



Tema 3: Estructuras de control

I

- Estructura secuencial
- Estructura alternativa

II

- Estructuras repetitivas
- Variables de trabajo

4. Estructuras Repetitivas

- Varían o alteran la secuencia normal de ejecución de un programa haciendo posible que un grupo de acciones se ejecute más de una vez de forma consecutiva. También reciben el nombre de bucles o lazos.
- Todo bucle tiene tres partes:
 - **Condición o Expresión Condicional**
 - **Cuerpo**, instrucción o instrucciones que se ejecutarán si es verdadera la expresión.
 - **Salida o Final del bucle**



Ejemplo 1

- Sumar los números comprendidos entre el 1 y el 5, mostrando el resultado de la suma

Análisis del problema

- **¿Datos?** Los números límite: el 1 y el 5
- **¿Entrada?** En este ejemplo son valores fijos. No vamos a pedir ningún dato de entrada
- **¿Salida?** La suma: 15

Algoritmo:

- Empezando por el 1, iremos acumulando la suma de cada número hasta haber sumado el último (5) y obtener el resultado final
- **¿Repeticiones?:** Para cada número del 1 al 5 habrá que hacer lo mismo: añadirlo a la suma
- **¿estructuras alternativas?** No
- **¿Condiciones?** Para finalizar el bucle (cuando el número a sumar sea > que el límite(5) acabará el bucle)
- **¿variables?**
 - Una, para guardar la suma. Acumulará las sumas parciales, hasta obtener el resultado final.
 - Otra para guardar el número a sumar
- **¿Inicialización de las variables?** El número lo inicializamos a 1 y la suma a 0

num

suma

1 + 0

2 + 1

3 + 3

4 + 6

5 + 10

~~6~~

15

Valor inicial
 $\text{num} \leftarrow 1$

$\text{suma} \leftarrow \text{suma} + \text{num}$

$\text{num} \leftarrow \text{num} + 1$

condición

$\text{num} \leq 5$

Valor final
 $\text{num} \leftarrow 5$

**La variable
de control
se modifica
dentro del
bucle**

Variable de
control del bucle.
La que determina
que sigamos
sumando



Ejemplo 2

- Pedir al usuario la introducción de una opción adecuada al siguiente menú. Una vez seleccionada la mostrará y acabará.
- Menú:
 1. Suma
 2. Resta
 3. Multiplica
 4. Salir

Valor inicial
 $\text{valida} \leftarrow \text{Falso}$

condición

no
valida

Salida

Valor final
 $\text{valida} \leftarrow \text{Verdadero}$

opcion valida

12 Falso

-1 Falso

6 Falso

1 Falso
~~Verdadero~~

Variable de
control del bucle.
La que determina
que sigamos
pidiendo opciones

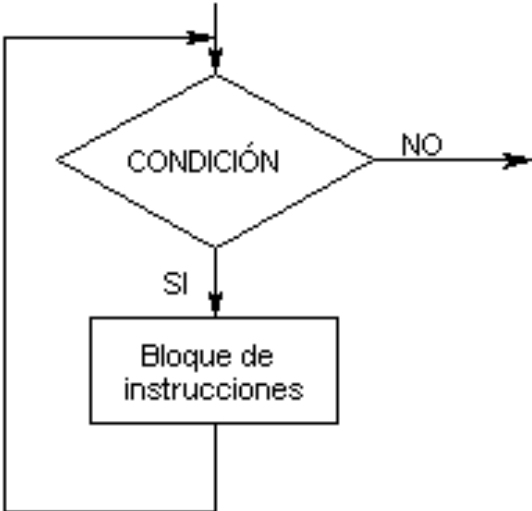
Leer opcion
 $\text{valida} \leftarrow \text{opcion} \geq 1 \text{ y } \text{opcion} \leq 4$
Si (no valida) entonces
 Escribir 'Opción no válida'
FinSi

La variable
de control
se modifica
dentro del
bucle

Estructura repetitiva Mientras

- La condición se evalúa antes de entrar al bucle
- Permite repetir un bloque de instrucciones de 0 a n veces dependiendo de la condición inicial, es decir:
 - Si la **condición** es **verdadera**, el n° de veces que se ejecutará dicho bloque de instrucciones será de **una** vez como mínimo y **n** como máximo.
 - Si la **condición** es **falsa** el bloque de instrucciones no se ejecutará **ninguna** vez.

Representación Mientras

FLUJOGRAMA	PSEUDOCÓDIGO
	<pre>MIENTRAS CONDICIÓN HACER INSTRUCCIÓN 1 INSTRUCCIÓN 2 ... INSTRUCCIÓN N FINMIENTRAS</pre>



En JAVA

```
while (condicion){
```

```
//Instrucciones
```

```
}
```

Ejemplo I con mientras

num		suma
1	+	0
2	+	1
3	+	3
4	+	6
5	+	10
6		15

Variable de control del bucle.
La que determina que sigamos sumando

Valor inicial
 $\text{num} \leftarrow 1$


$\text{num} \leftarrow 1$
Mientras $\text{num} \leq 5$ **hacer**
 $\text{suma} \leftarrow \text{suma} + \text{num}$
 $\text{num} \leftarrow \text{num} + 1$
FinMientras

condición

$\text{num} \leq 5$

Valor final
 $\text{num} \leftarrow 5$

La variable de control se modifica dentro del bucle



```
suma=0;
num=1;
while (num<=5){
    System.out.print(suma + "+" + num + "=");
    suma += num;
    System.out.println(suma);
    num++;
}
```

num	suma
	0
1	
	1
2	
	3
3	
	6
4	
	10
5	
	15
6	

Ejemplo 2 con mientras

Opcion valida

12 Falso

-1 Falso

6 Falso

1 Falso
Verdadero

Valor inicial
valida ← Falso

Salida

Valor final
valida ← Verdadero

condición

no
valida

La variable
de control
se modifica
dentro del
bucle

Variable de
control del bucle.
La que determina
que sigamos
pidiendo opciones

valida ← Falso
Mientras no valida hacer
Leer opcion
valida ← opcion ≥ 1 y opcion ≤ 4
Si (no valida) entonces
Escribir 'Opción no válida'
FinSi
FinMientras

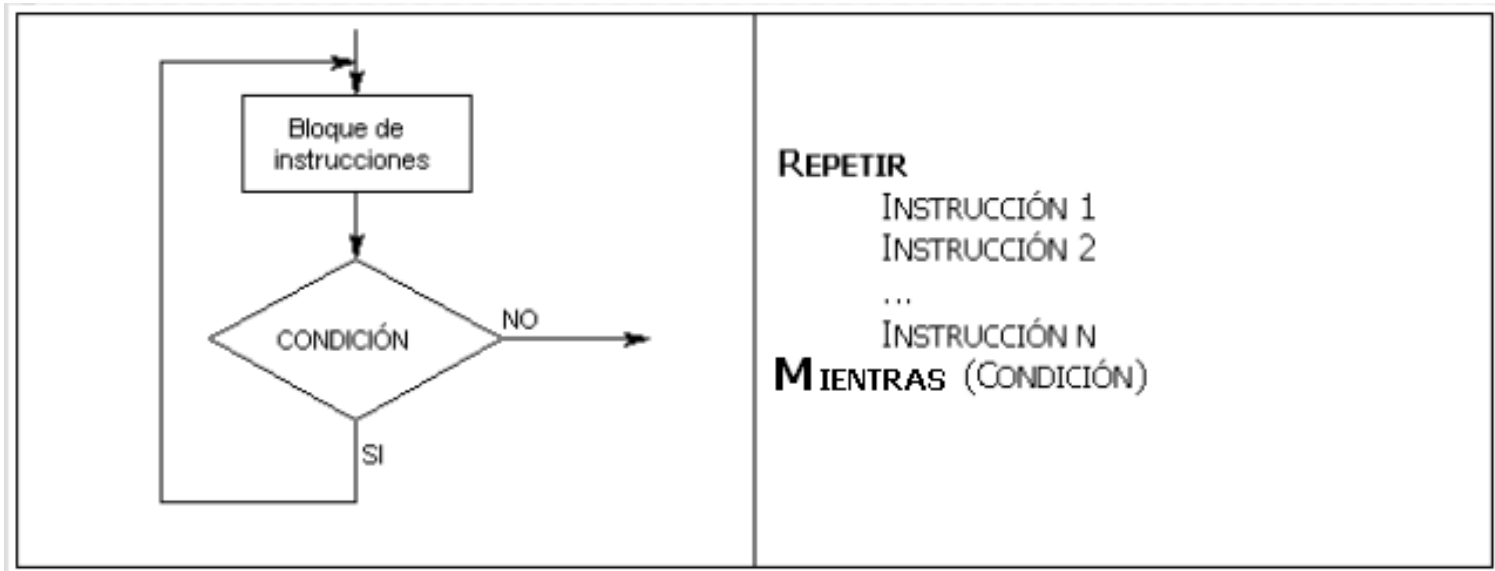


Estructura repetitiva

Repetir..Mientras

- Se caracteriza porque su diseño permite repetir un bloque de instrucciones de 1-n veces, es decir, si inicialmente la condición es:
 - Verdadera \rightarrow n° de veces 1 como mínimo.
 - Falsa \rightarrow n° de veces 1 vez como máximo.

Representación





En JAVA

do{

//Instrucciones

} while (condicion);

Ejemplo I con repetir..mientras

num		suma
1	+	0
2	+	1
3	+	3
4	+	6
5	+	10
6		15

Variable de control del bucle.
La que determina que sigamos sumando

Valor inicial
 $\text{num} \leftarrow 1$

$\text{suma} \leftarrow \text{suma} + \text{num}$
 $\text{num} \leftarrow \text{num} + 1$

condición

$\text{num} \leq 5$

Valor final
 $\text{num} \leftarrow 5$

La variable de control se modifica dentro del bucle

Ejemplo 1 en JAVA

```
void sumar5_r() {  
    int n = 1, suma = 0;  
    do {  
        suma = suma + n;  
        n++;  
    } while (n <= 5);  
    System.out.println(suma);  
}
```

run:

+1

+2

+3

+4

+5

15

BUILD SUCCESSFUL (total time: 1 second)

Ejemplo 2 con repetir..mientras

Valor inicial
valida ← Falso

condición

no
valida

opcion

valida

12

Falso

-1

Falso

6

Falso

1

Falso

Verdadero

Salida

Valor final
valida ← Verdadero

La variable
de control
se modifica
dentro del
bucle

Variable de
control del bucle.
La que determina
que sigamos
pidiendo opciones

Leer opcion

valida ← opcion ≥ 1 y opcion ≤ 4

Si (no valida) entonces

Escribir 'Opción no válida'

FinSi

Ejemplo 2 en JAVA

```
void leer_opcion_r() {  
    int opcion;  
    Scanner teclado = new Scanner(System.in);  
    do {  
        System.out.println("Nº Opción:");  
        opcion = teclado.nextInt();  
        if (opcion < 1 || opcion > 4) {  
            System.out.println("Opción incorrecta");  
        }  
    } while (opcion < 1 || opcion > 4);  
}
```

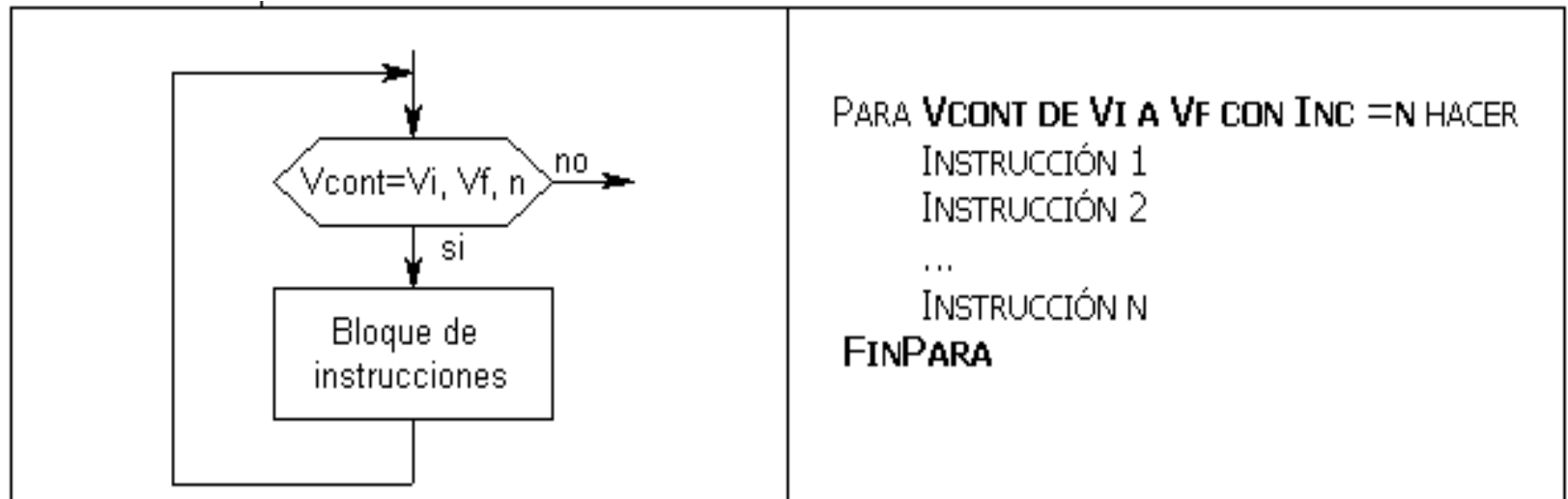
```
run:  
Nº Opción:  
0  
Opción incorrecta  
Nº Opción:  
7  
Opción incorrecta  
Nº Opción:  
2
```



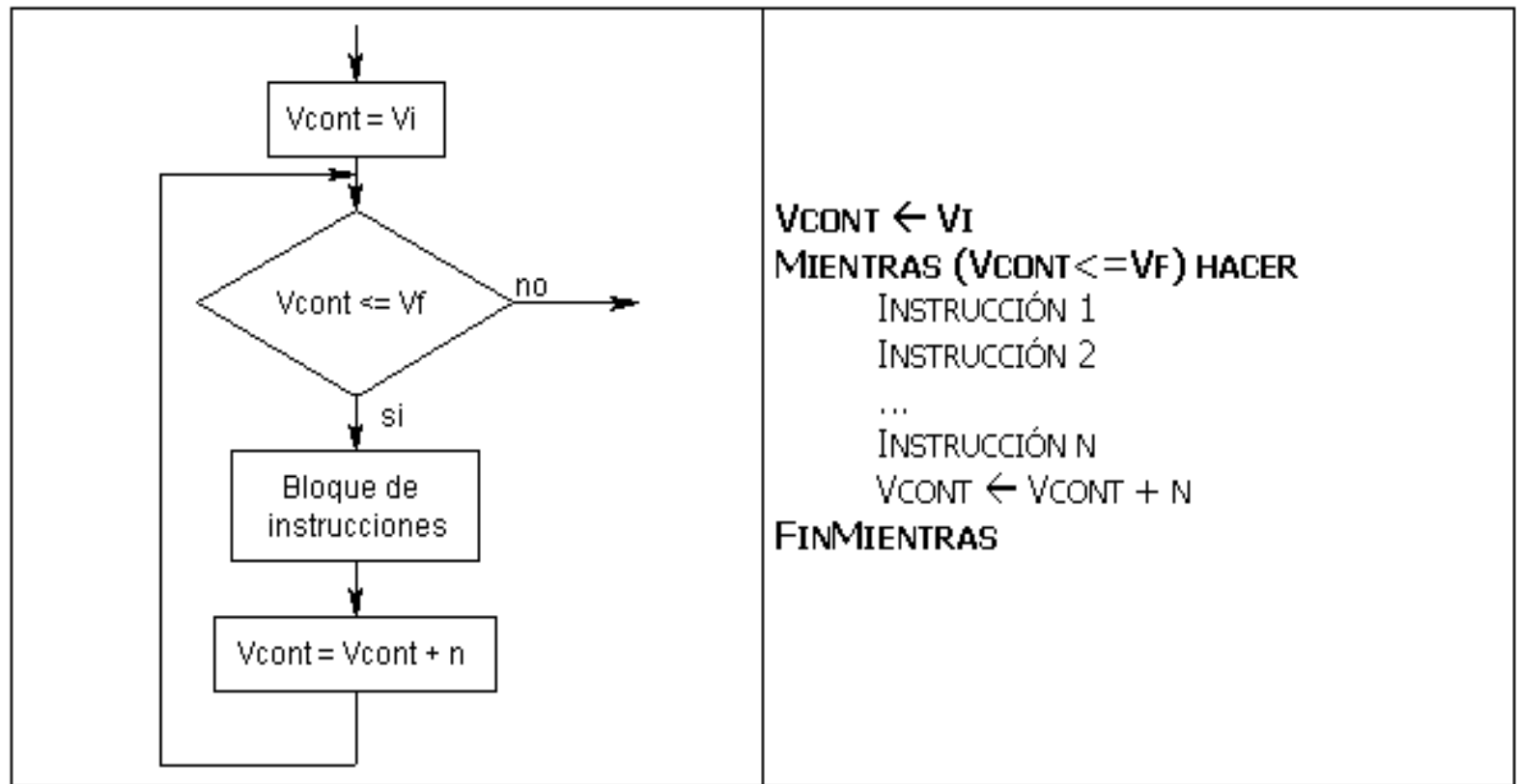
Estructura repetitiva Para

- Se caracteriza porque el n° de veces que se repetirá el bloque de instrucciones generalmente está fijado de antemano. Necesitaremos:
 - **Vcont** la variable contador del bucle
 - **Vi**, el valor inicial que toma Vcont
 - **Vf**, Valor final para Vcont
 - **N**: cantidad que incrementa o decrementa la variable Vcont al final de cada vuelta del bucle. Por defecto el valor es siempre positivo.

Representación



Equivalencia con mientras



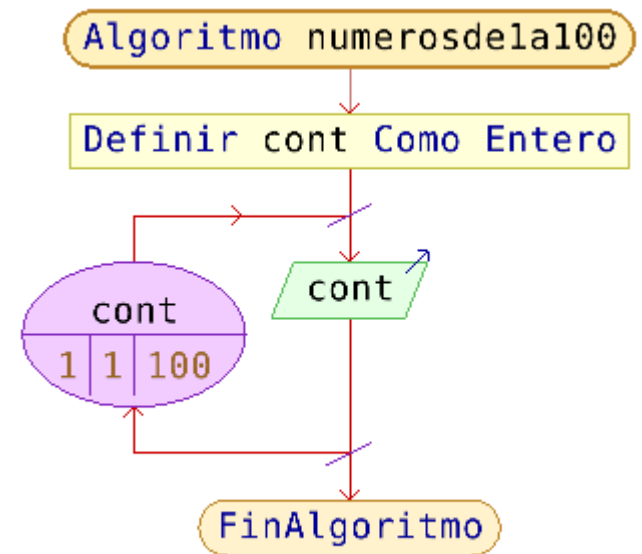


Ejemplo:

- Mostrar los números comprendidos entre el 1 y el 100

Solución

```
Algoritmo numerosde1a100
  definir cont como entero
  Para cont<-1 Hasta 100 Con Paso 1 Hacer
    ...   Escribir cont
  Fin Para
FinAlgoritmo
```



Hacerlo con un mientras

```
Algoritmo numerosde1a100
    definir cont como entero
    cont=1
    mientras con<=100
        Escribir cont
        cont=cont+1
    Fin mientras
FinAlgoritmo
```

Ejemplo 1 con "Para"

num suma

1 + 0

2 + 1

3 + 3

4 + 6

5 + 10

~~6~~

Variable de
control del bucle.
La que determina
que sigamos
sumando

Valor inicial
 $\text{num} \leftarrow 1$

$\text{suma} \leftarrow \text{suma} + \text{num}$

Valor final
 $\text{num} < -6$

condición

$\text{num} \leq 5$

La variable
de control
se modifica
dentro del
bucle

```

Proceso sumar_5
  suma<-0
  Para num<-1 Hasta 5 Con Paso 1 Hacer
    .....
    suma<-suma+num;
  FinPara
  Escribir suma;
FinProceso

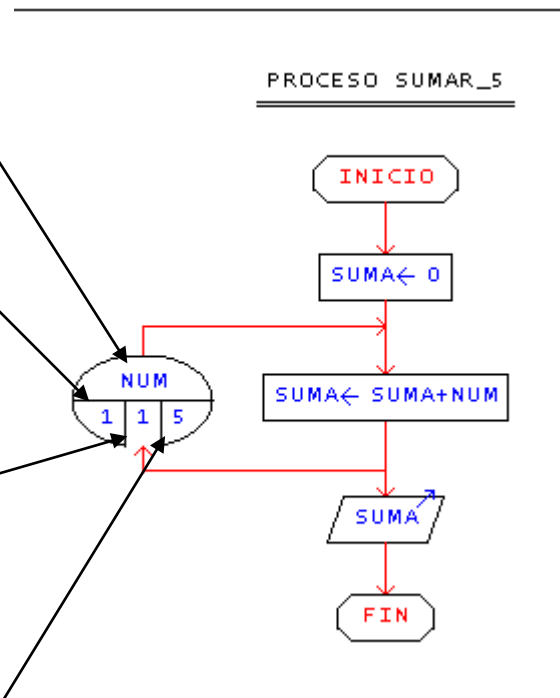
```

Variable de control

Valor inicial

Variación

Valor final



La instrucción Para en JAVA

```
for (inicialización; condición; variación) {  
    //instrucciones  
  
}
```

Ejemplo suma 5 en JAVA

```
void sumar5() {  
    int n, suma = 0;  
    for (n = 1; n <= 5; n++) {  
        System.out.println("+"+n);  
        suma = suma + n;  
    }  
    System.out.println(suma);  
}
```

```
run:  
+1  
+2  
+3  
+4  
+5  
15
```

Elegir el bucle más adecuado

■ Sabemos el n° de veces (Ejemplo 1)

■ Ninguna o más veces

- Para
- Mientras

■ Una o más veces

- Para
- Mientras
- Repetir

No sabemos el n° de veces (Ejemplo 2)

Ninguna o más veces

Mientras

Una o más veces

Repetir

Mientras



Las instrucciones break; y continue;

Estas instrucciones se utilizan dentro de los bucles:

- para abandonar el bucle (**break**)
- para saltar una iteración (**continue**)



Contadores

- Un contador variable cuyo valor se incrementa en una cantidad fija, positiva o negativa. Se utiliza en los siguientes casos:
 - Como variable de control de un bucle.
 - Para contar un suceso particular solicitado por el enunciado del problema (asociado a un bucle o independientemente)
- Un contador toma un valor inicial (0 en la mayoría de los casos) antes de comenzar su función, posteriormente, y cada vez que se realiza el suceso a contar, incrementa su valor (1 en la mayoría de los casos).
- En el ejemplo 1 (suma), num actuaba como contador



Ejemplo de contador

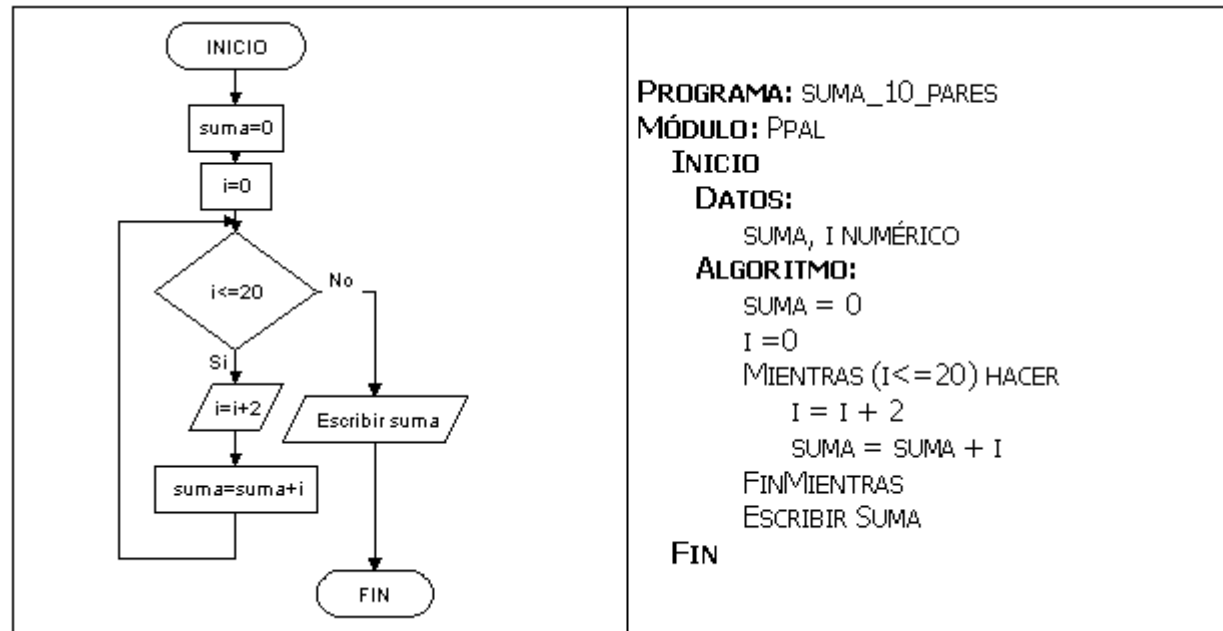
- Programa que lea 100 números y cuente cuantos de ellos son positivos.

Acumulador

- Es una variable de memoria cuyo valor se incrementa sucesivas veces en cantidades variables.
- Se utiliza en aquellos casos en que se desea obtener el **total acumulado** de un conjunto de cantidades
- Puede ser inicializado con el valor 0 o a 1 en función de si se acumulan sumas o productos.
- En el ejemplo 1, la variable suma actuaría como acumulador

EJEMPLO:

- Algoritmo que suma los 10 primeros números pares





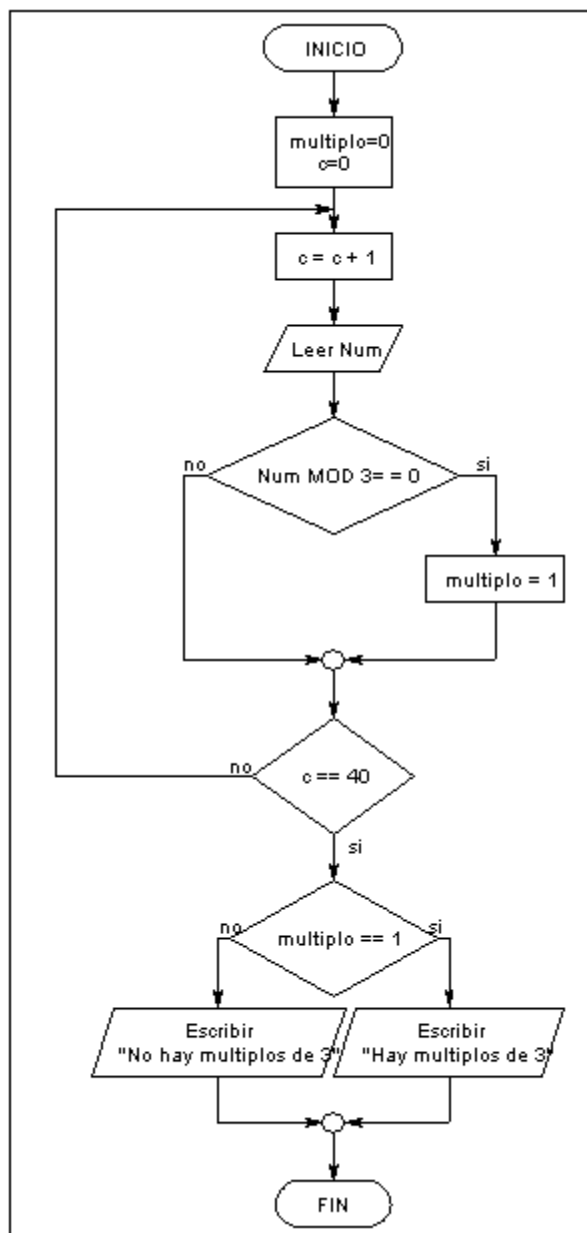
Interruptores (Switches)

- Es una variable que puede tomar dos valores exclusivos (0 y 1, V o F). Se utiliza para:
 - Recordar en un determinado punto de un programa la ocurrencia o no de un suceso anterior, para salir de un bucle o para decidir en una instrucción alternativa qué acción realizar.
 - Para hacer que dos acciones distintas se ejecuten alternativamente dentro de un bucle.
- Deben de ser inicializados con cualquiera de los dos únicos valores que podrán tomar.



EJEMPLO I

- Diseñar un algoritmo que lee 40 números, los procesa, y después de leer el último determina si alguno era múltiplo de 3



PROGRAMA: MÚLTIPLOS_3

MÓDULO: PPAL

INICIO

DATOS:

MÚLTIPLO BOOLEANO

NUM, C NUMÉRICO

ALGORITMO:

C = 0

MÚLTIPLO = -0

REPETIR

C = C + 1

LEER NUM

SI (NUM MOD 3=0)

MÚLTIPLO = 1

FINSI

HASTA (C = 100)

SI (MÚLTIPLO = 1)

ESCRIBIR "HAY MÚLTIPLOS DE 3"

SINO

ESCRIBIR "NO HAY MÚLTIPLOS DE 3"

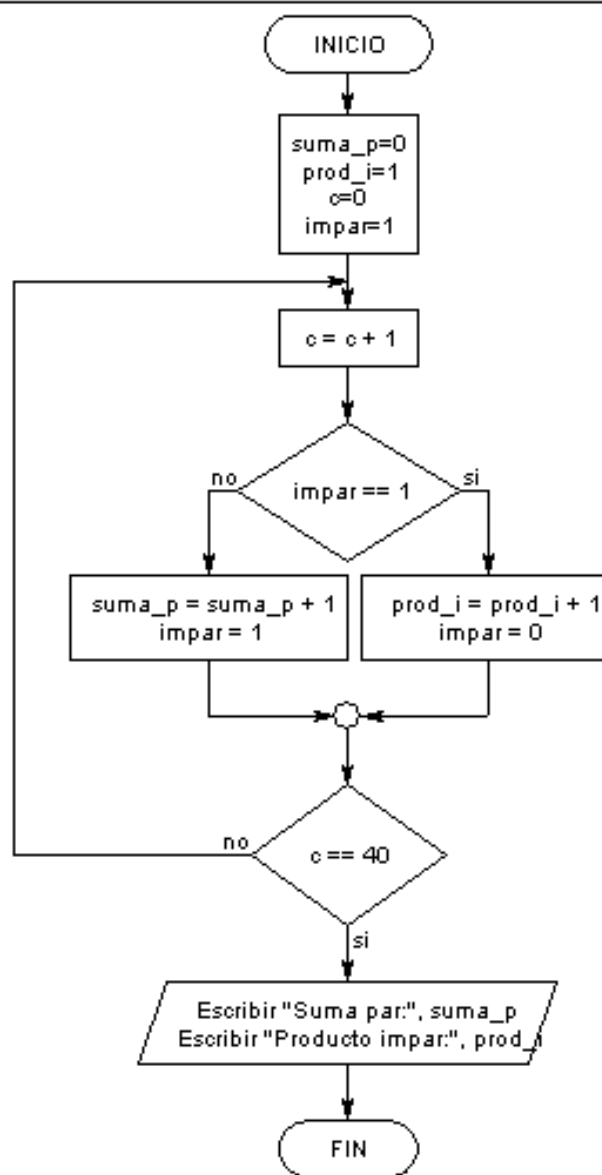
FINSI

FIN



Ejemplo 2

- Diseñar un algoritmo que calcule la suma de los 20 primeros números pares, y el producto de los 20 primeros números impares simultáneamente:



PROGRAMA: SUMA_PROD_PAR_IMP

MÓDULO: PPAL

INICIO

DATOS:

SUMA_P, PROD_I, CONT NUMÉRICOS

IMPAR BOOLEANA

ALGORITMO:

PROD_I = 1

SUMA_P = 0

C = -0

IMPAR=1

REPETIR

C = C + 1

SI (IMPAR=1)

PROD_I = PROD_I * C

IMPAR=0

SINO

SUMA_P=SUMA_P + C

IMPAR=1

FINSI

HASTA (C=40)

ESCRIBIR "SUMA PAR:", SUMA_P

ESCRIBIR "PRODUCTO IMPAR:", PROD_I

FIN



Ejercicio

- Diseña un algoritmo que introduzca por teclado un número y permita la introducción de otros números hasta adivinar el número tecleado en primer lugar. Cada vez que se introduzca un número se visualizará si es mayor o menor que el número a adivinar. Utiliza un interruptor para determinar cuando se adivina el número.