

BUCLES ANIDADOS

RECOMENDACIÓN DE USO

¿Sigues con dudas tras la clase de bucles? ¿Crees que no has entendido todos los conceptos? Te recomendamos leer este resumen con el fin de reforzar los conceptos dados en clase y tenerlo todo claro para el examen.

JUSTIFICACIÓN

Los bucles anidados son una herramienta básica en la realización de algoritmos. Como es muy importante tener claro cómo usarlos, cuándo y en qué orden; desde nuestro grupo hemos decidido crear unos apuntes claros y visualmente llamativos que permitan la comprensión de este tema, con el objetivo de poder avanzar en la asignatura de Fundamentos de Programación.

[Foro discusión tema del resumen.](#)

INTRODUCCIÓN

A menudo, al hacer un programa necesitamos una estructura que nos permita **repetir un proceso un número definido de veces** y esta es el bucle. Se utilizan en la mayoría de programas pues son casi imprescindibles.

En este caso nos vamos a centrar en los **bucles anidados** o también llamados secuenciales, usados comúnmente en algoritmos donde intervienen **vectores y/o matrices**, además de sumatorios y productorios. En esta explicación pondremos el foco en las matrices.

A continuación se encuentra la explicación y estructura, además de algunos trucos para evitar los errores más frecuentes y un ejemplo para que quede todo más claro.

¡Esperemos que te guste, y que quede todo más claro ;) !

EXPLICACIÓN

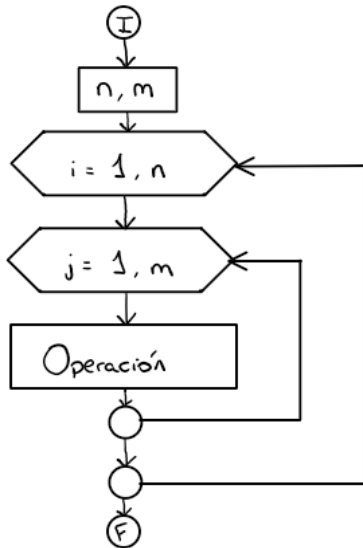
Los bucles anidados (*nested loops*) son una **sucesión de bucles**, hay varias variables para la misma operación y cada variable toma valores distintos. Puede haber tantos bucles como variables haya.

Se usan principalmente en matrices ya que tienen n filas y m columnas, por lo que hay dos variables, llamemoslas i y j que varían de forma distinta. Es por esto que se necesitan escribir dos bucles, uno para cada variable. Vamos a ver la estructura que debe seguir un organigrama y un pseudo-código para un algoritmo que tenga dos variables.

En general, se sigue este proceso:

$i=1, j=1$ operación
 $j=2$ operación
 $i=2, j=1$ operación
 $j=2$ operación
... etc

Organigrama:

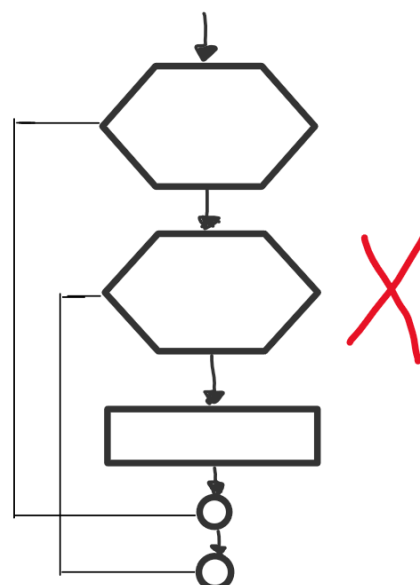
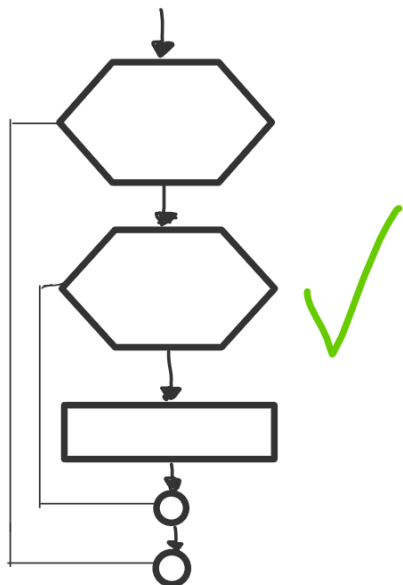


Pseudo-código:

```
Inicio
Leer n y m.
  Para i desde 1 hasta n.
    Para j desde 1 hasta m.
      Operación.
    Fin bucle en j.
  Fin bucle en i.
Fin
```

A TENER EN CUENTA

- EL BUCLE QUE SE ABRE PRIMERO SE CIERRA EL ÚLTIMO**
Nunca se pueden cruzar las líneas.



RAZONAR SI LA OPERACIÓN VA DENTRO O FUERA DEL BUCLE

Para saber si un proceso va fuera o dentro de un bucle, hace falta pensar y razonar sobre todo.

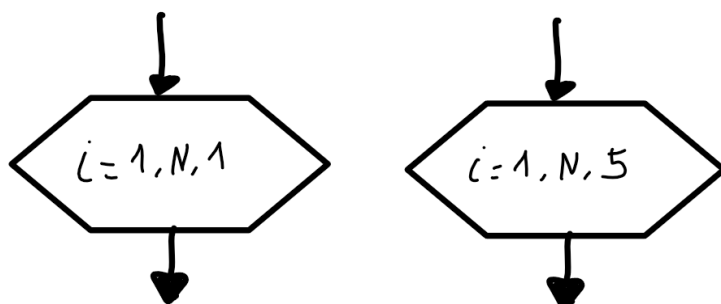
Lo que se realiza de forma repetida va en el interior del bucle y dependerá de una variable y lo que se quiera hacer una única vez va fuera.

TENER EN CUENTA LA SECUENCIA DEL BUCLE

Es decir, se puede indicar que el bucle se repite a intervalos de 1 en uno, de dos en dos, de cinco en cinco...

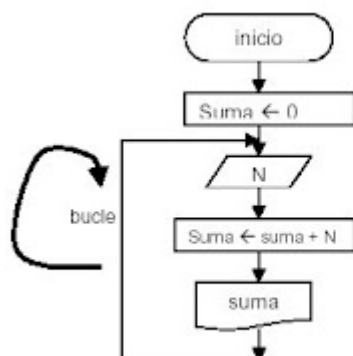
Esto se indica mediante un tercer número en la estructura.

Si no se indica, suponemos que va de uno en uno.



ES NECESARIO CERRAR EL BUCLE

Como observamos en el ejemplo, N no tiene fin, luego se repetirá indefinidamente. Para que no ocurra esto, hace falta cerrarlo.



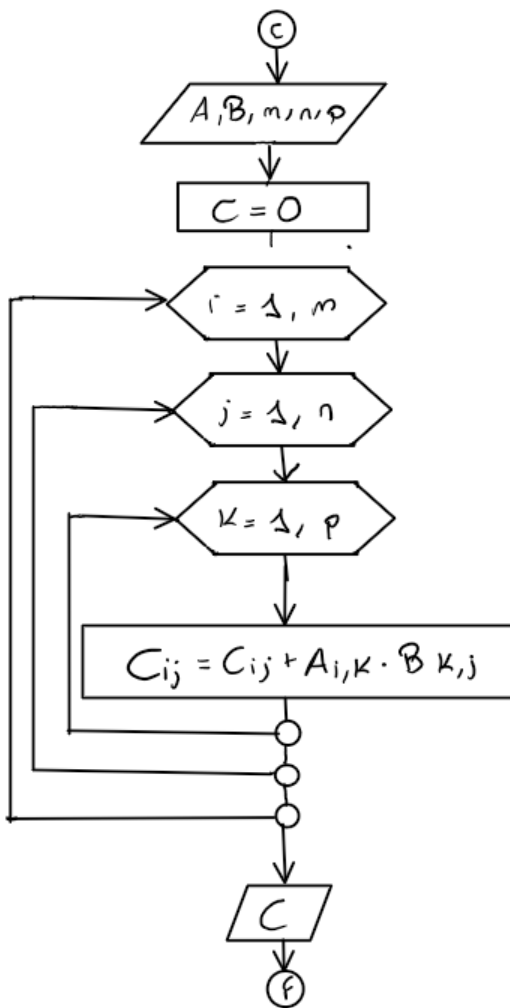
En este flujograma, el bucle se estará repitiendo indefinidamente ya que no existe ninguna condición que nos permita finalizar en algún momento.

EJEMPLO

Realizar un algoritmo para multiplicar matrices. $A(m,p) \times B(p,n) = C(m,n)$.

La multiplicación sigue esta fórmula: $C_{i,j} = \sum A_{i,k} \times B_{k,j}$. En un algoritmo no se puede poner un sumatorio por lo que se pone $C_{i,j} = C_{i,j} + A_{i,k} \times B_{k,j}$.

Nota: hay que poner $C=0$ porque es necesario que tenga un valor para cuando la introduzcamos en la operación.



```

  Inicio
  Leer A, B, m, n, p
  C=0
  Para i desde 1 hasta m
    Para j desde 1 hasta n
      Para k desde 1 hasta p
        Cij = Cij + Ai,k x Bk,j
      Fin bucle en k
    Fin bucle en j
  Fin bucle en i
  Escribir C
  Fin
  
```

Operaciones que realiza el algoritmo paso a paso:

Inicio

Lectura de datos: $A = \begin{pmatrix} 1 & 2 \\ -3 & 0 \end{pmatrix}$ $B = \begin{pmatrix} 3 & 5 \\ 4 & 1 \end{pmatrix}$ $m=2$ $n=2$ $p=2$

$i=1$

$j=1$

$k=1$

$C_{1,1} = 0 + 1 \times 3 = 3$

$k=2$

$$C_{1,1} = 3 + 2 \times 4 = 11$$

$$j=2$$

$$k=1$$

$$C_{1,2} = 0 + 1 \times 5 = 5$$

$$k=2$$

$$C_{1,2} = 5 + 2 \times 1 = 7$$

$$i=2$$

$$j=1$$

$$k=1$$

$$C_{2,1} = 0 + (-3) \times 3 = -9$$

$$k=2$$

$$C_{2,1} = -9 + 0 \times 4 = -9$$

$$j=2$$

$$k=1$$

$$C_{2,2} = 0 + (-3) \times 5 = -15$$

$$k=2$$

$$C_{2,2} = -15 + 0 \times 1 = -15$$

Resultado: $C = \begin{pmatrix} 11 & 7 \\ -9 & -15 \end{pmatrix}$

Fin

METAINFORMACIÓN:

<u>Tema</u>	<u>Tiempo</u>	<u>Tipo</u>	<u>Destinatario</u>
Algoritmia: Bucles Anidados	Impartida por Arturo Hidalgo Lopez durante el primer semestre de la asignatura de fundamentos de programación del año 2021, dirigido al primer parcial .	Es un resumen que incluye un ejemplo y reglas básicas	Está dirigido a un alumno que quiera ampliar sus apuntes de la asignatura así como aquel que no se haya enterado muy bien en clase.

CONTROL DE CALIDAD

Este trabajo ha sido repasado por todos los alumnos del equipo.