

BUCLES

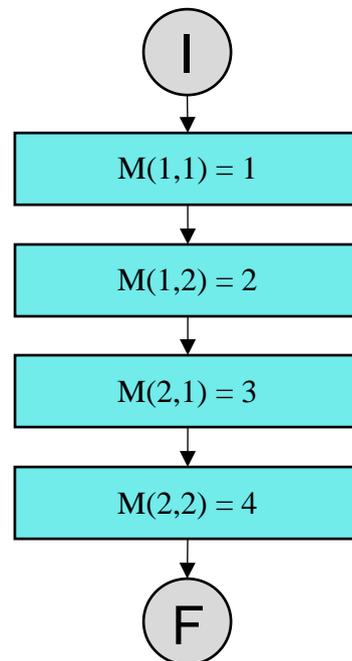
¿Cómo se puede construir una matriz de tantas filas y tantas columnas, tal que sus valores sean una secuencia de números naturales (1, 2, 3...)?

Si la matriz es de 2 filas y 2 columnas, sería:

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

Y el algoritmo para crearla solo constaría de cuatro comandos: la definición de cada uno de sus cuatro elementos.

Sin embargo, si la matriz es de medidas más grandes, como 10x10 o 15x30, o no se conocen sus dimensiones (son genéricas), no podemos construirla de esta manera. En este caso recurrimos a los bucles, porque los elementos siguen un patrón.

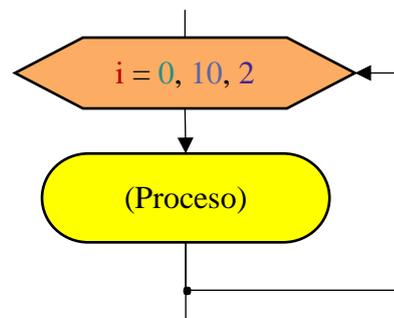


Un bucle es una secuencia o mecanismo de instrucciones que se repite tantas veces como se tenga indicado en sus condiciones, es decir, hasta que estas se dejen de cumplir. Funcionan en casos como el ejemplo anterior, en algoritmos que requieren efectuar operaciones similares una y otra vez, de modo que estos se simplifican. Conoceremos a continuación los bucles “for” y los bucles “while”.

BUCLES FOR

Actúan durante un número de veces definido anteriormente, en sus condiciones.

En un organigrama se representan con un hexágono aplanado. Su estructura es una **variable** igualada a una secuencia de tres valores, que son, respectivamente, su **valor inicial**, su **valor final** y el **incremento**. Es decir, en el ejemplo de la derecha, se define la variable *i*, que tomará valores del 0 al 10, de dos en dos (serán 0, 2, 4, 6, 8 y 10). El incremento puede omitirse y no escribirse (se interpretaría como 1).

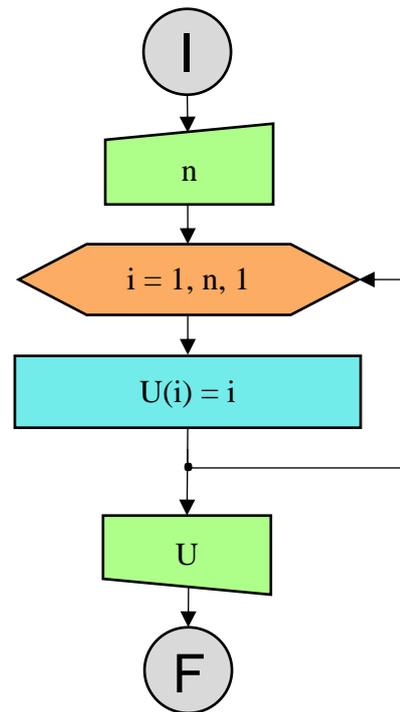


Las fórmulas a partir de las que se programan bucles “for” se pueden identificar porque tienen, al menos, una variable con subíndice a la izquierda del “=”.

Por ejemplo, para construir un vector con las condiciones planteadas anteriormente (que sus valores sean una secuencia de números naturales en orden), la fórmula sería:

$$U_i = i \quad i = (1, 2, 3 \dots, n)$$

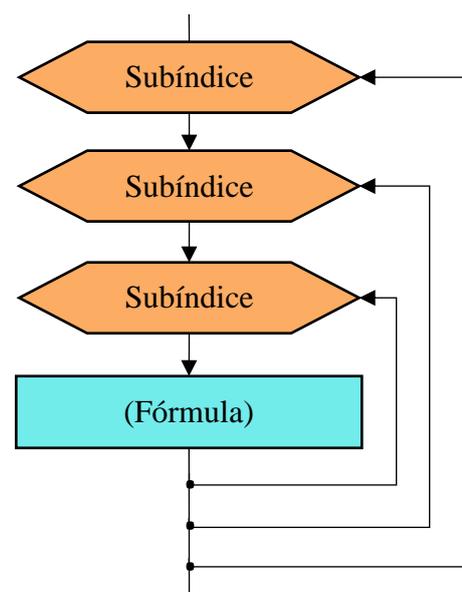
Siendo U el vector y n el número de componentes. El algoritmo para formar tal vector sería el de la derecha. Primero, se introduce el valor de la variable n, que será el número de componentes del vector (y el número de veces que se repetirá el bucle). De este modo podemos generalizar el algoritmo para cualquier valor de n. Luego introducimos el bucle utilizando la variable i, que irá de 1 a n, con la secuencia para definir cada componente del vector U (la fórmula del vector). Se cierra el bucle y se escribe el vector U obtenido.



Pasos para programar un algoritmo con bucles “for”:

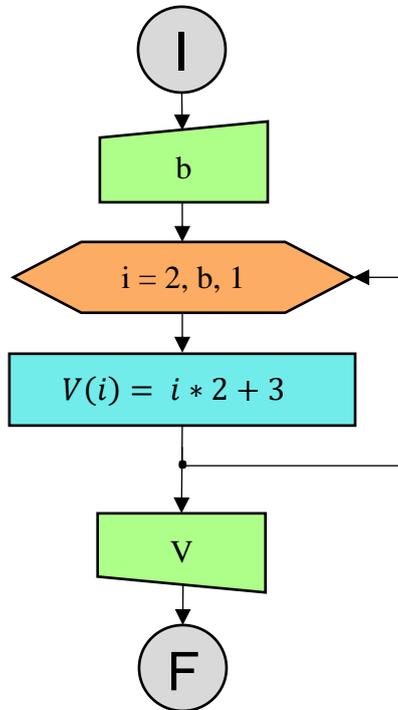
A partir de una fórmula es fácil construir un algoritmo con bucles, de modo que se pueden seguir las siguientes instrucciones:

1. Escribir la fórmula en un comando de proceso u operación (rectángulo).
2. Dibujar tantos bucles “for” como subíndices expresados en variables haya a la izquierda del “=” (uno dentro de otro, es decir, bucles anidados).
3. Escribir, en las instrucciones de cada bucle, un subíndice igualado a, en orden, su valor inicial, valor final e incremento (estos valores deben estar presentes junto a la fórmula, porque explican cómo varía cada subíndice).

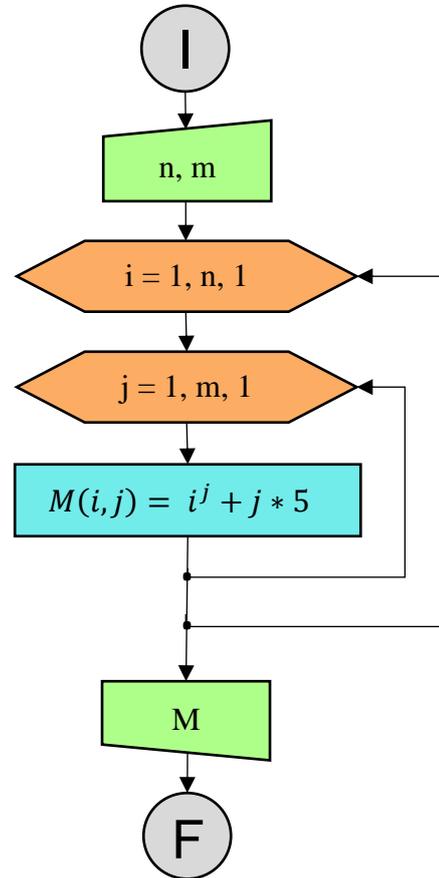


Ejemplos:

1) $V_i = i * 2 + 3$ $i = (2, 3, 4 \dots, b)$
(vector)



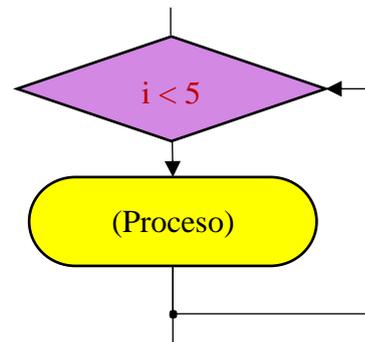
2) $M_{i,j} = i^j + j * 5$ $i = (1, 2, 3 \dots, n)$
(matriz) $j = (1, 2, 3 \dots, m)$



BUCLES WHILE

Actúan hasta que la condición (o condiciones) que los rige se deje de cumplir.

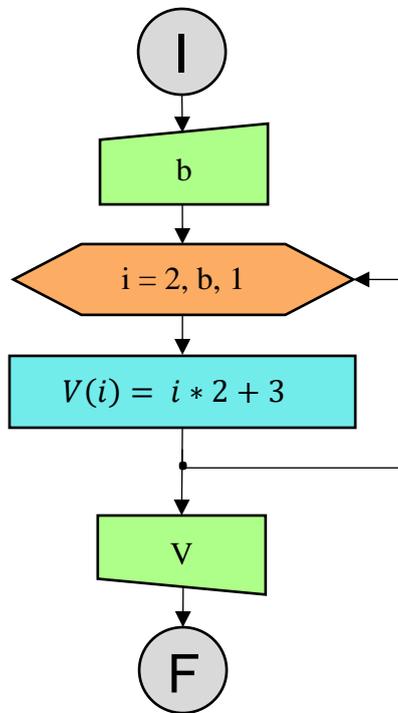
En un organigrama se representan con un rombo, al igual que los condicionales. Su estructura es una **condición cualquiera** (pueden ser varias), de modo que mientras esta se cumpla, el proceso dentro del bucle se efectuará repetidamente. Por ello, dentro del proceso de este bucle debe estar un mecanismo que haga que en algún momento la condición deje de cumplirse, para que el bucle se detenga y continúe el algoritmo.



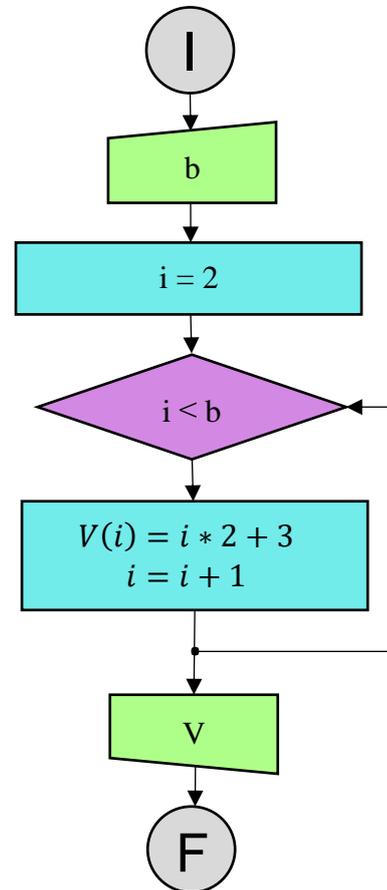
Estos bucles pueden sustituir a los “for” en casos como el ejemplo anterior:

$$V_i = i * 2 + 3 \quad i = (2, 3, 4 \dots, b)$$

Con Bucle “For”



Con Bucle “While”



Para hacer el mismo procedimiento con un bucle “while”, escribimos como su condición que la variable subíndice sea menor o igual que el valor máximo que debe tomar. Luego definimos el valor de la variable subíndice como el valor inicial, y después del proceso usual del bucle, escribimos que se le suma a la variable subíndice el valor del incremento. De este modo, el bucle se detendrá cuando la variable subíndice sea mayor que el valor final que esta debe tomar. El resultado será el mismo.