

## 1. Iteración o repetición

---

En los programas vistos hasta ahora cada instrucción se ejecutaba una sola vez, en el orden que aparece en el programa. Pero a veces es necesario repetir la ejecución de un grupo de instrucciones, para lo cual se van a utilizar las estructuras de repetición o iteración.

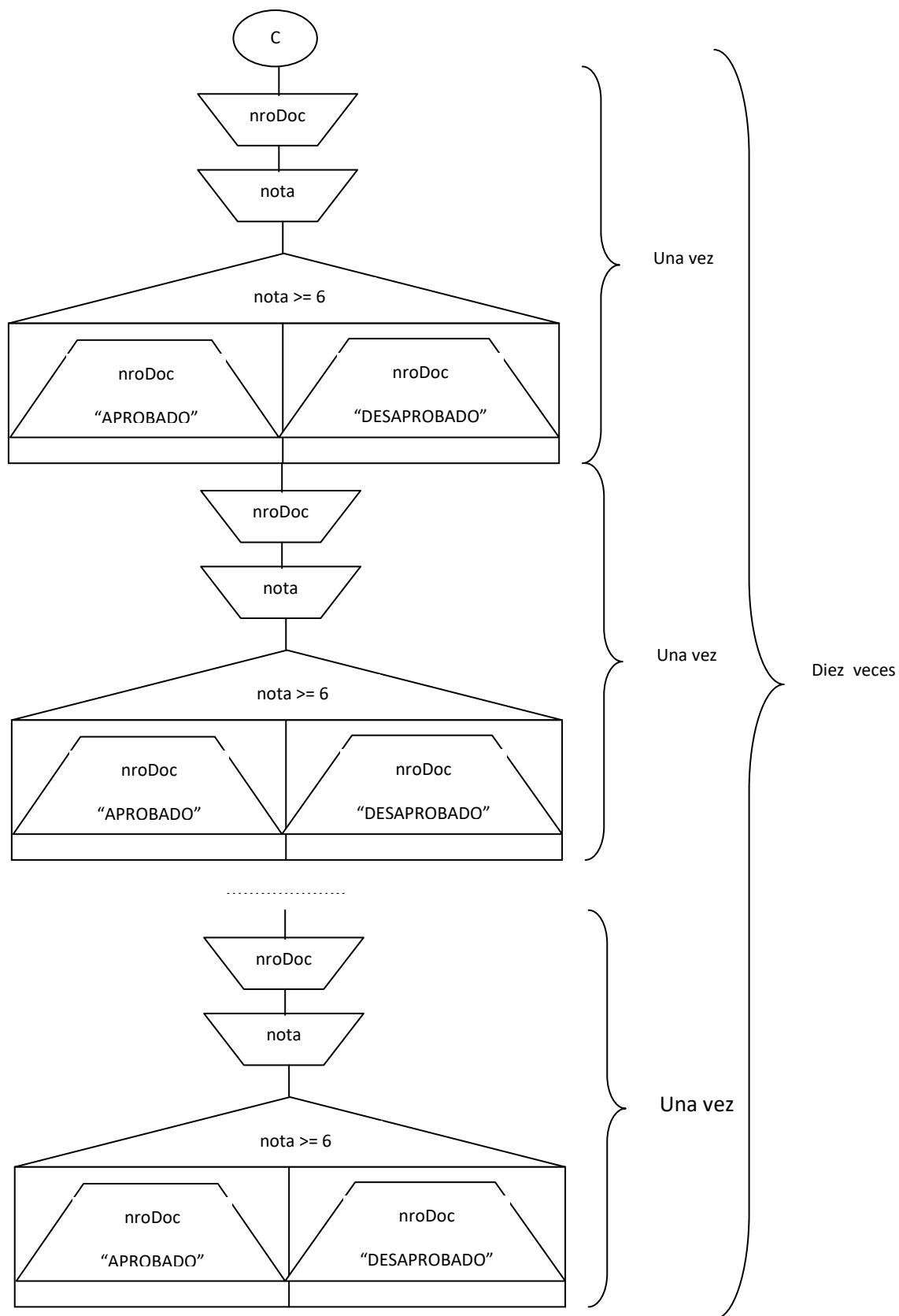
Existen dos tipos de ciclos de repetición:

- a) **ITERACIÓN DEFINIDA:** Cuando se conoce de antemano la cantidad “exacta” de veces que se debe repetir ese proceso o grupo de sentencias.
- b) **ITERACIÓN CONDICIONADA:** Cuando **NO** se conoce la cantidad de iteraciones a efectuar, es decir, que la repetición depende del cumplimiento de cierta condición.

## 2. Estructuras de iteración definida

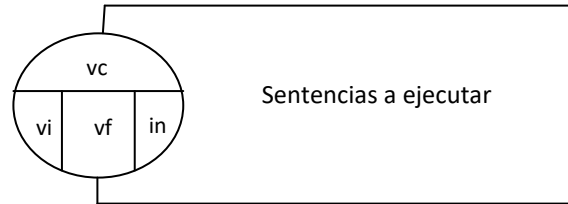
---

Se desarrolla el siguiente ejemplo: Ingresar número de documento y nota de los 10 alumnos de un curso, informar por cada uno el número de documento y la leyenda “APROBADO” o “DESAPROBADO” si la nota es mayor o igual a 6 o no.



Analizando el diagrama se observa que un grupo de instrucciones se repiten 10 veces. Para eliminar las repeticiones se debe escribir las instrucciones repetidas una sola vez y efectuar un mecanismo para realizar la repetición 10 veces.

Para ello, se debe agregar en el diagrama el ciclo de repetición definida:

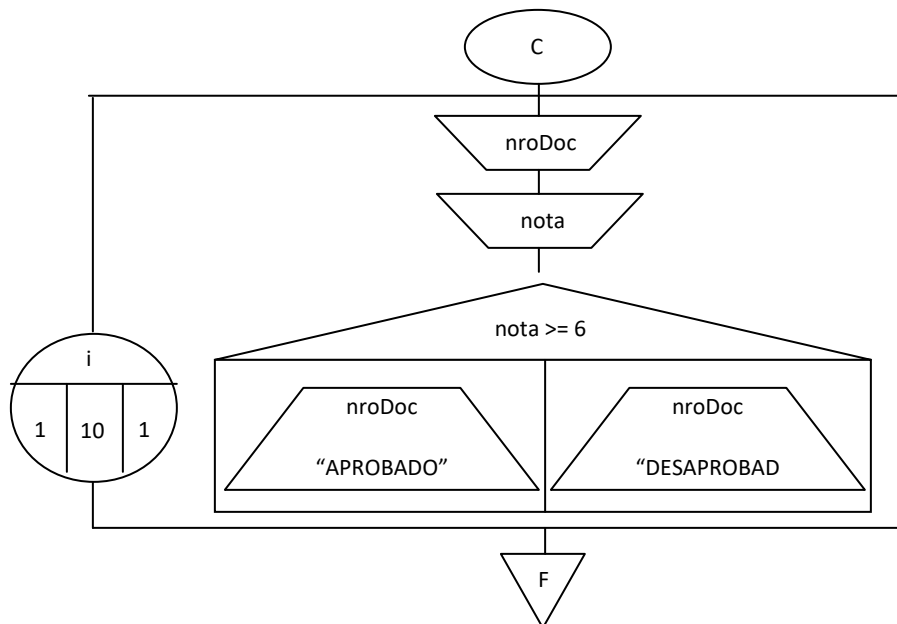


Donde:

- **vc**: es la variable de control, ya que con ella se controla el ciclo.
- **vi**: es el valor inicial que toma la variable de control.
- **vf**: es el valor final que debe tomar la variable.
- **in**: incremento que se le aplica a la variable cada vez que se ejecuta el ciclo.

**Funcionamiento:** Se asigna a la variable de control el valor inicial, se compara con el valor final, si es menor o igual que éste, se ejecutan las sentencias dentro del ciclo, luego incrementa a la variable con el valor del incremento, si sigue siendo menor o igual al valor final realiza otra ejecución y así hasta que llegue a superar el valor final.

Se aplica el siguiente diagrama al problema anterior:

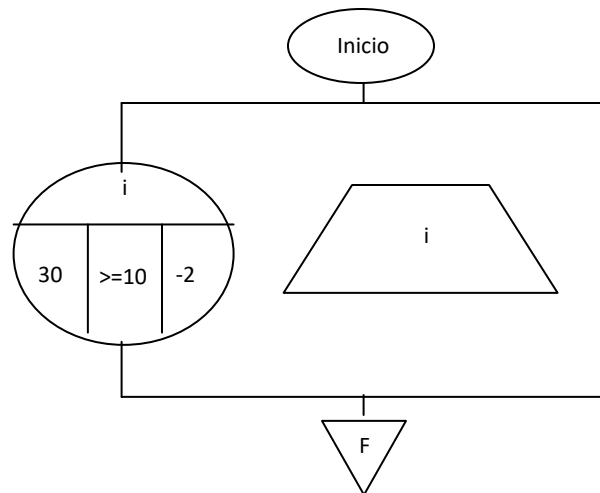


El ciclo de iteración definida no siempre necesariamente debe iniciar en 1 ni ser creciente, por ejemplo para repetir 10 veces un proceso también se puede:

- Poner como valor inicial el 10
- Poner como valor final la condición  $> 0$
- Poner -1 como el valor de incremento, generando de esta forma un decremento.

Observe entonces que en realidad el valor final no es un número sino una condición, que se denomina condición de permanencia dentro del ciclo. Si esa condición se cumple entonces el ciclo se repite. Tampoco el incremento o decremento debe ser siempre 1 o -1 sino que puede ser cualquier otro valor que modifique la variable de control.

**Ejemplo:** Realizar un programa para mostrar los números pares entre 30 y 10 en forma descendente.



### 3. Contadores y acumuladores

Un contador es una variable en la memoria que se incrementará en una unidad cada vez que se ejecute el proceso. El contador se utiliza para llevar la cuenta de determinadas acciones que se pueden solicitar durante la resolución de un problema. Hay que realizar la inicialización del contador o contadores. La inicialización consiste en poner el valor inicial de la variable que representa al contador (generalmente con el valor 0).

Un acumulador es una variable que suma sobre sí misma un conjunto de valores, para de esta manera tener la suma de todos ellos en una sola variable. De igual forma, se pueden efectuar decrementos en un acumulador.

La diferencia entre un contador y un acumulador es que mientras el primero va aumentando de uno en uno, el acumulador va aumentando o decrementando en una cantidad variable.

**Ejemplo:** Ingresar 50 números, informar la cantidad de números negativos y la sumatoria de los positivos.

Se deben ingresar los números en un ciclo de repetición definido. Por cada número se tiene que consultar si es menor que cero o no. Si es menor que cero se cuenta utilizando un contador (incrementando el contador en uno), si no se suma en un acumulador (sumarle al acumulador ese número).

