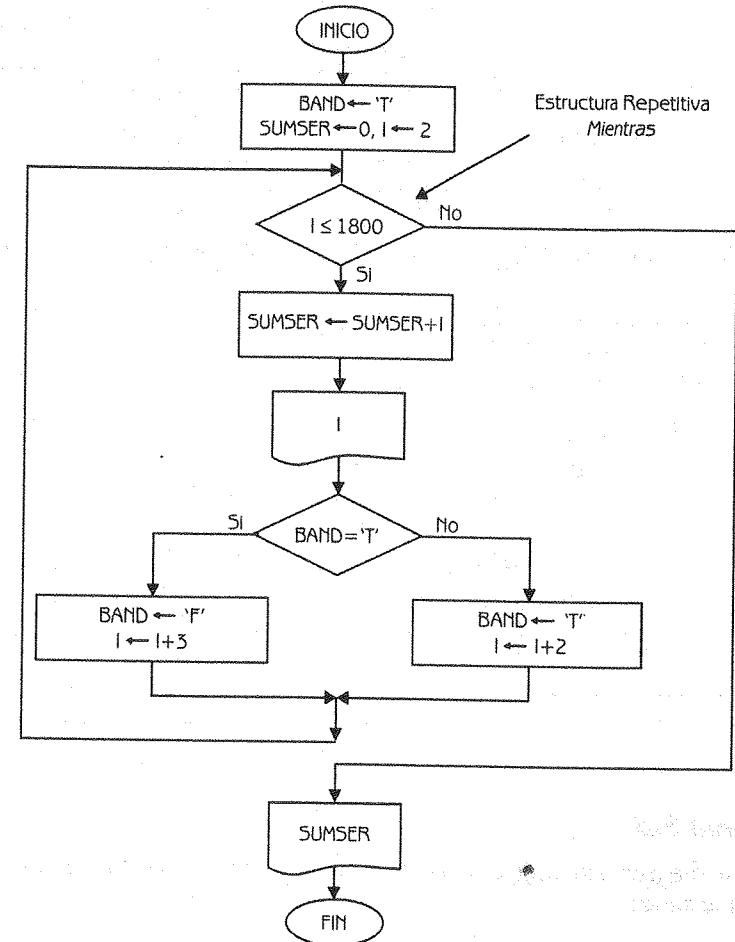
{I, NUM y CUEPAR son variables de tipo entero. SUMPAR, SUMIMP y PROPAR son variables de tipo real}

1. Hacer SUMPAR  $\leftarrow 0$ , SUMIMP  $\leftarrow 0$ , CUEPAR  $\leftarrow 0$  e I  $\leftarrow 1$
2. Repetir con I desde 1 hasta 270
  - Leer NUM
  - 2.1 Si (NUM  $\neq 0$ ) entonces
    - 2.1.1 Si  $(-1 \cdot \text{NUM}) > 0$  entonces
      - Hacer SUMPAR  $\leftarrow \text{SUMPAR} + \text{NUM}$  y CUEPAR  $\leftarrow \text{CUEPAR} + 1$
      - sino
        - Hacer SUMIMP  $\leftarrow \text{SUMIMP} + \text{NUM}$
    - 2.1.2 {Fin de condicional del paso 2.1.1}
  - 2.2 {Fin del condicional del paso 2.1}
  - Hacer I  $\leftarrow I + 1$
3. {Fin del ciclo del paso 2}
4. Hacer PROPAR  $\leftarrow \text{SUMPAR} / \text{CUEPAR}$
5. Escribir PROPAR y SUMIMP

**Problema 3.2**

Escriba un diagrama de flujo que obtenga la suma e imprima los términos de la siguiente serie:

2, 5, 7, 10, 12, 15, 17, ..., 1 800

**Diagrama de Flujo 3.9****Explicación de las variables**

- I: Variable de tipo entero. Se utiliza para incrementar el valor de los términos de la serie.
- SUMSER: Variable de tipo entero. Acumula los términos de la serie.
- BAND: Variable de tipo carácter. Es una variable auxiliar que se utiliza para indicar si al siguiente término de la serie hay que sumarle 3 o 2.

A continuación presentamos el diagrama de flujo en lenguaje algorítmico

**Programa 3.8****SUMA\_E\_IMPRIME\_SERIE**

{El programa Imprime y suma los términos de una serie}

{I y SUMSER son variables de tipo entero. BAND es una variable de tipo carácter}

1. Hacer SUMSER  $\leftarrow 0$ , BAND  $\leftarrow 'T'$  e I  $\leftarrow 2$ 2. Mientras ( $I \leq 1800$ ) Repetir    Hacer SUMSER  $\leftarrow$  SUMSER + I

Escribir I

2.1 Si BAND = 'T'

entonces

            Hacer BAND  $\leftarrow 'F'$  e I  $\leftarrow I + 3$ 

sino

            Hacer BAND  $\leftarrow 'T'$  e I  $\leftarrow I + 2$ 

2.2 {Fin del condicional del paso 2.1}

3. {Fin del ciclo del paso 2}

4. Escribir SUMSER

**Problema 3.3**

Escriba un diagrama de flujo que lea un número entero N y calcule el resultado de la siguiente serie:

$$1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots \pm \frac{1}{N}$$

Datos: N (variable de tipo entero que representa el número de términos de la serie).

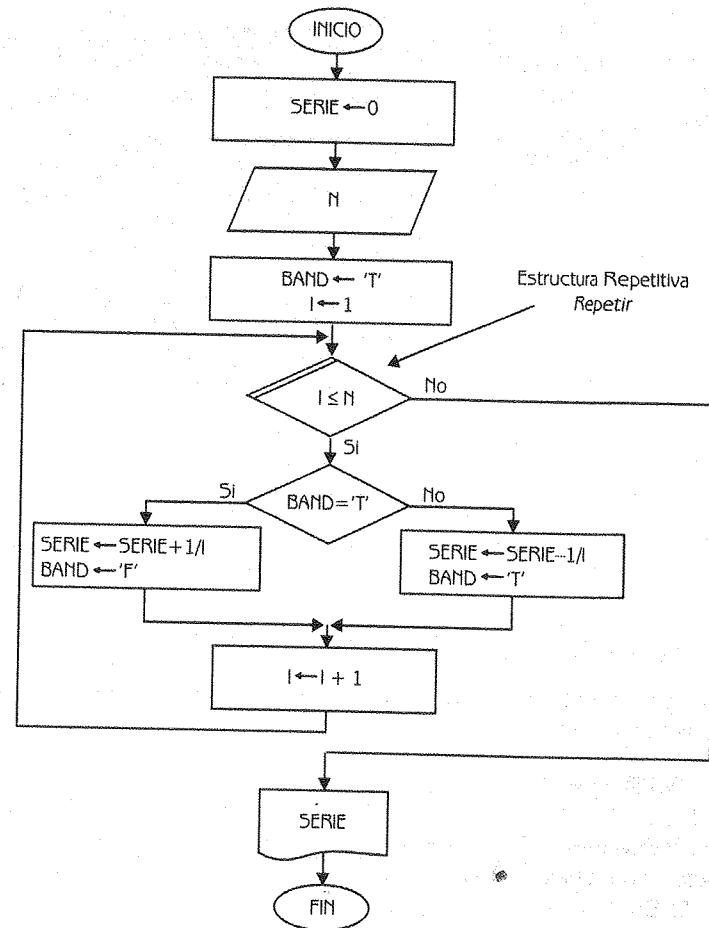


Diagrama de Flujo 3.10

**Explicación de las variables**

I: Variable de tipo entero. Representa la variable de control del ciclo.

N: Variable de tipo entero.

SERIE: Variable de tipo real. Acumula el resultado de los términos de la serie.

BAND: Variable de tipo carácter. Es una variable auxiliar que nos ayuda a determinar si debemos sumar o restar el término de la serie.

A continuación en la siguiente tabla podemos observar el comportamiento del algoritmo.

Inicia el Ciclo →

Fin del Ciclo →

Tabla 3.5			
I	N	BAND	SERIE
1	9	'T'	0
2		'F'	1
3		'T'	0.5
4		'F'	0.8333
5		'T'	0.5833
6		'F'	0.7833
7		'T'	0.6166
8		'F'	0.7594
9		'T'	0.6344
10		'F'	0.7456

: Expresa valores que se imprimen.

**Programa 3.9****SERIE\_TRES**

{El programa calcula el resultado de una determinada serie}

{I y N son variables de tipo entero. SERIE es una variable de tipo real. BAND es una variable de tipo carácter}

1. Hacer SERIE ← 0
2. Leer N
3. Hacer BAND ← 'T' e I ← 1
4. Repetir con I desde 1 hasta N
  - 4.1 Si BAND = 'T'  
entonces  
Hacer SERIE ← SERIE + 1 / I y BAND ← 'F'  
sino  
Hacer SERIE ← SERIE - 1 / I y BAND ← 'T'
  - 4.2 {Fin del condicional del paso 4.1}
  - Hacer I ← I + 1
5. {Fin del ciclo del paso 4}
- Escribir SERIE

**Problema 3.4**

Calcule el aumento de sueldo para un grupo de empleados de una empresa teniendo en cuenta el siguiente criterio:

**Problemas resueltos**

Si el sueldo es inferior a \$ 1 000

: Aumento 15%

Si el sueldo es mayor o igual a \$ 1 000

: Aumento 12%

Imprima el sueldo nuevo del trabajador y el total de nómina de la empresa, considerando este nuevo aumento.

Datos: SUE<sub>1</sub>, SUE<sub>2</sub>, ..., -1

Donde:

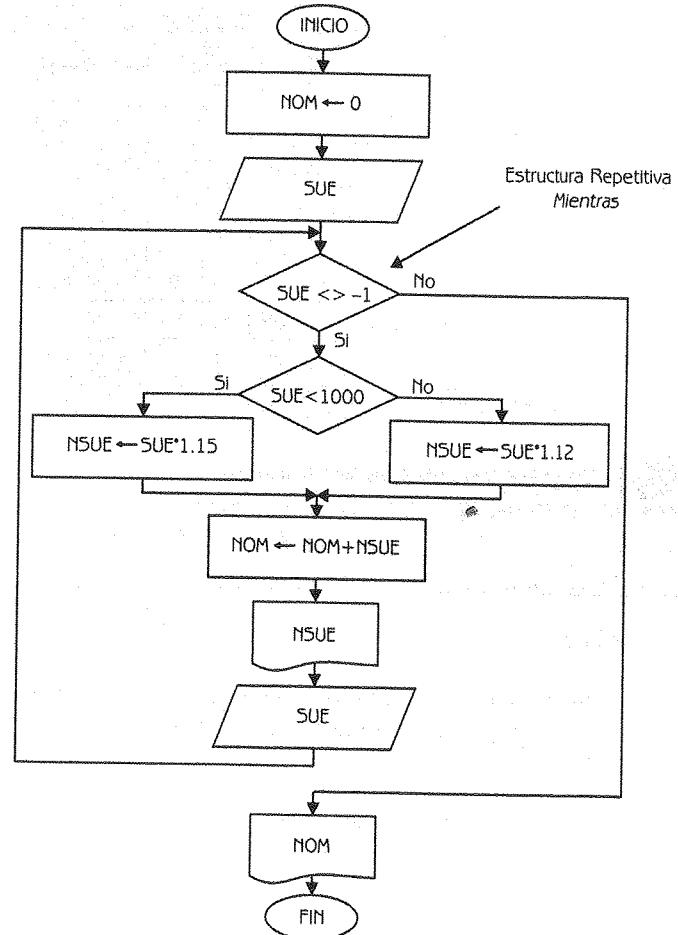
SUE<sub>i</sub> es una variable de tipo real que representa el sueldo del trabajador i. El fin de datos se expresa con -1.

Diagrama de Flujo 3.11

**Explicación de las variables**

**NOM:** Variable de tipo real. Acumula el nuevo sueldo de los trabajadores.

**SUE:** Variable de tipo real.

**NSUE:** Variable de tipo real. Expresa el nuevo sueldo del trabajador.

A continuación en la tabla 3.6, podemos observar el seguimiento del algoritmo.

Inicia el Ciclo →

Tabla 3.6		
NOM	SUE	NSUE
0	855	
983.25	760.32	983.25
1857.61	1100.20	874.56
3089.83	614	1232.22
3795.93	2600	706.10
6707.93	817.50	2912.00
7648.05	1280.30	940.12
9081.98	687	1433.93
9872.03	-1	790.05

Fin del Ciclo →

: Expresa valores que se imprimen.

**Programa 3.10****AUMENTO\_SUELDO**

{El programa, teniendo en cuenta ciertos criterios, calcula el aumento de sueldo para un grupo de trabajadores. Imprime el nuevo sueldo del trabajador y la nómina correspondiente}

{NOM, SUE y NSUE son variables de tipo real}

1. Hacer NOM ← 0
2. Leer SUE
3. Mientras (SUE < > -1) Repetir

**Problemas resueltos****3.1 Si SUE < 1000**

entonces

Hacer NSUE ← SUE \* 1.15

sino

Hacer NSUE ← SUE \* 1.12

**3.2 {Fin del condicional del paso 3.1}**

Hacer NOM ← NOM + NSUE

Escribir NSUE

Leer SUE

**4. {Fin del ciclo del paso 3}****5. Escribir NOM****Problema 3.5**

Dado N números enteros como dato haga un diagrama de flujo que:

- a) Obtenga cuántos números leídos fueron mayores que cero.
- b) Calcule el promedio de los números positivos.
- c) Obtenga el promedio de todos los números.

**Datos:** N, NUM<sub>1</sub>, NUM<sub>2</sub>, ..., NUM<sub>N</sub>

**Donde:**

N es una variable de tipo entero que representa el número de datos que vamos a ingresar.

NUM<sub>i</sub> es una variable de tipo entero que representa el número i que se ingresa ( $1 \leq i \leq N$ ).

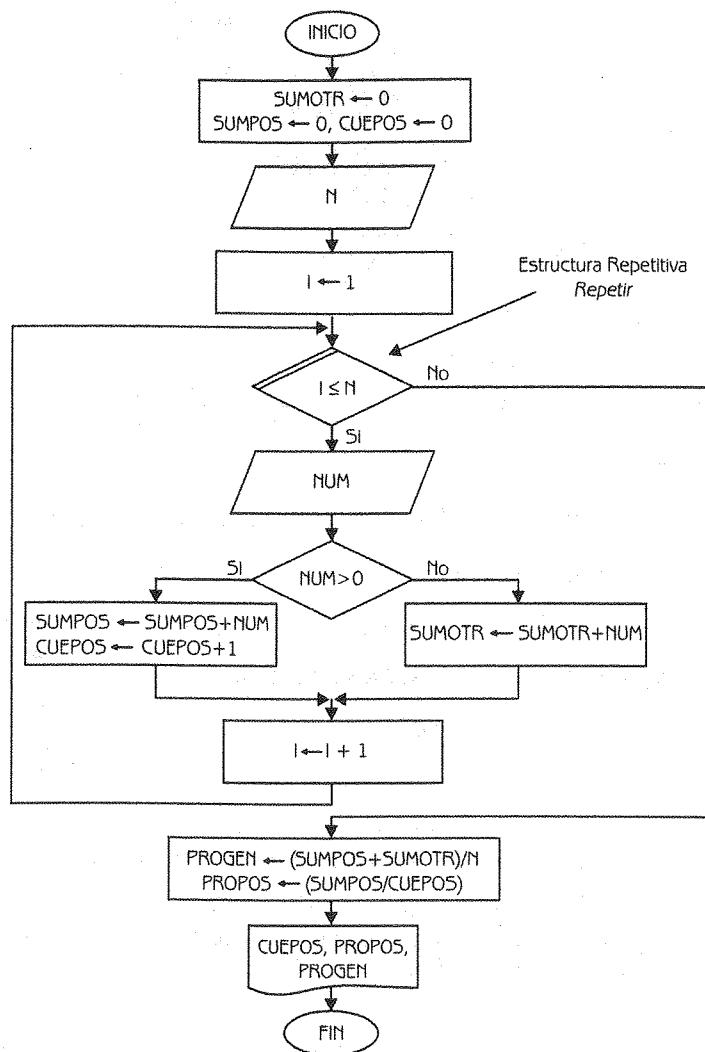


Diagrama de Flujo 3.12

## Explicación de las variables

- I: Variable de tipo entero. Representa la variable de control del ciclo.
- SUMPOS: Variable de tipo real. Acumula los números positivos.
- SUMOTR: Variable de tipo real. Acumula los números que no son positivos.
- CUEPOS: Variable de tipo entero. Cuenta los números positivos.
- N: Variable de tipo entero.

**NUM:** Variable de tipo entero.

**PROGEN:** Variable de tipo real. Almacena el promedio general de los números.

**PROPOS:** Variable de tipo real. Almacena el promedio de los números positivos.

A continuación en la siguiente tabla podemos observar el seguimiento del algoritmo.

Tabla 3.7

I	N	NUM	SUMPOS	CUEPOS	SUMOTR	PROGEN	PROPOS
1	20		0	0	0		
2		7	7	1			
3		12	19	2			
4		0				0	
5		6	25	3			
6		11	36	4			
7		-3				-3	
8		17	53	5			
9		12	65	6			
10		19	84	7			
11		0				-3	
12		11	95	8			
13		4	99	9			
14		-5				-8	
15		-1				-9	
16		0				-9	
17		3	102	10			
18		24	126	11			
19		0				-9	
20		3	129	12			
21		-16				-25	
						5.2	10.75

Fin del Ciclo →

: Expresa valores que se imprimen.

**Programa 3.11****PROMEDIO\_NUMEROS**

{El programa dado N números enteros como dato, obtiene el número de números positivos, el promedio de los números positivos y el promedio de todos los números}

{I, CUEPOS, NUM y N son variables de tipo entero. SUMPOS, SUMOTR, PROGEN y PROPOS son variables de tipo real.}

1. Hacer SUMPOS  $\leftarrow 0$ , CUEPOS  $\leftarrow 0$  y SUMOTR  $\leftarrow 0$
2. Leer N
3. Hacer I  $\leftarrow 1$
4. Repetir con I desde 1 hasta N
  - Leer NUM
  - 4.1 Si NUM > 0
    - entonces
      - Hacer SUMPOS  $\leftarrow$  SUMPOS + NUM y CUEPOS  $\leftarrow$  CUEPOS + 1
      - sino
        - Hacer SUMOTR  $\leftarrow$  SUMOTR + NUM
    - 4.2 {Fin del condicional del paso 4.1}
  - Hacer I  $\leftarrow I + 1$
5. {Fin del ciclo del paso 4}
6. Hacer PROGEN  $\leftarrow (SUMPOS + SUMOTR)/N$  y PROPOS  $\leftarrow SUMPOS/CUEPOS$
7. Escribir CUEPOS, PROPOS y PROGEN

**Problema 3.6**

Construya un diagrama de flujo tal que dado N números enteros como dato, calcule el mayor y el menor de estos números.

Datos: N, NUM<sub>1</sub>, NUM<sub>2</sub>, ..., NUM<sub>N</sub>

Donde:

N es una variable de tipo entero que indica el número de enteros que se ingresan.

NUM<sub>i</sub> es una variable de tipo entero que representa el número i que se ingresa ( $1 \leq i \leq N$ ).

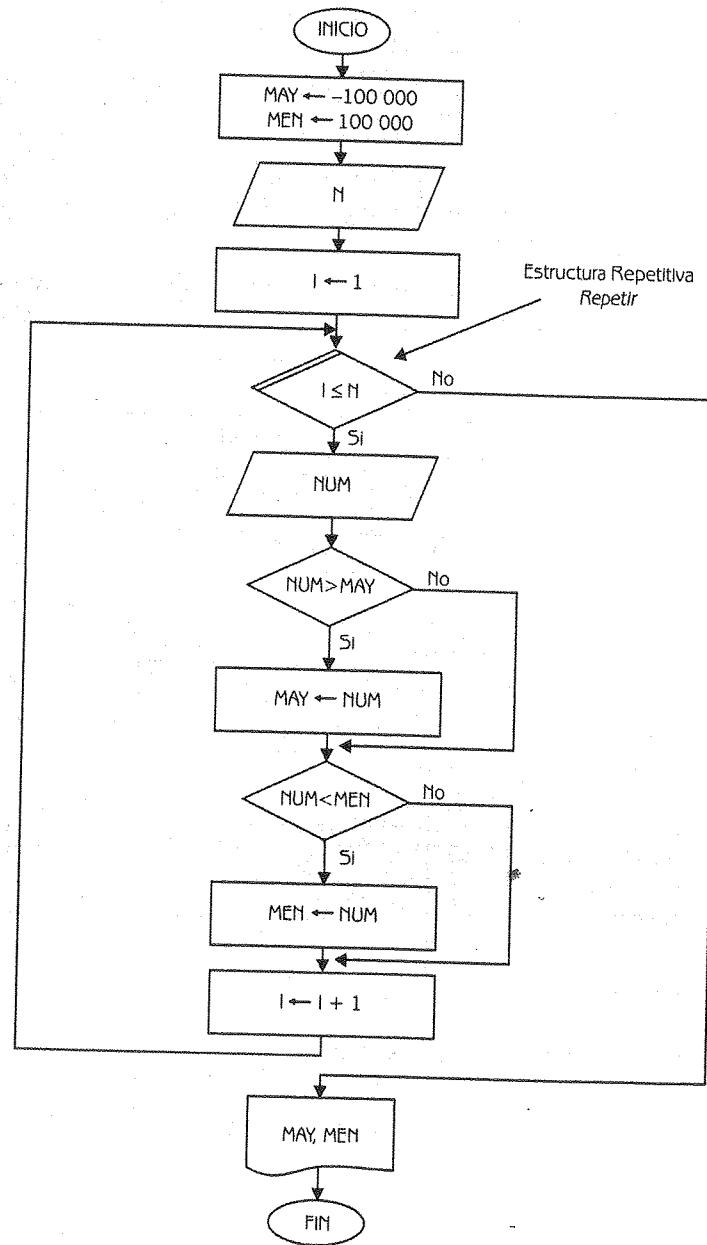


Diagrama de Flujo 3.13

**Explicación de las variables**

*I:* Variable de tipo entero. Representa la variable de control del ciclo.

*N y NUM:* Variables de tipo entero.

*MAY:* Variable de tipo entero. Almacena el máximo valor. Se inicializa con un valor negativo alto para que tome el valor del primer número que se ingresa.

*MEN:* Variable de tipo entero. Almacena el mínimo valor. Se inicializa con un valor alto para que tome el valor del primer número que se ingresa.

**Tabla 3.8**

I	N	NUM	MAY	MEN
1	12		-100 000	100 000
2		170	170	170
3		44		44
4		815	815	
5		1700	1700	
6		38		38
7		140		
8		380		
9		1020		
10		116		
11		14		14
12		730		
13		960		

Inicio del Ciclo →

Fin del Ciclo →

: Expresa valores que se imprimen.

A continuación presentamos el diagrama de flujo en lenguaje algorítmico.

**Programa 3.12****MAYOR\_Y\_MEMOR**

{El programa obtiene el mayor y menor de N números enteros que se ingresan}

{I, N y NUM son variables de tipo entero. MAY y MEN son variables de tipo real}

1. Hacer MAY ← -100 000 y MEN ← 100 000
2. Leer N
3. Hacer I ← 1
4. Repetir con I desde 1 hasta N
  - Leer NUM
  - 4.1 Si NUM > MAY entonces  
Hacer MAY ← NUM
  - 4.2 {Fin del condicional del paso 4.1}
  - 4.3 Si NUM < MEN entonces  
Hacer MEN ← NUM
  - 4.4 {Fin del condicional del paso 4.3}
  - Hacer I ← I + 1
5. {Fin del ciclo del paso 4}
6. Escribir MAY y MEN

**Problema 3.7**

Un vendedor ha hecho una serie de ventas y desea conocer aquellas de \$200 o menos, las mayores a \$200 pero inferiores a \$400, y el número de ventas de \$400 o superiores a tal cantidad. Haga un diagrama de flujo que le proporcione al vendedor esta información después de haber leído los datos de entrada.

Datos: N, V<sub>1</sub>, V<sub>2</sub>, ..., V<sub>N</sub>

Donde:

N es una variable de tipo entero que representa el número de ventas del vendedor.

V<sub>i</sub> es una variable de tipo real que indica la venta i del vendedor ( $1 \leq i \leq N$ ).

**Explicación de las variables**

*I:* Variable de tipo entero. Representa la variable de control del ciclo.

*CHI, MED y* Variables de tipo entero. Acumulan el número de ventas menores

*GRA:* a \$200, a \$400 y mayores a \$400, respectivamente.

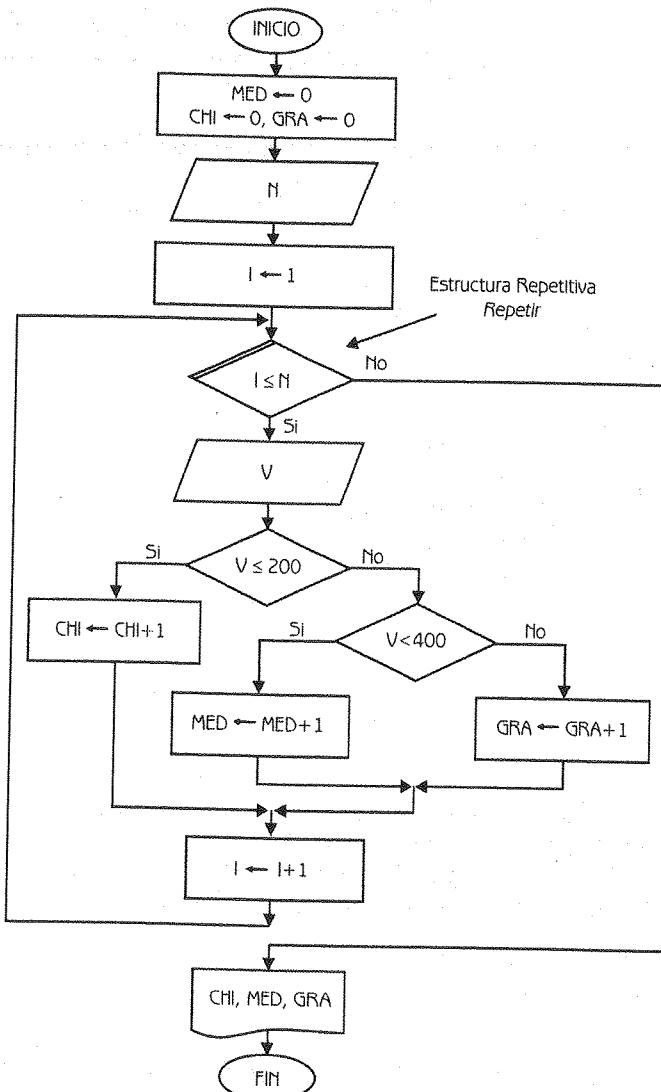


Diagrama de Flujo 3.14

N: Variable de tipo entero.  
V: Variable de tipo real.

A continuación en la tabla 3.9, podemos observar el seguimiento del algoritmo.

Tabla 3.9

I	N	MED	CHI	GRA	V
1	12	0	0	0	
2			1		180.25
3				1	470.30
4				2	150.25
5				3	88.60
6		1			230.15
7				4	170.20
8				5	40.30
9			2		201.90
10				6	60.32
11			3		280.30
12				7	15.70
13				8	140.20

Inicio del Ciclo →

Fin del Ciclo →

: Expresa valores que se imprimen.

**Programa 3.13****VENDEDOR**

{El programa dada información sobre las ventas que hizo un vendedor, obtiene cuantas ventas fueron chicas, medianas y grandes}

{I, CHI, MED, GRA y N son variables de tipo entero. V es una variable de tipo real}

1. Hacer CHI ← 0, MED ← 0 y GRA ← 0
2. Leer N
3. Hacer I ← 1
4. Repetir con I desde 1 hasta N  
 Leer V  
 4.1 Si  $V \leq 200$   
 entonces  
 Hacer CHI ← CHI + 1  
 sino  
 4.1.1 Si  $V < 400$   
 entonces  
 Hacer MED ← MED + 1  
 sino  
 Hacer GRA ← GRA + 1  
 4.1.2 {Fin del condicional del paso 4.1.1}  
 4.2 {Fin del condicional del paso 4.1}  
 Hacer I ← I + 1
5. {Fin del ciclo del paso 4}
6. Escribir CHI, MED y GRA

**Problema 3.8**

La siguiente se llama la conjetura de ULAM en honor del matemático S.Ulam:

- Comience con cualquier entero positivo
- Si es par, divídalo entre 2; si es impar, multiplíquelo por 3 y agréguele 1.
- Obtenga enteros sucesivamente repitiendo el proceso.

Al final, obtendrá el número 1, independientemente del entero inicial. Por ejemplo, cuando el entero inicial es 26, la secuencia será: 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1.

Construya un diagrama de flujo que lea un entero positivo y obtenga e imprima la sucesión de ULAM.

Dato: NUM (variable de tipo entero que representa el número que se ingresa).

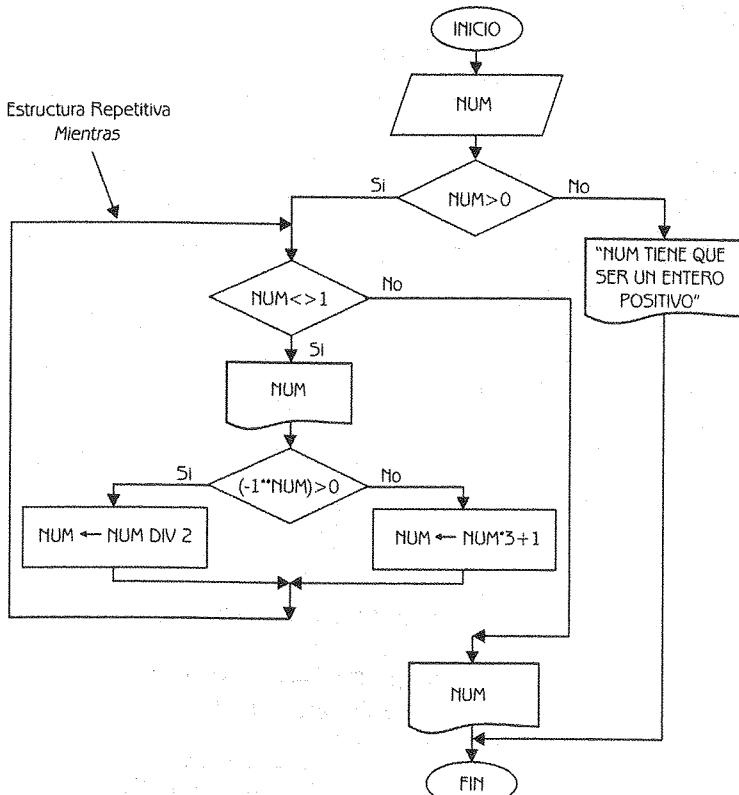


Diagrama de Flujo 3.15

**Explicación de las variables**

NUM: Variable de tipo entero.

A continuación en la siguiente tabla presentamos el seguimiento del algoritmo.

Tabla 3.10
NUM
25
76
38
19
58
29
88
44
22
11
34
17
52
26
13
40
20
10
5
16
8
4
2
1

Inicio del Ciclo →

Fin del Ciclo →

: Expresa valores que se imprimen.

**Programa 3.14****ULAM**

{El programa, dado un entero positivo, obtiene e imprime la sucesión de ULAM}

{NUM es una variable de tipo entero}

1. Leer NUM
2. Si NUM > 0 entonces
  - 2.1 Mientras (NUM < > 1) Repetir
 

Escribir NUM

    - 2.1.1 Si (-1 \*\* NUM) > 0 entonces
 

Hacer NUM ← NUM DIV 2

sino

Hacer NUM ← NUM \* 3 + 1
    - 2.1.2 {Fin del condicional del paso 2.1.1}
  - 2.2 {Fin del ciclo del paso 2.1}
 

Escribir NUM

sino

Escribir "NUM tiene que ser un entero positivo"
3. {Fin del condicional del paso 2}

**Problema 3.9**

Escriba un diagrama de flujo que lea un número entero N y calcule la suma de la siguiente serie:

$$1^1 + 2^2 + 3^3 + \dots + N^N$$

Dato: N (variable de tipo entero que representa el número de términos de la serie).

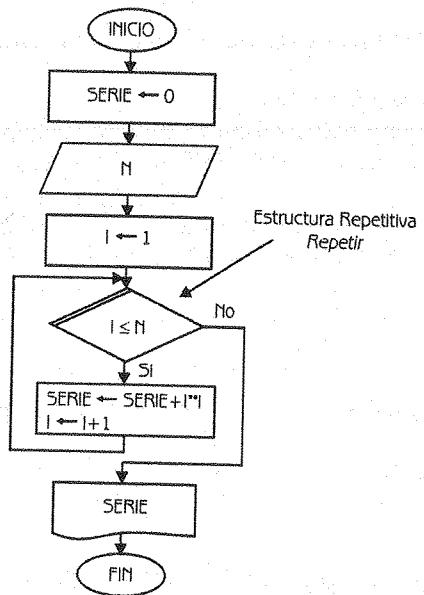


Diagrama de Flujo 3.16

**Explicación de las variables**

I: Variable de tipo entero. Representa la variable de control del ciclo.  
 N: Variable de tipo entero.

SERIE: Variable de tipo real. Es un acumulador. Acumula los resultados de cada término, obteniendo finalmente el resultado de la serie.

A continuación en la tabla 3.11 podemos observar el seguimiento del algoritmo.

Tabla 3.11		
I	N	SERIE
1	8	0
2		1
3		5
4		32
5		288
6		3413
7		50069
8		873612
9		17650828

Inicio del Ciclo →

Fin del Ciclo →

: Expresa el valor que se imprime.

**Programa 3.15****SERIE**

{El programa calcula la suma de una serie}

{I y N son variables de tipo entero. Serie es una variable de tipo real}

1. Hacer SERIE  $\leftarrow 0$
2. Leer N
3. Hacer I  $\leftarrow 1$
4. Repetir con I desde 1 hasta N  
    Hacer SERIE  $\leftarrow$  SERIE + I \* I e I  $\leftarrow$  I + 1
5. {Fin del ciclo del paso 4}
6. Escribir SERIE

**Problema 3.10**

Haga un diagrama de flujo que calcule el término número 180 de la secuencia FIBONACCI. Recuerde que los dos primeros números de la serie son 0 y 1. El resto se calcula como la suma de los dos números inmediatos que le preceden.

Ejemplo de la serie: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ... ,

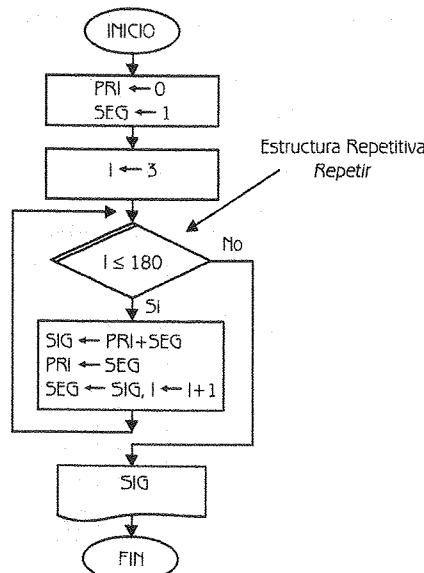


Diagrama de Flujo 3.17

**Explicación de las variables**

**I:** Variable de tipo entero. Representa la variable de control del ciclo. Normalmente se inicializa en 1, en este caso se inicializa en 3 puesto que existen dos asignaciones previas al comienzo del ciclo.

**PRI:** Variable de tipo entero. Representa al primero de los dos números que hay que sumar para obtener el siguiente número de la serie.

**SEG:** Variable de tipo entero. Representa al segundo de los dos números que hay que sumar para obtener el siguiente número de la serie.

**SIG:** Variable de tipo entero. Representa el siguiente número de la serie y se obtiene como la suma de (PRI + SEG)

En la siguiente tabla podemos observar el seguimiento del algoritmo. Consideremos además que en lugar de imprimir el término 680, calcularemos el término número 7.

Tabla 3.12			
I	PRI	SEG	SIG
3	0	1	
4	1	1	1
5	1	2	2
6	2	3	3
7	3	5	5
8	5	8	(8)

Inicio del Ciclo →

Fin del Ciclo →

○ : Expresa el valor que se imprime.

**Programa 3.16**

FIBONACCI\_680

{El programa calcula el término 180 de la secuencia FIBONACCI}

{I, PRI, SEG y SIG son variables de tipo entero}

1. Hacer  $PRI \leftarrow 0, SEG \leftarrow 1, I \leftarrow 3$
2. Repetir con  $I$  desde 3 hasta 180  
Hacer  $SIG \leftarrow PRI + SEG, PRI \leftarrow SEG, SEG \leftarrow SIG$  e  $I \leftarrow I + 1$
3. {Fin del ciclo del paso 2}
4. Escribir  $SIG$

**Problema 3.11**

Supóngase que en una reciente elección hubo cuatro candidatos (con identificadores 1, 2, 3, 4). Usted habrá de encontrar, mediante un programa, el número de votos correspondiente a cada candidato y el porcentaje que obtuvo respecto al total de los votantes. El usuario tecleará los votos de manera desorganizada, tal y como se obtuvieron en la elección, el final de datos está representado por un cero. Observe, como ejemplo, la siguiente lista:

1 3 1 4 2 2 1 4 1 1 1 2 1 3 1 4 0

Donde 1 representa un voto para el candidato 1; 3 un voto para el candidato 3; y así sucesivamente.

Datos:  $VOTO_1, VOTO_2, \dots, 0$

Donde:

$VOTO_i$ : es una variable de tipo entero que representa el voto para uno de los cuatro candidatos.

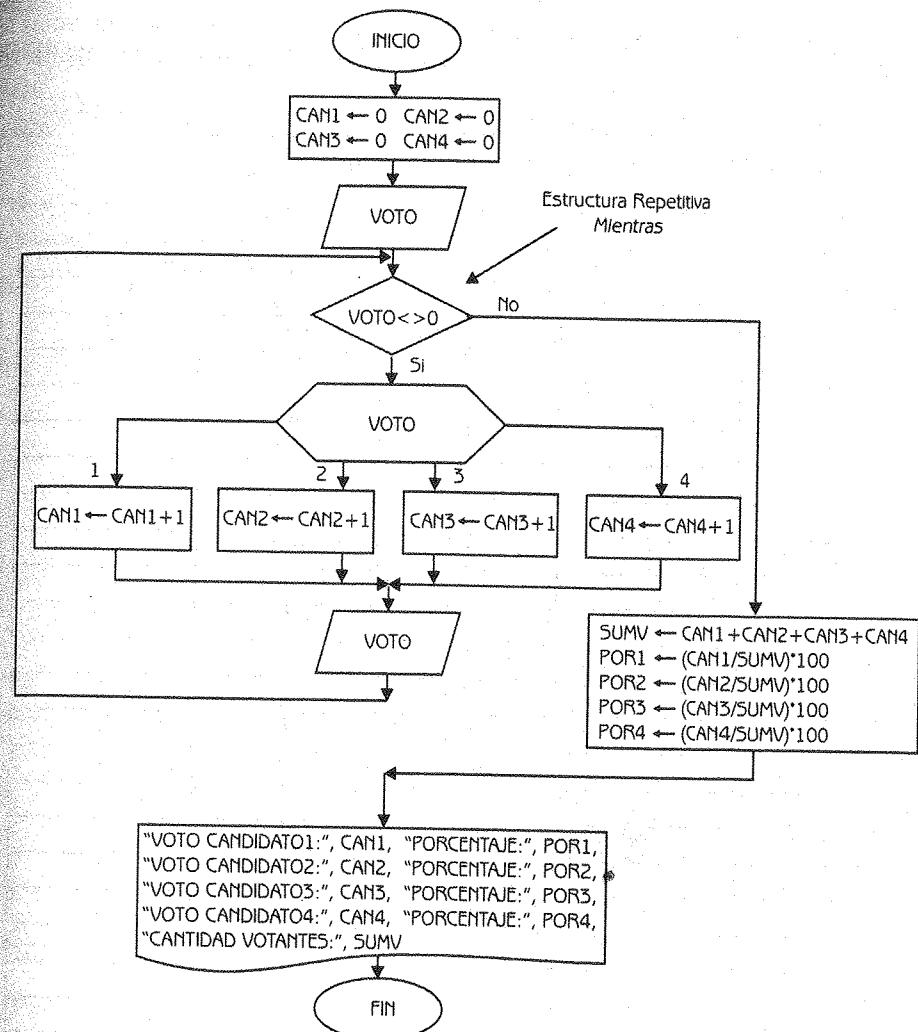


Diagrama de Flujo 3.18

**Explicación de las variables**

CAN1, CAN2, CAN3, CAN4: Variables de tipo real. Son acumuladores. Acumulan el número de votos de los candidatos 1, 2, 3, y 4, respectivamente.

VOTO: Variable de tipo entero.

SUMV: Variable de tipo real. Almacena el total de votos emitidos en la elección.

POR1, POR2, POR3, Variables de tipo real. Almacenan el porcentaje de votos  
 POR4: obtenidos por los candidatos 1, 2, 3 y 4, respectivamente.

A continuación en la siguiente tabla podemos observar el seguimiento del algoritmo.

VOTO	CAN1	CAN2	CAN3	CAN4	SUMV	POR1	POR2	POR3	POR4
1	0	0	0	0					
1	1								
2	2								
1		1							
3	3								
1			1						
1	4								
2	5								
4		2							
4			1						
3				2					
1			2						
1	6								
2	7								
3		3							
1				3					
4	8								
1				3					
2	9								
1			4						
2	10								
4		5							
4				4					
2					5				
1				6					
0	11								
					25	44.00	24.00	12.00	20.00

: Expresa valores que se imprimen.

### Programa 3.17

#### ELECCIONES

{El programa, obtiene el total de votos y el porcentaje obtenido por los cuatro candidatos de una elección}

{VOTO es una variable de tipo entero. POR1, POR2, POR3, POR4, CAN1, CAN2, CAN3, CAN4, y SUMV son variables de tipo real}

1. Hacer CAN1  $\leftarrow$  0, CAN2  $\leftarrow$  0, CAN3  $\leftarrow$  0 y CAN4  $\leftarrow$  0
2. Leer VOTO
3. Mientras (VOTO < > 0) Repetir
  - 3.1 Si VOTO Igual
    - 1 : Hacer CAN1  $\leftarrow$  CAN1 + 1
    - 2 : Hacer CAN2  $\leftarrow$  CAN2 + 1
    - 3 : Hacer CAN3  $\leftarrow$  CAN3 + 1
    - 4 : Hacer CAN4  $\leftarrow$  CAN4 + 1
  - 3.2 {Fin del condicional del paso 3.1}
- Leer VOTO
4. {Fin del ciclo del paso 3}
5. Hacer SUMV  $\leftarrow$  CAN1 + CAN2 + CAN3 + CAN4  
 POR1  $\leftarrow$  (CAN1/SUMV) \* 100  
 POR2  $\leftarrow$  (CAN2/SUMV) \* 100, POR3  $\leftarrow$  (CAN3/SUMV) \* 100 y  
 POR4  $\leftarrow$  (CAN4/SUMV) \* 100
6. Escribir "VOTOS CANDIDATO 1 : ", CAN1, "PORCENTAJE : ", POR1  
 "VOTOS CANDIDATO 2 : ", CAN2, "PORCENTAJE : ", POR2  
 "VOTOS CANDIDATO 3 : ", CAN3, "PORCENTAJE : ", POR3  
 "VOTOS CANDIDATO 4 : ", CAN4, "PORCENTAJE : ", POR4  
 "CANTIDAD VOTANTES : ", SUMV

#### Problema 3.12

De una empresa con N empleados, necesitamos obtener el número de empleado y sueldo del trabajador con el mayor sueldo de la empresa. Haga el diagrama de flujo correspondiente.

Datos: N, NUMEMP<sub>1</sub>, SUE<sub>1</sub>, NUMEMP<sub>2</sub>, SUE<sub>2</sub>, ..., NUMEMP<sub>N</sub>, SUE<sub>N</sub>

Donde:

N es una variable de tipo entero que representa el número de empleados de la empresa.

- NUMEMP<sub>i</sub>** es una variable de tipo entero que representa el número del empleado  $i$  ( $1 \leq i \leq N$ ).  
**SUE<sub>i</sub>** es una variable de tipo real que representa el sueldo del empleado  $i$  ( $1 \leq i \leq N$ ).

Nota sobre la variable **MASUE**:

Cuando se trata de localizar un máximo, la variable se inicializa con un valor mínimo. En la primera vuelta del ciclo, **MASUE** toma el valor del primer sueldo.

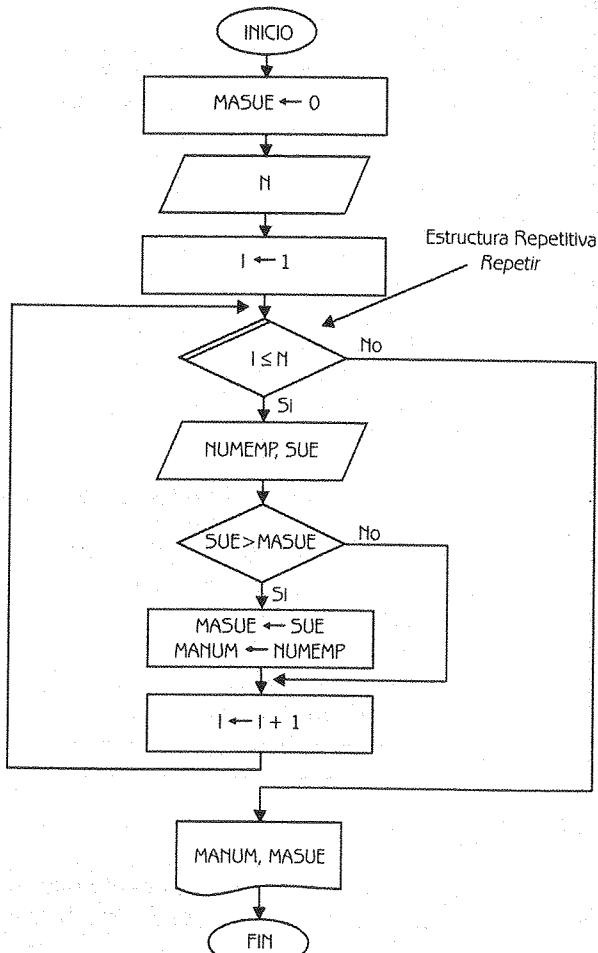


Diagrama de Flujo 3.19

### Explicación de las variables

**I:** Variable de tipo entero. Representa la variable de control del ciclo.

- N:** Variable de tipo entero.  
**NUMEMP:** Variable de tipo entero.  
**SUE:** Variable de tipo real.  
**MASUE:** Variable de tipo real. Almacena el mayor sueldo de los empleados.  
**MANUM:** Variable de tipo entero. Almacena el número del empleado que tenga el mayor sueldo.

En la tabla 3.14 podemos seguir el algoritmo para un valor de  $N = 6$ . Los datos que ingresaremos son los siguientes:

NUMEMP	SUE
1235	2580
1260	1400
1325	3600
1418	2720
1650	4120
1717	1950

Tabla 3.14					
I	N	NUMEMP	SUE	MASUE	MANUM
1	6			0	
2		1235	2580	2580	1235
3		1260	1400		
4		1325	3600	3600	1325
5		1418	2720		
6		1650	4120	4120	1650
7		1717	1950		

: Señala los valores que se imprimen.

**Programa 3.18****MAYOR\_SUELDO**

{El programa obtiene el número de empleado y el sueldo, del trabajador con mayor ingreso}

{I, N, NUMEMP y MANUM son variables de tipo entero. SUE y MASUE son variables de tipo real}

1. Hacer MASUE  $\leftarrow 0$
2. Leer N
3. Hacer I  $\leftarrow 1$
4. Repetir con I desde 1 hasta N  
    Leer NUMEMP y SUE  
    4.1 Si SUE > MASUE entonces  
        Hacer MASUE  $\leftarrow$  SUE y MANUM  $\leftarrow$  NUMEMP
- 4.2 {Fin del condicional del paso 4.2}  
    Hacer I  $\leftarrow$  I + 1
5. {Fin del ciclo del paso 4}
6. Escribir MANUM y MASUE

**Problema 3.13**

En el centro meteorológico de Argentina se llevan los promedios mensuales de las lluvias caídas en las principales regiones cerealeras del país. Existen 3 regiones importantes denominadas NORTE, CENTRO y SUR. Haga un diagrama de flujo para calcular lo siguiente:

- a) El promedio anual de la región centro.
- b) El mes y registro con menor lluvia en la región sur.
- c) La región con mayor lluvia anual (Considere que los registros anuales de las regiones son diferentes).

Datos: RNO<sub>1</sub>, RCE<sub>1</sub>, RSU<sub>1</sub>, ..., RNO<sub>12</sub>, RCE<sub>12</sub>, RSU<sub>12</sub>

Donde:

RNO<sub>i</sub>: es una variable de tipo real que indica la lluvia caída en la región norte en el mes i ( $1 \leq i \leq 12$ ).

RCE<sub>i</sub>: es una variable de tipo real que expresa la lluvia caída en la región centro en el mes i ( $1 \leq i \leq 12$ ).

RSU<sub>i</sub>: es una variable de tipo real que representa la lluvia caída en la región sur en el mes i ( $1 \leq i \leq 12$ ).

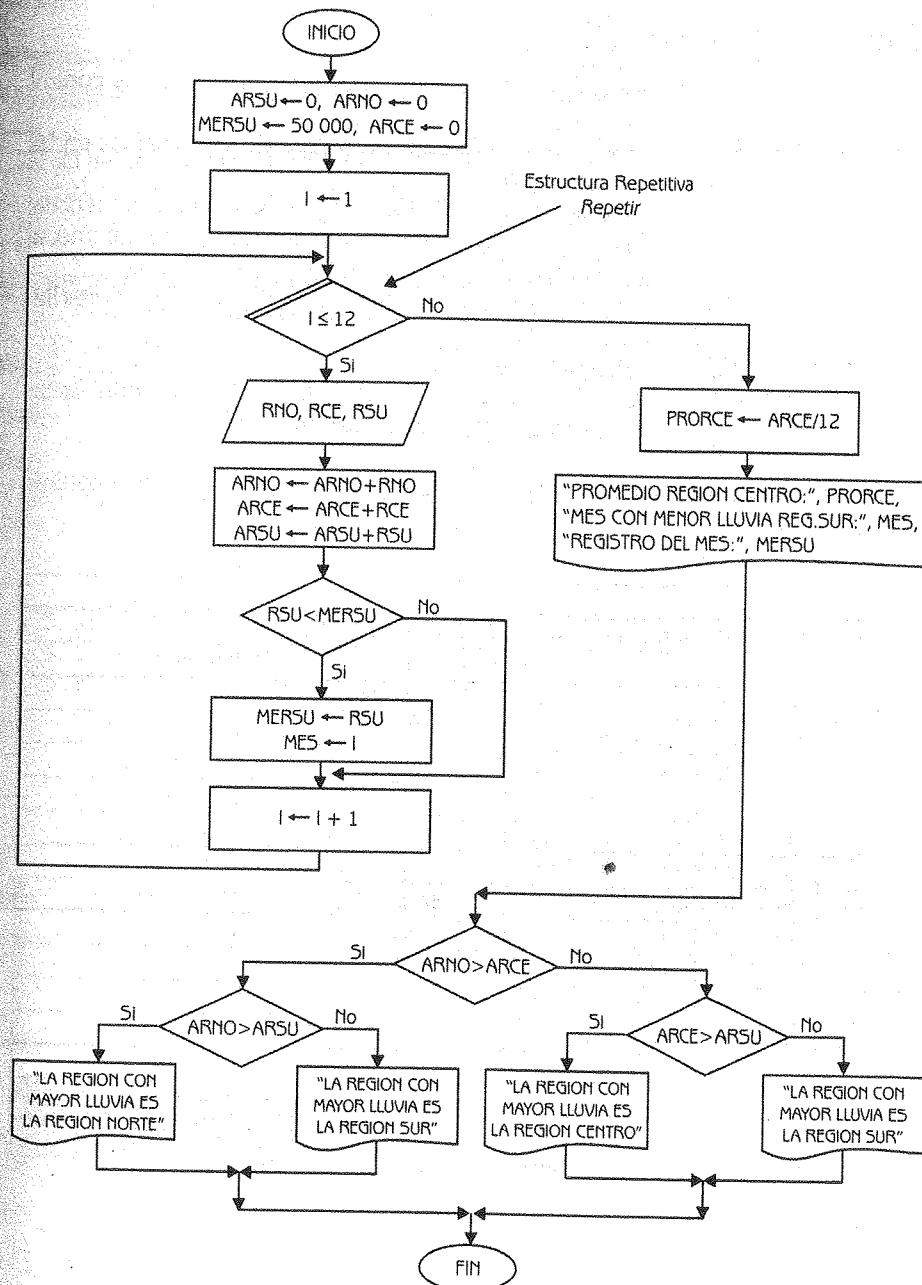
**Problemas resueltos**

Diagrama de Flujo 3.20

## Explicación de las variables

- I: Variable de tipo entero. Representa la variable de control del ciclo.
- ARNO, ARCE y ARSU:
- MERSU:
- MES:
- RNO, RCE y RSU:
- PRORCE:
- Variables de tipo real. Acumulan las lluvias caídas en las regiones norte, centro y sur.
- Variable de tipo real. Almacena el menor registro mensual de la región sur. Como se trata de localizar un mínimo, al principio se inicializa con un valor muy elevado.
- Variable de tipo entero. Almacena el mes con menores lluvias en la región sur.
- Variables de tipo real.
- Variable de tipo real. Almacena el promedio anual de las lluvias caídas en la región centro.

A continuación en la tabla 3.15 podemos observar el seguimiento del diagrama de flujo.

Tabla 3.15										
I	RNO	RCE	RSU	ARNO	ARCE	ARSU	MERSU	MES	PRORCE	
1				0	0	0	500 000			
2	80	70	170	80	70	170	170	1		
3	60	85	100	140	155	270	100	2		
4	120	100	185	260	255	455				
5	100	90	88	360	345	543	88	4		
6	70	82	80	430	427	623	80	5		
7	150	130	122	580	557	745				
8	100	95	67	680	652	812	67	7		
9	47	38	55	727	690	867	55	8		
10	95	115	110	822	805	977				
11	70	62	83	892	867	1060				
12	100	87	107	992	954	1167				
13	130	103	148	1122	1057	1315				
							88.08	"LA REGION CON MAYOR LLUVIA ES LA REGION SUR"		

: Expresa valores que se imprimen

## Programa 3.19

## LLUVIAS

{El programa dada información sobre las lluvias caídas en las principales zonas cे-realeras de Argentina, realiza cálculos estadísticos}

{I y MES son variables de tipo entero. ARNO, ARCE, ARSU, MERSU, RNO, RCE, RSU y PRORCE son variables de tipo real}

1. Hacer ARNO  $\leftarrow 0$ , ARCE  $\leftarrow 0$ , ARSU  $\leftarrow 0$ , MERSU  $\leftarrow 500\ 000$  e I  $\leftarrow 1$
2. Repetir con I desde 1 hasta 12
  - Leer RNO, RCE y RSU
  - Hacer ARNO  $\leftarrow$  ARNO + RNO, ARCE  $\leftarrow$  ARCE + RCE y ARSU  $\leftarrow$  ARSU + RSU
  - 2.1 Si RSU < MERSU entonces
    - Hacer MERSU  $\leftarrow$  RSU y MES  $\leftarrow$  I
  - 2.2. {Fin del condicional del paso 2.1}
  - Hacer I  $\leftarrow$  I + 1
3. {Fin del ciclo del paso 2}
4. Hacer PRORCE  $\leftarrow$  ARCE / 12
5. Escribir "PROMEDIO REGION : ", PRORCE,  
"MES COM MENOR LLUVIA REGION SUR : ", MES,  
"REGISTRO DEL MES : ", MERSU
6. Si ARNO > ARCE
  - entonces
    - 6.1 Si ARNO > ARSU
      - entonces
        - Escribir "LA REGION CON MAYOR LLUVIA ES LA NORTE"
        - sino
          - Escribir "LA REGION CON MAYOR LLUVIA ES LA SUR"
      - 6.2 {Fin del condicional del paso 6.1}
      - sino
    - 6.3 Si ARCE > ARSU
      - entonces
        - Escribir "LA REGION CON MAYOR LLUVIA ES LA CENTRO"
        - sino
          - Escribir "LA REGION CON MAYOR LLUVIA ES LA SUR"
  7. {Fin del condicional del paso 6}

**Problema 3.14**

En un estadio se tienen 5 tipos diferentes de localidades, las cuales se identifican por una clave numérica que es un valor comprendido entre 1 y 5. Los precios de cada localidad y los datos referentes a las ventas de boletos para el próximo juego se proporcionan como sigue:

Datos: P1, P2, P3, P4, P5

CLAVE<sub>1</sub>, CANT<sub>1</sub>  
CLAVE<sub>2</sub>, CANT<sub>2</sub>

... , ...

Donde:

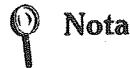
P1, P2, P3, P4 y P5 son variables de tipo real que representan los precios de las localidades 1, 2, 3, 4 y 5, respectivamente.

CLAVE<sub>i</sub> es una variable de tipo entero que representa el tipo de localidad de la venta i.

CANT<sub>i</sub> es una variable de tipo entero que indica la cantidad de boletos vendidos de un cierto tipo, en la venta i.

Construya un diagrama de flujo que:

- Lea los precios.
- Lea los datos de las ventas de boletos.
- Imprima para cada venta, la clave, la cantidad y el importe total de los boletos vendidos en esta venta.
- Calcule e imprima la cantidad de boletos vendidos de cada tipo.
- Calcule e imprima la recaudación total del estadio.

**Nota**

Considere que en una venta sólo se venden boletos de un tipo

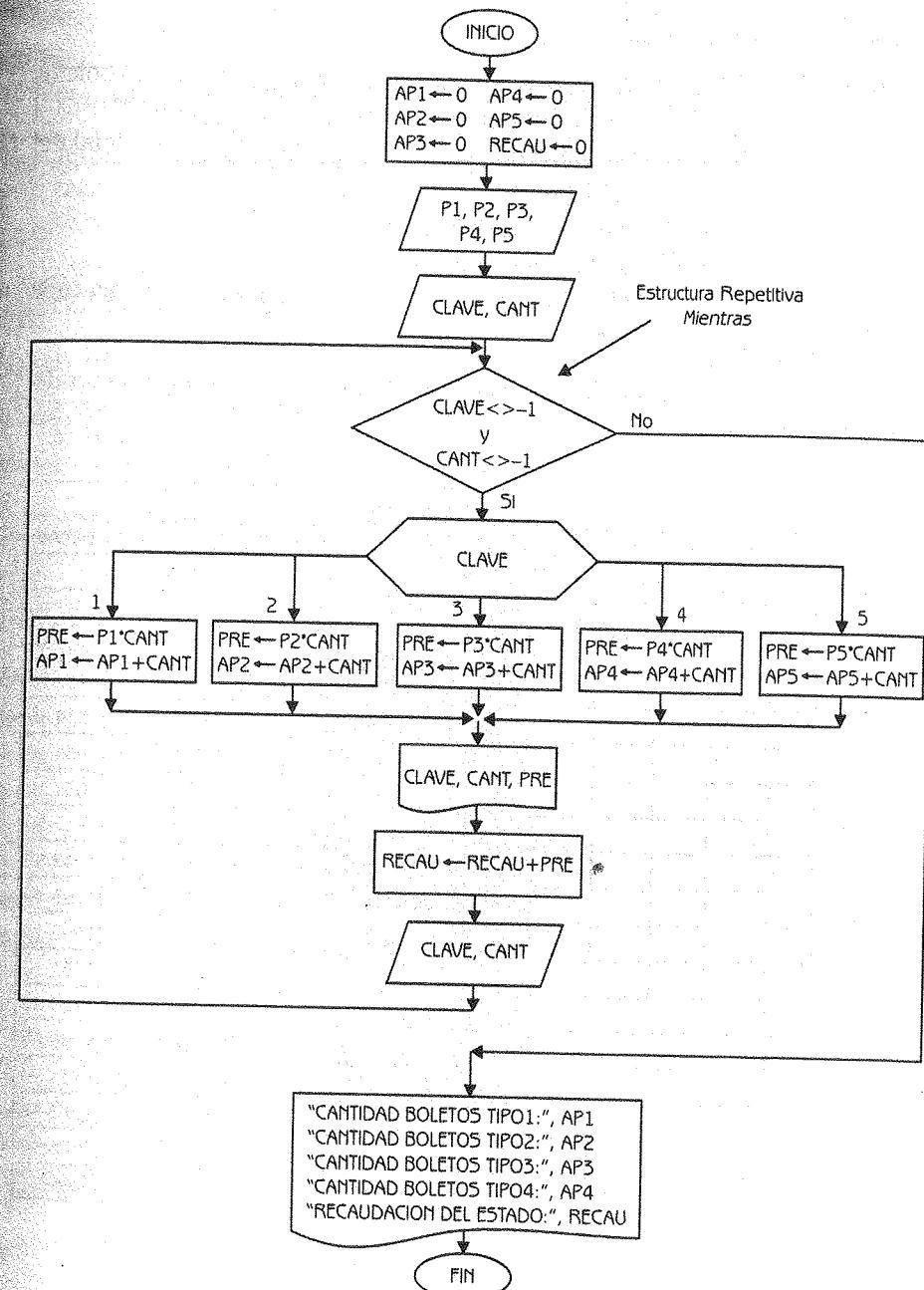


Diagrama de Flujo 3.21

**Explicación de las variables**

- AP1, AP2, AP3, AP4**: Variables de tipo entero. Acumulan el total de boletos vendidos del tipo 1, 2, 3, 4 y 5, respectivamente.
- RECAU**: Variable de tipo real. Acumula la recaudación total del estadio.
- P1, P2, P3, P4 y P5**: Variables de tipo real.
- CLAVE y CANT**: Variable de tipo entero.
- PRE**: Variable de tipo real. Almacena el total vendido en cada venta.

A continuación en la siguiente tabla, podemos observar el seguimiento del diagrama de flujo.

Tabla 3.16

P1	P2	P3	P4	P5	CLAVE	CANT	PRE	AP1	AP2	AP3	AP4	AP5	RECAU
7.25	15.80	25.00	50.00	75.00	2	3		0	0	0	0	0	0
					3	8	47.40		3				47.40
					2	4	200.00			8			247.40
					1	6	63.20		7				310.60
					4	5	43.5	6					354.10
					1	12	250.00			5			604.10
					2	8	87.00	18					691.10
					5	5	126.40		15				817.50
					2	7	225.00				3		1042.50
					3	14	110.60	22					1153.10
					4	1	550.00		22				1503.10
					2	11	50.00			6			1553.10
					4	9	173.80	35					1726.90
					5	7	450.00			15			2176.90
					1	23	525.00				10		2701.90
					1	18	166.75	41					2868.65
					5	4	130.50	59					2999.15
					-1	-1	100.00		26				3099.15

Fin del Ciclo →

: Expresa cantidades que se imprimen

**Programa 3.20****ESTADIO\_PRECIOS**

{El programa, dado los precios de distintas localidades y las cantidades vendidas de cada una de ellas en una determinada venta, obtiene el total vendido en esa venta, las cantidades de boletos vendidos de cada localidad y la recaudación total del estadio}

{AP1, AP2, AP3, AP4, AP5, CLAVE y CANT son variables de tipo entero. RECAU, P1, P2, P3, P4, PS y PRE son variables de tipo real}

- Hacer AP1 ← 0, AP2 ← 0, AP3 ← 0, AP4 ← 0, AP5 ← 0, y  
RECAU ← 0
- Leer P1, P2, P3, P4 y P5
- Leer CLAVE y CANT
- Mientras (CLAVE < > -1) y (CANT < > -1) Repetir
  - Si CLAVE Igual
    - 1 : Hacer PRE ← P1 \* CANT y AP1 ← AP1 + CANT
    - 2 : Hacer PRE ← P2 \* CANT y AP2 ← AP2 + CANT
    - 3 : Hacer PRE ← P3 \* CANT y AP3 ← AP3 + CANT
    - 4 : Hacer PRE ← P4 \* CANT y AP4 ← AP4 + CANT
    - 5 : Hacer PRE ← P5 \* CANT y AP5 ← AP5 + CANT
  - {Fin del condicional del paso 4.1}
 

Escribir CLAVE, CANT y PRE  
Hacer RECAU ← RECAU + PRE  
Leer CLAVE y CANT
- {Fin del ciclo del paso 4}
- Escribir "CANTIDAD BOLETOS TIPO1 : ", AP1,  
"CANTIDAD BOLETOS TIPO2 : ", AP2,  
"CANTIDAD BOLETOS TIPO3 : ", AP3,  
"CANTIDAD BOLETOS TIPO4 : ", AP4,  
"CANTIDAD BOLETOS TIPO5 : ", AP5,  
"RECAUDACION DEL ESTADIO : ", RECAU

**Problema 3.15**

Haga un diagrama de flujo para calcular lo que hay que pagar por un conjunto de llamadas telefónicas. Por cada llamada se ingresa el tipo (Internacional, Nacional, Local) y la duración en minutos. El criterio que se sigue para calcular el costo de cada llamada es el siguiente:

- Internacional**: 3 primeros minutos \$7.59  
Cada minuto adicional \$3.03
- Nacional**: 3 primeros minutos \$1.20  
Cada minuto adicional \$0.48

**Local:** Las primeras 50 llamadas no se cobran. Luego, cada llamada cuesta \$0.60

Datos: TIPO<sub>1</sub>, DUR<sub>1</sub>, TIPO<sub>2</sub>, DUR<sub>2</sub>, ..., X, -1

Donde:

TIPO<sub>i</sub> es una variable de tipo carácter que expresa el tipo de la llamada i. Toma el valor de 'I' si la llamada es internacional, 'N' si es nacional y 'L' si es local.

DUR<sub>i</sub> es una variable de tipo entero que indica la duración de la llamada i en minutos.

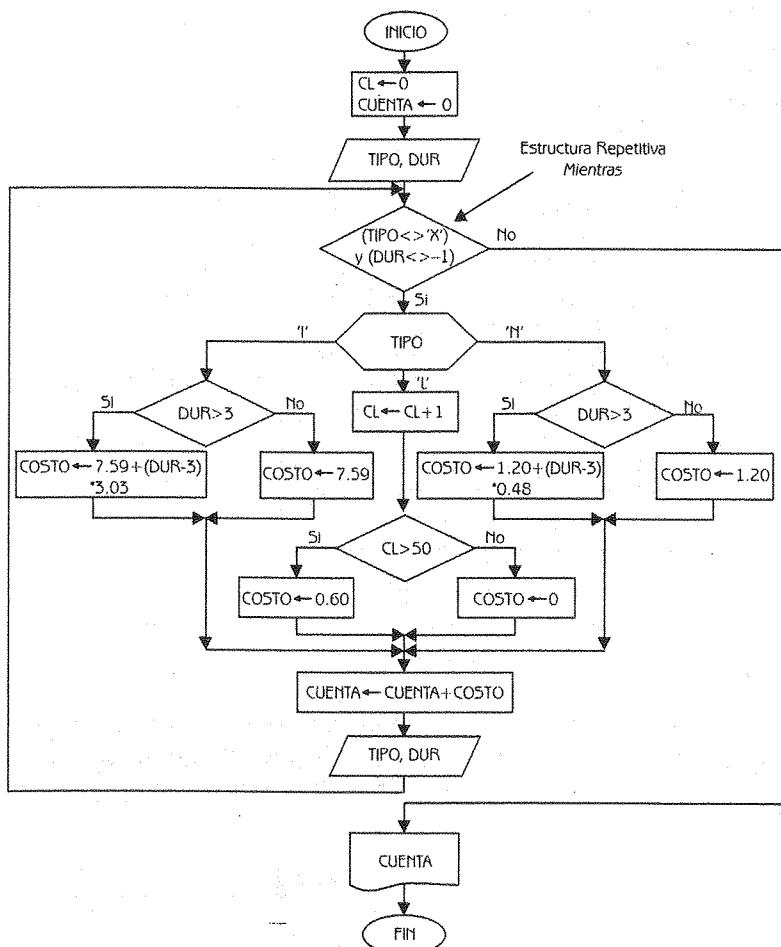


Diagrama de Flujo 3.22

### Explicación de las Variables

CL: Variable de tipo entero. Acumula el número de llamadas locales.

CUENTA: Variable de tipo real. Acumula el costo de cada llamada.

TIPO: Variable de tipo carácter.

DUR: Variable de tipo entero.

COSTO: Variable de tipo real. Almacena el costo de cada llamada.

En la siguiente tabla se puede observar el seguimiento del algoritmo para el siguiente conjunto de llamadas.

Datos: I, 7, N, 6, N, 12, L, 5, L, 7, L, 15, N, 16, L, 7, L, 6, L, 4, I, 11, X, -1

Tabla 3.17

TIPO	DUR	CL	COSTO	CUENTA
I	7	0		0
N	6		19.71	19.71
N	12		2.64	22.35
L	5		5.52	27.87
L	7	1	0	27.87
L	15	2	0	27.87
N	16	3	0	27.87
L	7		7.44	35.31
L	6	4	0	35.31
L	4	5	0	35.31
I	11	6	0	35.31
X	-1		31.83	67.14

Inicia el Ciclo →

Fin del Ciclo →

: Expresa el valor que se imprime.

**Programa 3.21****TELEFONO**

{El programa calcula el costo de un conjunto de llamadas telefónicas}

{CL y DUR son variables de tipo entero. CUENTA y COSTO son variables de tipo real. TIPO es una variable de tipo carácter}

1. Hacer CL  $\leftarrow 0$  y CUENTA  $\leftarrow 0$
2. Leer TIPO y DUR
3. Mientras (TIPO < > "X") y (DUR < > -1) Repetir
  - 3.1 Si TIPO Igual "I":
    - 3.1.1 Si DUR > 3 entonces Hacer COSTO  $\leftarrow 7.59 + ((DUR-3) * 3.03)$
    - sino Hacer COSTO  $\leftarrow 7.59$
  - 3.1.2 {Fin del condicional 3.1.1.}
 

"L": Hacer CL  $\leftarrow CL + 1$
  - 3.1.3 Si CL > 50 entonces Hacer COSTO  $\leftarrow 0.60$ 

sino Hacer COSTO  $\leftarrow 0$
  - 3.1.4 {Fin del condicional 3.1.3}
 

"N":
  - 3.1.5 Si DUR > 3 entonces Hacer COSTO  $\leftarrow 1.20 + ((DUR-3)*0.48)$ 

sino Hacer COSTO  $\leftarrow 1.20$
- 3.1.6 {Fin del condicional 3.1.5}
- 3.2 {Fin del condicional del paso 3.1}
 

Hacer CUENTA  $\leftarrow CUENTA + COSTO$   
Leer TIPO y DUR
4. {Fin del ciclo del paso 3}
5. Escribir CUENTA

**Problema 3.16**

En una bodega se tiene información sobre las cantidades producidas de cada tipo de vino, a lo largo de los últimos años. Haga un diagrama de flujo que calcule e imprima lo siguiente:

- El total producido de cada tipo de vino (son 5 tipos) a lo largo de los N años.
- El total producido de vino por año.
- Año en que se produjo la mayor cantidad de litros de vino del tipo 2. Imprimir también la cantidad de litros.
- Verificar si hubo algún año en el cual no se produjo el vino tipo 3. Si existe dicho año, imprimirlo.

Datos: N

$$\begin{aligned} V_{1,1}, V_{1,2}, \dots, V_{1,5} \\ V_{2,1}, V_{2,2}, \dots, V_{2,5} \\ \dots \\ V_{N,1}, V_{N,2}, \dots, V_{N,5} \end{aligned}$$

Donde:

N es una variable de tipo entero que representa el número de años, sobre los que se obtendrá la información pedida.

$V_{i,j}$  es una variable de tipo real que indica la cantidad de litros de vino del tipo j, producidos en el año i ( $1 \leq i \leq N, 1 \leq j \leq 5$ ).

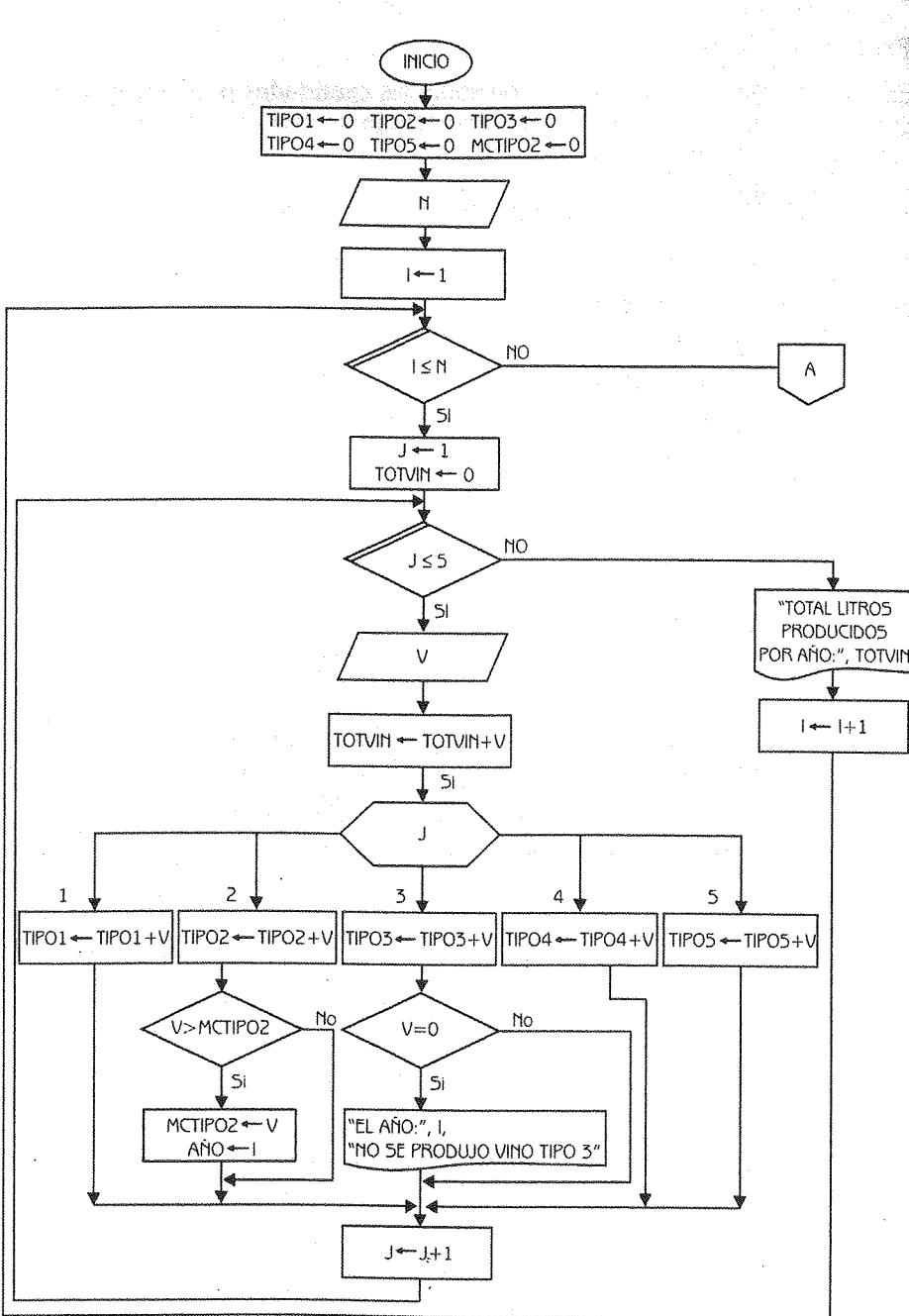


Diagrama de Flujo 3.23 (continúa)

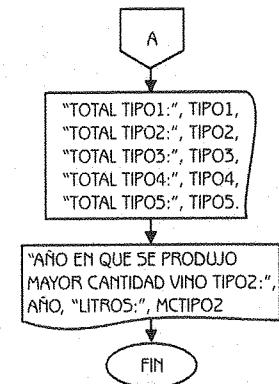


Diagrama de Flujo 3.23 (continúa)

**Explicación de las Variables**

- I:** Variable de tipo entero. Representa la variable de control del ciclo externo.
- TIPO1, TIPO2, TIPO3, TIPO4 y TIPO5:** Variables de tipo real. Acumulan el total de litros de vino producidos en los `N` años de los tipos 1,2,3,4 y 5, respectivamente.
- MCTIPO2:** Variable de tipo real. Acumula la cantidad de vino tipo 2 producido en el mayor año de producción. Localiza un máximo, por eso se inicializa con un valor mínimo (cero en este caso).
- AÑO:** Variable de tipo entero. Almacena el año en que se presenta la mayor producción, del vino tipo2.
- J:** Variable de tipo entero. Representa al contador del ciclo interno.
- N:** Variable de tipo entero.
- TOTVIN:** Variable de tipo real. Acumula el total de vinos producido en el año.
- V:** Variable de tipo real.

A continuación presentamos el programa correspondiente.

**Programa 3.22****BODEGA**

{El programa dada información sobre los vinos producidos por una bodega, obtiene información estadística de utilidad}

{I, AÑO, J y N son variables de tipo entero. TIPO1, TIPO2, TIPO3, TIPO4, MCTIPO2, TOTVIN y V son variables de tipo real}

1. Hacer TIPO1  $\leftarrow 0$ , TIPO2  $\leftarrow 0$ , TIPO3  $\leftarrow 0$ , TIPO4  $\leftarrow 0$ , TIPO5  $\leftarrow 0$   
y MCTIPO2  $\leftarrow 0$
2. Leer N
3. Hacer I  $\leftarrow 1$
4. Repetir con I desde 1 hasta N  
 Hacer TOTVIN  $\leftarrow 0$  y J  $\leftarrow 1$ 
  - 4.1 Repetir con J desde 1 hasta 5  
 Leer V  
 Hacer TOTVIN  $\leftarrow$  TOTVIN + V
    - 4.1.1 Si J Igual  
 1 : Hacer TIPO1  $\leftarrow$  TIPO1 + V  
 2 : Hacer TIPO2  $\leftarrow$  TIPO2 + V
      - 4.1.1.1 Si V > MCTIPO2 entonces  
 Hacer MCTIPO2  $\leftarrow$  V y AÑO  $\leftarrow$  I
      - 4.1.1.2 {Fin del condicional del paso 4.1.1.1}
    - 3 : Hacer TIPO3  $\leftarrow$  TIPO3 + V
      - 4.1.1.3 Si (V=0) entonces  
 Escribir "En el año : ", I, " No se produjo vino TIPO3 "
      - 4.1.1.4 {Fin del condicional del paso 4.1.1.3}
  - 4 : Hacer TIPO4  $\leftarrow$  TIPO4 + V
  - 5 : Hacer TIPO5  $\leftarrow$  TIPO5 + V
    - 4.1.2 {Fin del condicional del paso 4.1.1}
    - Hacer J  $\leftarrow$  J + 1
- 4.2 {Fin del ciclo del paso 4.1}  
 Escribir "Total de litros producidos por año: ", TOTVIN  
 Hacer I  $\leftarrow$  I + 1
5. {Fin del ciclo del paso 4}
6. Escribir " TOTAL TIPO1 : ", TIPO1,  
 " TOTAL TIPO2 : ", TIPO2,  
 " TOTAL TIPO3 : ", TIPO3,  
 " TOTAL TIPO4 : ", TIPO4,  
 " TOTAL TIPO5 : ", TIPO5,
7. Escribir "Año en que se produjo mayor cantidad vino TIPO2 : ", AÑO,  
 " LITROS : ", MCTIPO2

### Problema 3.17

Se dice que un entero positivo N es un número primo si los únicos enteros positivos que lo dividen son exactamente 1 y N. Diseñe un diagrama de flujo que lea un número M, y obtenga y cuente todos los números primos menores a M.

Dato: M (variable de tipo entero que representa el número límite que se ingresa).

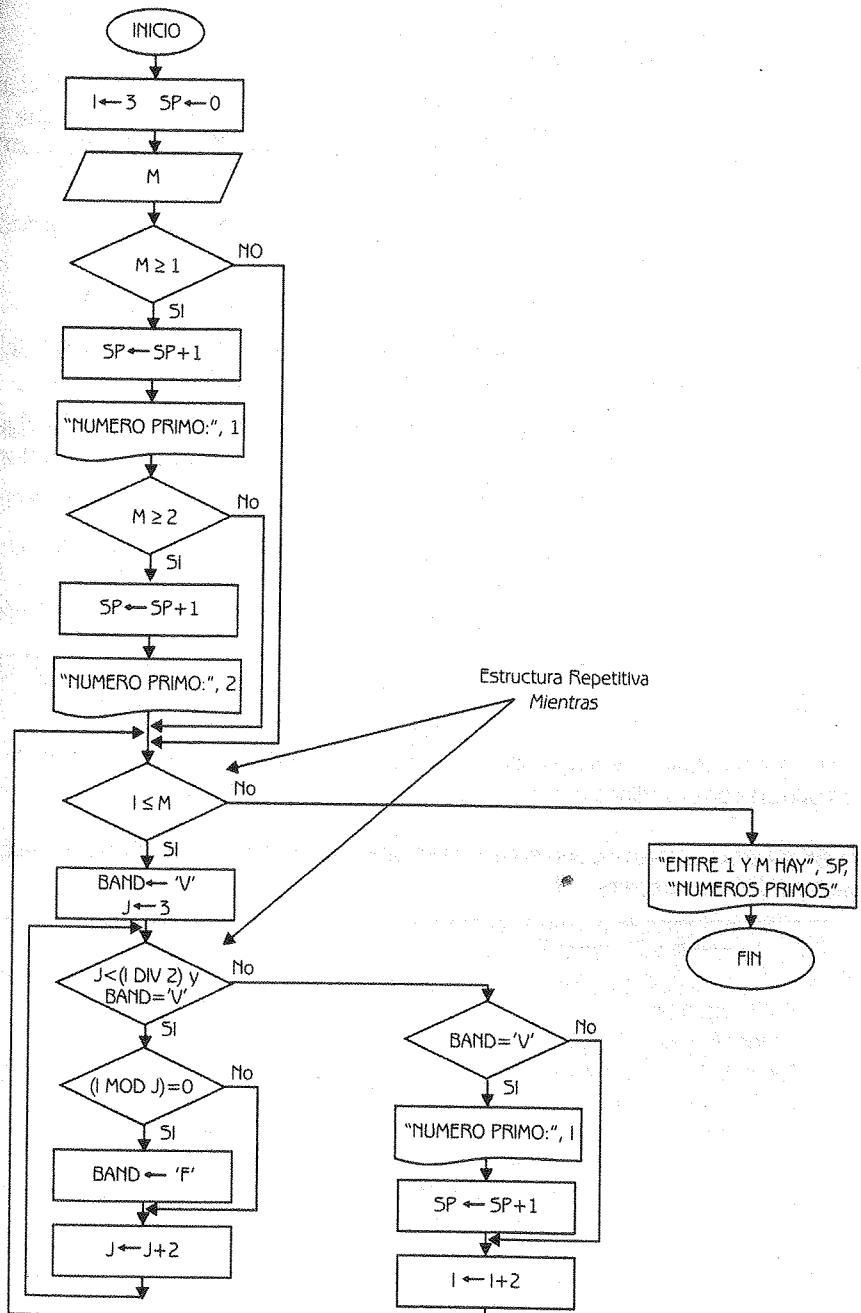


Diagrama de Flujo 3.24

**Explicación de las variables**

- I:* Variable de tipo entero. Representa la variable de control del ciclo externo. Se inicializa en 3 porque sabemos por definición que el número 1 y los números pares (con excepción del 2) no son primos, y se incrementa de 2 en 2.
- SP:* Variable de tipo entero. Acumula el número de números primos comprendidos entre 1 y M.
- M:* Variable de tipo real.
- J:* Variable de tipo entero. Representa la variable de control del ciclo interno. Se inicializa en 3 y se incrementa de 2 en 2.
- BAND:* Variable de tipo carácter. Se inicializa en verdadero ('V'). Si alguna de las divisiones que se realizan produce módulo cero, entonces toma el valor de falso ('F').

A continuación, presentamos el programa correspondiente.

**Programa 3.23****PRIMOS**

{El programa, dado un entero positivo M, obtiene e imprime la cantidad de números primos comprendidos entre 1 y M}

{I, SP y J son variables de tipo entero. M es una variable de tipo real. BAND es una variable de tipo carácter}

1. Hacer I  $\leftarrow 3$  y SP  $\leftarrow 0$
2. Leer M
3. Si  $M \geq 1$  entonces
  - Hacer SP  $\leftarrow SP + 1$
  - Escribir "Número Primo:", 1
- 3.1 Si  $M > 2$  entonces
  - Hacer SP  $\leftarrow SP + 1$
  - Escribir "Número Primo:", 2
- 3.2 {Fin del condicional del paso 3.1}
4. {Fin del condicional del paso 3}
5. Mientras  $I \leq M$  Repetir
  - Hacer BAND  $\leftarrow 'V'$  y J  $\leftarrow 3$
  - 5.1 Mientras ( $J < (I \text{ DIV } 2)$ ) y (BAND='V') Repetir

**Problemas resueltos**

- 5.1.1 Si  $(I \text{ MOD } J)=0$  entonces
  - Hacer BAND  $\leftarrow 'F'$
- 5.1.2 {Fin del condicional del paso 5.1.1}
  - Hacer J  $\leftarrow J+2$
- 5.2 {Fin del ciclo del paso 5.1}
- 5.3 Si (BAND='V') entonces
  - Escribir "Número primo:", I
  - Hacer SP  $\leftarrow SP + 1$
- 5.4 {Fin del condicional del paso 5.3}
  - Hacer I  $\leftarrow I+2$
6. {Fin del ciclo del paso 5}
4. Escribe "ENTRE 1 y M hay", SP, "NUMEROS PRIMOS"

**Problema 3.18**

Los datos reunidos en la Secretaría de Industrias relacionado a la producción de N fábricas ( $N \leq 1000$ ) en cada uno de los meses del año anterior, se proporcionan de la siguiente forma:

Datos: N

FABRICA<sub>1</sub>, MES<sub>1,1</sub>, MES<sub>1,2</sub>, ..., MES<sub>1,12</sub>

FABRICA<sub>2</sub>, MES<sub>2,1</sub>, MES<sub>2,2</sub>, ..., MES<sub>2,12</sub>

FABRICA<sub>N</sub>, MES<sub>N,1</sub>, MES<sub>N,2</sub>, ..., MES<sub>N,12</sub>

Donde:

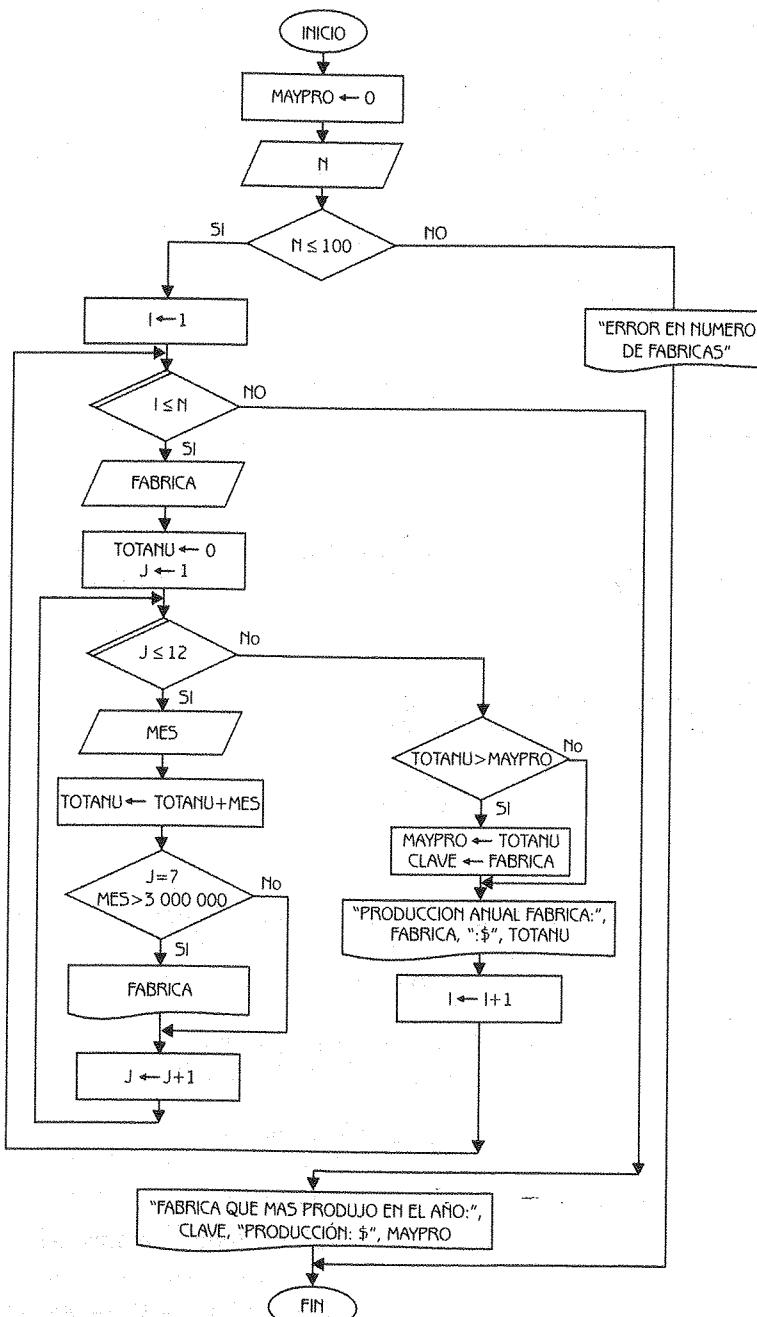
N es una variable de tipo entero que indica el número de fábricas registradas en la Secretaría de Industrias ( $N \leq 100$ ).

FABRICA<sub>i</sub> es una variable de tipo entero que indica la clave que identifica a la fábrica<sub>i</sub> ( $1 \leq i \leq N$ ).

MES<sub>i,j</sub> es una variable de tipo real que representa la producción de la fábrica<sub>i</sub> en el mes<sub>j</sub> ( $1 \leq i \leq N, 1 \leq j \leq 12$ ).

Haga un diagrama de flujo que calcule lo siguiente:

- a) Los totales anuales de producción de cada fábrica.
- b) La clave de la fábrica que más produjo en el año. Indicar también el total de la producción.
- c) Imprimir las claves de las fábricas cuyas producciones en el mes de julio superaron los \$ 3 000 000.



### Diagrama de Fluo 3.25

## **Explicación de las variables**

- I: Variable de tipo entero. Representa la variable de control del ciclo externo.
  - O: Variable de tipo real. Almacena el total de la fábrica que más produjo en el año. Se utiliza para localizar un máximo, por eso se inicializa en cero.
  - E: Variable de tipo entero. Almacena la clave de la fábrica que más dinero produjo en el año.
  - J: Variable de tipo entero. Representa la variable de control del ciclo interno.
  - U: Variable de tipo real. Acumula lo que vendió la fábrica en el año.
  - A: Variables de tipo entero.
  - S: Variable de tipo real.

### Programa 3.24

SECRETARIA DE INDUSTRIAS

{El programa, dada cierta información que recibe de la Secretaría de Industrias relacionado a la producción de N fábricas, obtiene información estadística de la misma}

{I, CLAVE, J, N y FABRICA son variables de tipo entero. MAYPRO, TOTANU y MES son variables de tipo real}

1. Hacer MAYPRO  $\leftarrow$  0
  2. Leer N
  3. Si  $N \leq 1000$   
entonces  
    Hacer I  $\leftarrow$  1  
    3.1 Repetir con I desde 1 hasta N  
        Leer FABRICA  
        Hacer TOTANU  $\leftarrow$  0 y J  $\leftarrow$  1  
        3.1.1 Repetir con J desde 1 hasta 12  
            Leer MES  
            Hacer TOTANU  $\leftarrow$  TOTANU + MES

3.1.1.1 Si ( $J = 7$ ) y ( $MES > 3\ 000\ 000$ ) entonces  
     Escribir FABRICA  
 3.1.1.2 {Fin del condicional del paso 3.1.1.1}  
     Hacer  $J \leftarrow J + 1$   
 3.1.2 {Fin del ciclo del paso 3.1.1}  
 3.1.3 Si TOTANU > MAYPRO entonces  
     Hacer MAYPRO  $\leftarrow$  TOTANU y CLAVE  $\leftarrow$  FABRICA  
 3.1.4 {Fin del condicional del paso 3.1.3}  
     Escribir "Producción anual fábrica : ", FABRICA,  
     ": \$ ", TOTANU  
     Hacer I  $\leftarrow I + 1$   
 3.2 {Fin del ciclo del paso 3.1}  
     Escribir "FABRICA que más produjo en el año : ", CLAVE,  
     " PRODUCCION : \$ ", MAYPRO  
 sino  
     Escribir "ERROR EN NUMERO DE FABRICAS"  
 4. {Fin del condicional del paso 3}

### Problema 3.19

Un número es perfecto si “la suma de sus divisores excepto el mismo es igual al propio número”. Haga un diagrama de flujo para calcular e imprimir los números perfectos menores o iguales que N.

Dato: N (variable de tipo entero que representa el límite de los números naturales enteros que se probarán para determinar si son perfectos).

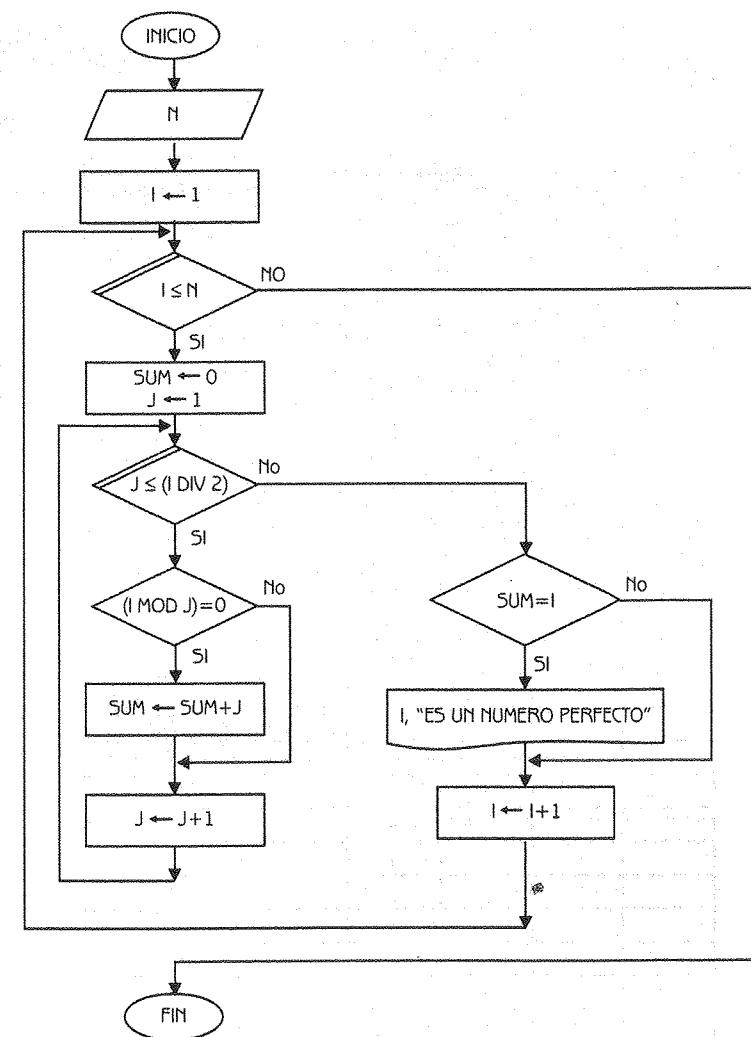


Diagrama de Flujo 3.26

### Explicación de las variables

- I: Es una variable de tipo entero. Representa la variable de control del ciclo externo.
- N: Variable de tipo entero.
- SUM: Variable de tipo entero. Acumula los divisores para posteriormente probar si el número es perfecto.
- J: Variable de tipo entero. Representa la variable de control del ciclo interno.

A continuación en la siguiente tabla, podemos observar el seguimiento del diagrama de flujo:

Tabla 3.18				
M	I	SUM	J	(IMPRESION)
9	1			
	2	0	1	
		0	1	
3	1	2		
		0	1	
4	1	2		
		0	1	
		1	2	
5	3	3		
		0	1	
		1	2	
6		3		
		0	1	
		1	2	
		3	3	
7	6	4	6 ES UN NUMERO PERFECTO	
		0	1	
		1	2	
		3		
8		4		
		0	1	
		1	2	
		3	3	
		7	4	
9		5		
		0	1	
		1	2	
		4	3	
			4	
10		5		

: Expresa valores que se imprimen.

## Programa 3.25

## NUMEROS\_PERFECTOS

{El programa dado un número entero N, obtiene e imprime los números perfectos comprendidos entre 1 y N}

{I,N,SUM y J son variables de tipo entero}

1. Leer N
2. Hacer I  $\leftarrow$  1
3. Repetir con I desde 1 hasta N  
Hacer SUM  $\leftarrow$  0 y J  $\leftarrow$  1
  - 3.1 Repetir con J desde 1 hasta (I DIV 2)
    - 3.1.1 Si (I MOD J) = 0 entonces  
Hacer SUM  $\leftarrow$  SUM + J
    - 3.1.2 {Fin del condicional del paso 3.1.1}  
Hacer J  $\leftarrow$  J+1
  - 3.2 {Fin del ciclo del paso 3.1}
  - 3.3 Si SUM = I entonces  
Escribir I, " Es un número perfecto "
  - 3.4 {Fin del condicional del paso 3.3}  
Hacer I  $\leftarrow$  I + 1
4. {Fin del ciclo del paso 3}