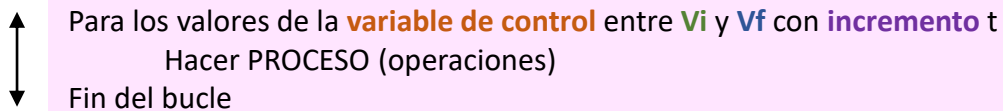


# BUCLES SECUENCIALES (*for*)

Los bucles secuenciales o tipo *for* son estructuras que **permiten realizar de forma repetitiva una serie de sentencias** que se encuentran en su interior.

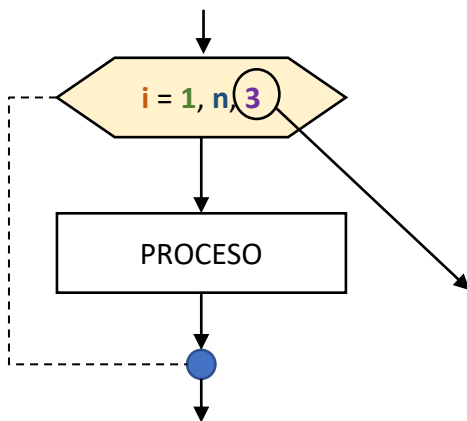
El bucle *for* funciona mientras cierta **variable de control**, normalmente utilizamos la letra *i* (iterador), no exceda un determinado valor. De esta manera, la variable de control irá variando desde un **valor inicial (Vi)** hasta un **valor final (Vf)** siempre con un **incremento** determinado.

- Pseudocódigo general:



## ESTRUCTURA

En organigramas, el **símbolo** que utilizaremos será EL **HEXÁGONO**



ELEMENTOS BUCLE:

**Variable de control**

**Valor inicial**

**Valor final**

**Incremento**

**APUNTE:** en este caso, hemos marcado el **incremento** del bucle porque toma valores de 3 en 3 (1,4,7,10,...).

- Si el incremento fuera **DE UNO EN UNO** (1,2,3,4,...), no sería necesario.

```
En R:  
for (i in 1: n) {  
    Proceso de cálculo  
}
```

## USOS FRECUENTES

- Los bucles secuenciales o tipo *for* se emplean sobre todo en **operaciones en las que aparecen VECTORES o MATRICES**: suma y resta de sus componentes, producto de vectores/matrices, etc...
- En ejercicios que pidan hallar **EL ELEMENTO DE MÁXIMO/MENOR VALOR**.

**Ejemplo 1:** realice el organigrama y pseudocódigo para obtener el valor del sumatorio:

$$P = \sum_{i=1}^n i^2(1+i)$$

El primer paso es comprender el enunciado que nos proponen:

**P** es el valor que queremos hallar con el algoritmo, el cual se corresponde con el **sumatorio de un proceso**.

En el sumatorio aparece una **variable de control, i**, que va a tomar **valores desde 1 hasta n**, por lo que será **nuestro bucle**.

Una vez establecido el **bucle i = 1, n**, tenemos que identificar en qué proceso participa la toma de valores. En este caso, se corresponde con el interior del sumatorio, es decir,  **$i^2(1+i)$** . Así, por cada vuelta que dé el bucle (cada valor atribuido a i), el proceso será el siguiente:

**P=0**, ya que aquí vamos a almacenar el valor anterior del sumatorio más el correspondiente al de cada vuelta del bucle.

$P = 0$

$i = 1 \quad P = 0 + 2 = 2$

$i = 2 \quad P = 2 + 12 = 14$

$i = 3 \quad P = 14 + 36 = 50$

(i = ...)

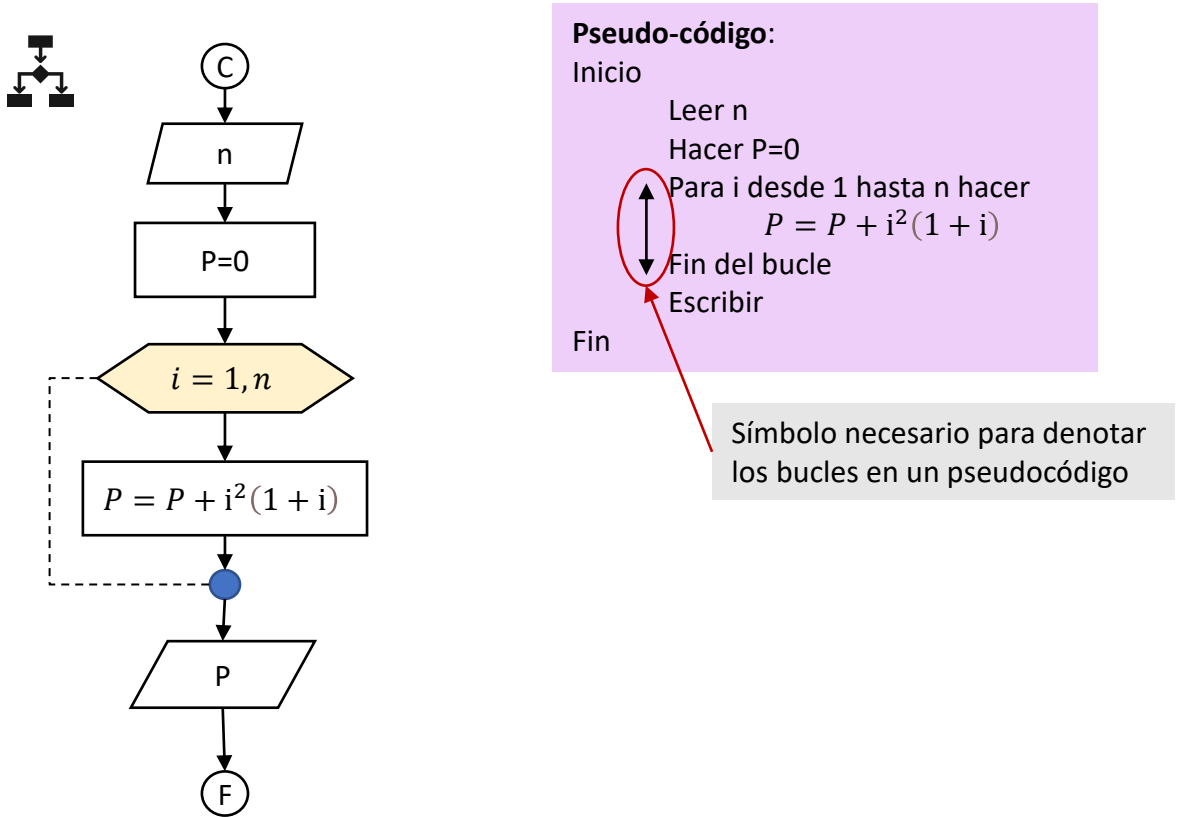
$i = n \quad P = P + n^2(1+n)$

}

**Fórmula general:**

$P = P + i^2(1+i)$

Una vez tenemos la fórmula general que va dentro del bucle, podemos **realizar el organigrama y el pseudocódigo**:



# BUCLES ANIDADOS

Un bucle anidado es **un bucle dentro de otro**, es decir, un bucle de  $i$  ( $i = 1, n$ ) tomando valores en función de los valores de  $j$  ( $j = 1, m$ ), siendo  $m$  menor o igual a  $n$ .

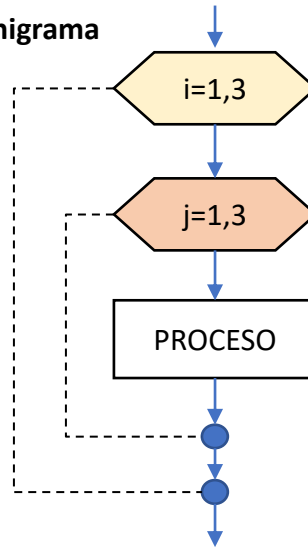
**Ejemplo de asimilación:**

Desarrollo de valores por los bucles:

```

i = 1, j=1
      j=2
      j=3
i = 2, j=1
      j=2
      j=3
i = 3, j=1
      j=2
      j=3
    
```

Organigrama



Estos bucles se emplean sobre todo en operaciones con MATRICES

**Ejercicio:** realice el pseudocódigo para restar dos matrices,  $A[m,n]$  y  $B[m,n]$ , y obtenga  $C[m,n]$ .

**Pseudocódigo:**

```

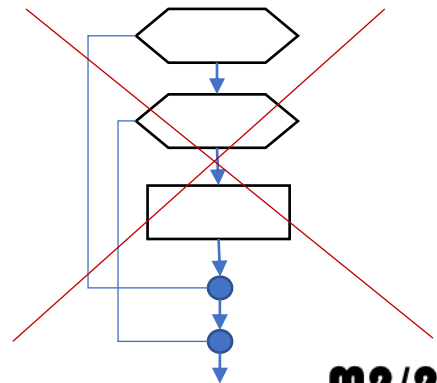
Inicio pseudocódigo
Leer A,B,m,n
  Para i desde 1 a m
    Para j desde 1 a n
      Hacer C[i,j] = A[i,j]-B[i,j]
    Fin del bucle en j
  Fin del bucle en i
Leer C
Fin pseudocódigo
    
```

Atribuimos la **variable de control  $i$**  al número de **fila** y la **variable de control  $j$**  al de **columna**.  
De esta manera, con sus respectivos bucles, **podemos ir moviéndonos a lo largo de las componentes de cada matriz** y así poder sumarlas para crear la matriz C.



Un **error frecuente** en los bucles anidados es **NO RESPETAR EL ORDEN LÓGICO DE CIERRE**: el primer bucle abierto es el último que se cierra; el segundo, el penúltimo;..., etc.  
En un organigrama, las **líneas de los bucles NO DEBEN CRUZARSE**.

**Contraejemplo:**



# BUCLES CONDICIONALES (*while*)

Los bucles condicionales o tipo *while* son estructuras que permiten repetir una serie de procesos mientras una condición se cumpla. La verificación o no de esta condición determina que el bucle continúe (si se cumple) o finalice y el flujo salga de él (si no se cumple).

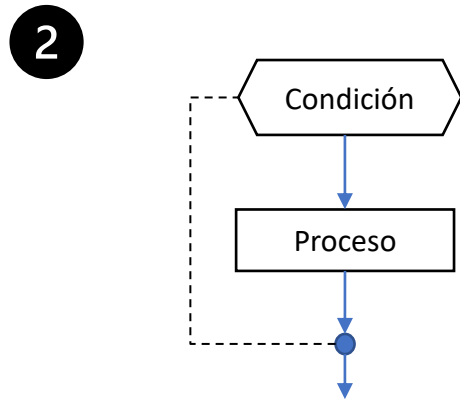
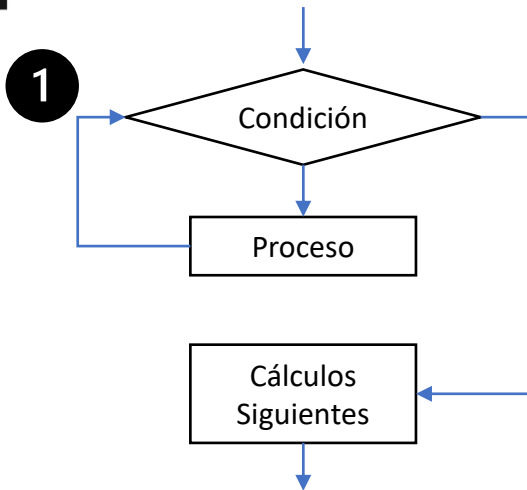
## Pseudocódigo general:

```
Inicio pseudocódigo
  Datos
  Instrucciones
  ↑ 'Mientras se cumpla cierta condición'
  ↓   Hacer PROCESO DE CÁLCULO
  ↓ Fin del bucle
  Instrucciones
  Escribir resultado
Fin del bucle
```

```
En R:
while (condición) {
  Proceso de cálculo
}
```



En cuanto al **organigrama**, se proponen dos tipos:



Un **error frecuente** en los bucles condicionales es **GENERAR UN BUCLE INFINITO**: hemos de asegurarnos de que en algún momento la condición deje de cumplirse para que finalice el bucle y siga el programa.

Asimismo, **si la condición no se cumple al inicio**, ni siquiera entraríamos en el bucle, por lo que procuraremos que esta sí se verifique.

## NOTA: diferencia entre bucle *for* y *while*

Los bucles *for* realizan la repetición en función de los valores que toma su variable de control (conocemos el número de vueltas), mientras que los bucles *while* precisan de la verificación de una condición (no conocemos el número de vueltas).

**Ejemplo de asimilación:** realice un algoritmo que sume los números naturales mientras que la suma sea menor o igual a P y obtenga como resultado:

- Suma obtenida ( $\leq P$ )
- Último número natural sumado

*Desarrollo hacia el algoritmo*

1. Llamaremos al último número natural sumado 'K' y lo inicializamos a 1,  $K=1$ , ya que será el primer número natural sumado y en él se almacenará la suma del resto.
2. Como se realiza la suma de los números naturales, esto **implica un sumatorio**, por lo que **llamamos a una variable 'suma'** y la **inicializamos a cero,  $\text{suma}=0$** , ya que aquí almacenaremos la suma de valores en cada vuelta del bucle.
3. Observamos el **desarrollo de la suma propuesta en función de**, por ejemplo,  $P=9$ :

Si  $P=9$

$K=1, \text{suma}=0$  (se cumple  $\text{suma} \leq 9$ )

Entramos en:  $\text{suma}=0+1=1$

$K=1+1=2$

$K=2, \text{suma}=1$  (se cumple  $\text{suma} \leq 9$ )

Entramos en:  $\text{suma}=1+2=3$

$K=2+1=3$

$K=3, \text{suma}=3$  (se cumple  $\text{suma} \leq 9$ )

Entramos en:  $\text{suma}=3+3=6$

$K=3+1=4$

$K=4, \text{suma}=6$  (se cumple  $\text{suma} \leq 9$ )

Entramos en:  $\text{suma}=6+4=10$

$K=4+1=5$

$K=5, \text{suma}=10$  (NO SE CUMPLE  $\text{suma} \leq 9$ )

El bucle finaliza con  $K=5$  y  $\text{suma}=10$ . Sin embargo, nosotros queremos  $\text{suma} \leq 9$  y el último número natural sumado (K).

La última vuelta que cumple la condición  $\text{suma} \leq 9$  es en la que  $K=4$  y  $\text{suma}=6$ .

- Podemos conseguir  $\text{suma}=6$  restándole 4 a  $\text{suma}=10$ . Si a la K obtenida,  $K=5$ , le restamos la unidad ( $K=5-1$ ), nos da  $K=4$ . Así, al restar  $K=4$  a  $\text{suma}=10$ , su resultado,  $\text{suma}=6$ , sí respeta la condición  $\text{suma} \leq 9$ . Haremos  $\text{suma}-(K-1)$ .
- En cuanto al **valor de K final**, la última K sumada es  $K=3$  ( $K=4$  no llega a sumarse). Para obtenerla, tendremos que hacer  $K-2$  ( $K=5-2=3$ ).



El organigrama quedaría así:

