

Técnico Universitario en Programación

UNIVERSIDAD TECNOLÓGICA NACIONAL

Facultad Regional Gral. Pacheco

Apuntes de clase de la asignatura

Programación I

CICLO EXACTO

2024

Abel Oscar Faure

Lorena Raquel Palermo



CICLOS REPETITIVOS

Durante la solución de problemas, es común encontrar operaciones que se pueden ejecutar un número determinado de veces. Si bien las instrucciones son las mismas, los datos varían. El conjunto de instrucciones que se ejecuta repetidamente recibe el nombre de ciclo.

Los ciclos repetitivos en programación son estructuras de control que permiten ejecutar un bloque de instrucciones varias veces de manera automática. Este tipo de estructuras es fundamental para realizar tareas repetitivas sin necesidad de escribir la misma instrucción múltiples veces.

Tipos de Ciclos Repetitivos:

- **Ciclo Exacto.**
- **Ciclo Inexacto.**

Veamos el siguiente problema resuelto sin estructuras repetitivas:

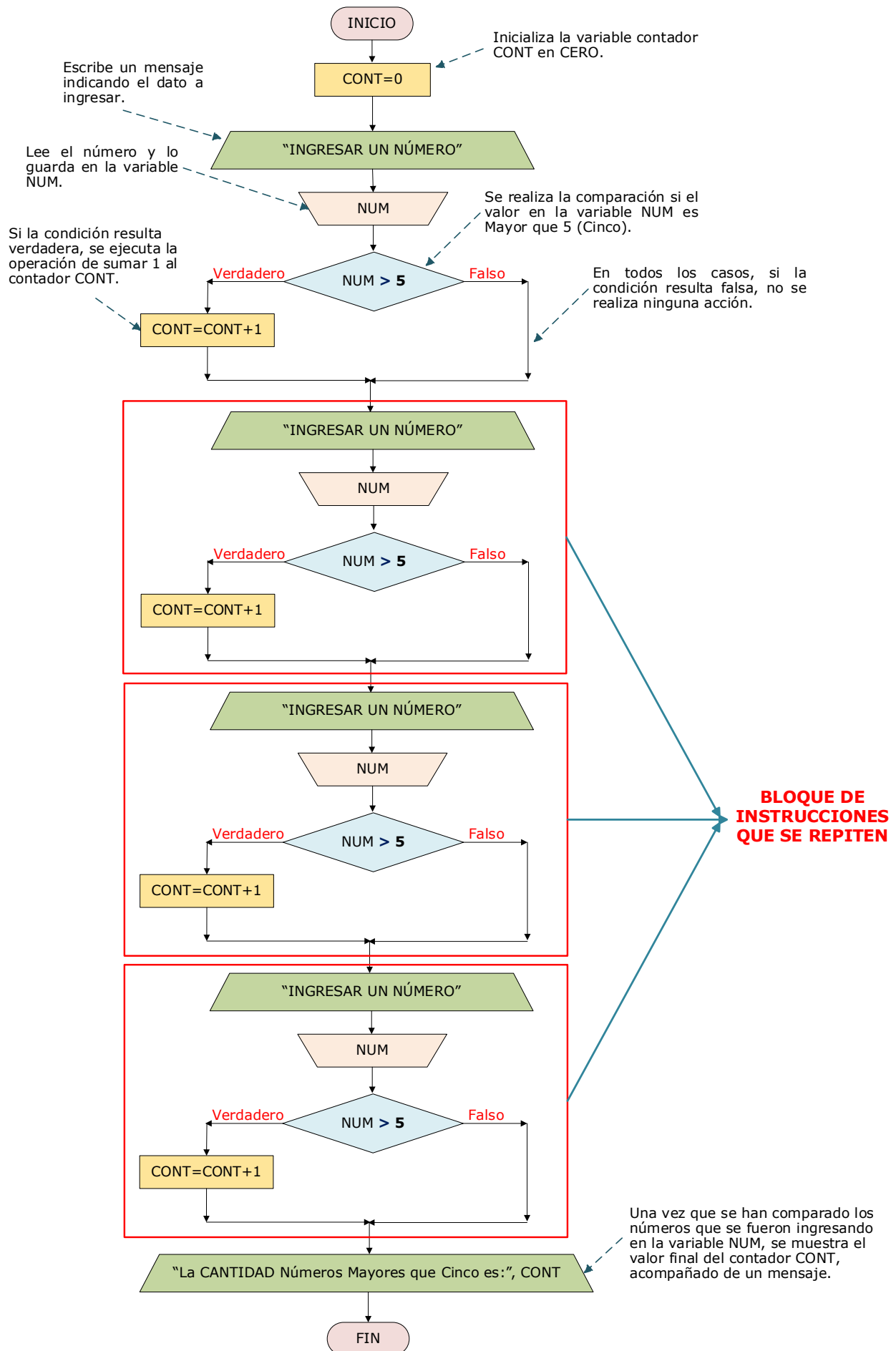
Se requiere implementar un algoritmo que lea cuatro números enteros y, mediante un mensaje, informe cuántos de ellos son mayores que cinco. Representar el algoritmo utilizando un diagrama de flujo.

Cuando explicamos contadores, resolvimos este mismo ejercicio. Sin embargo, ahora realizaremos una variante al algoritmo: en lugar de utilizar cuatro variables para el ingreso de datos, solo utilizaremos una, llamada NUM.

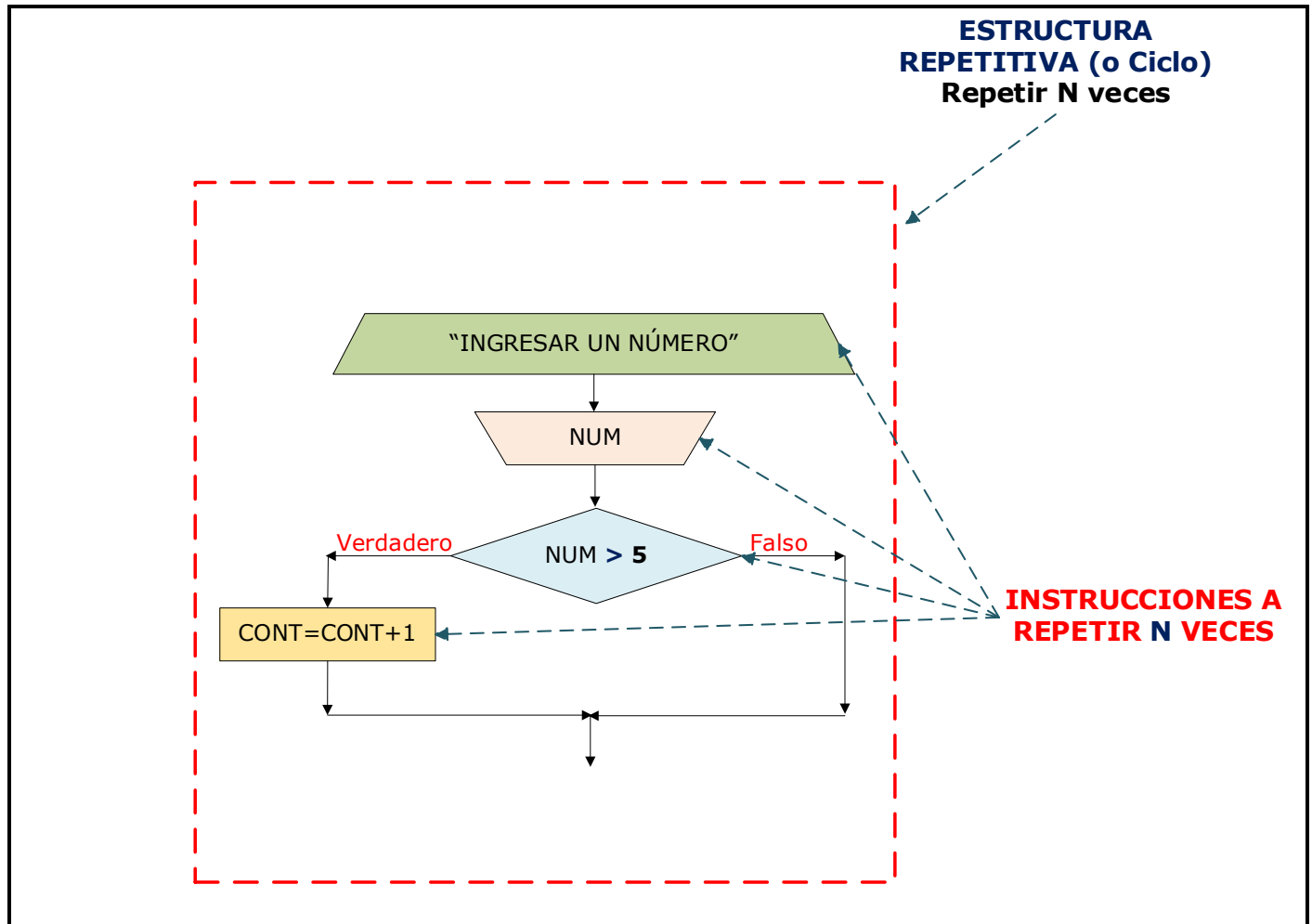
El algoritmo funciona de la siguiente manera: solicita al usuario que ingrese un número y lo guarda en la variable NUM. Luego, compara el valor almacenado en NUM para verificar si es mayor que cinco. Si esta condición es verdadera, se incrementa el contador CONT en uno; de lo contrario, no se realiza ninguna acción.

Después de la estructura selectiva, se solicita nuevamente al usuario que ingrese otro número, el cual se guarda en la variable NUM, sobrescribiendo el valor anterior. Se repite el proceso de comparación: si el nuevo valor en NUM es mayor que cinco, se incrementa el contador CONT; si no lo es, no se realiza ninguna acción. Este ciclo de ingreso de datos, comparación, y posible incremento del contador se repite para cada uno de los números restantes.

En el algoritmo, los bloques de instrucciones que se repiten están marcados con un recuadro rojo. Finalmente, una vez que se han completado todas las comparaciones, se muestra en pantalla el valor final del contador CONT.



Es en este punto donde entran en juego las estructuras repetitivas o ciclos. Estas permiten tomar un bloque de instrucciones que se repiten y colocarlo dentro de una estructura repetitiva, de manera que el bloque se ejecute automáticamente tantas veces como sea necesario.

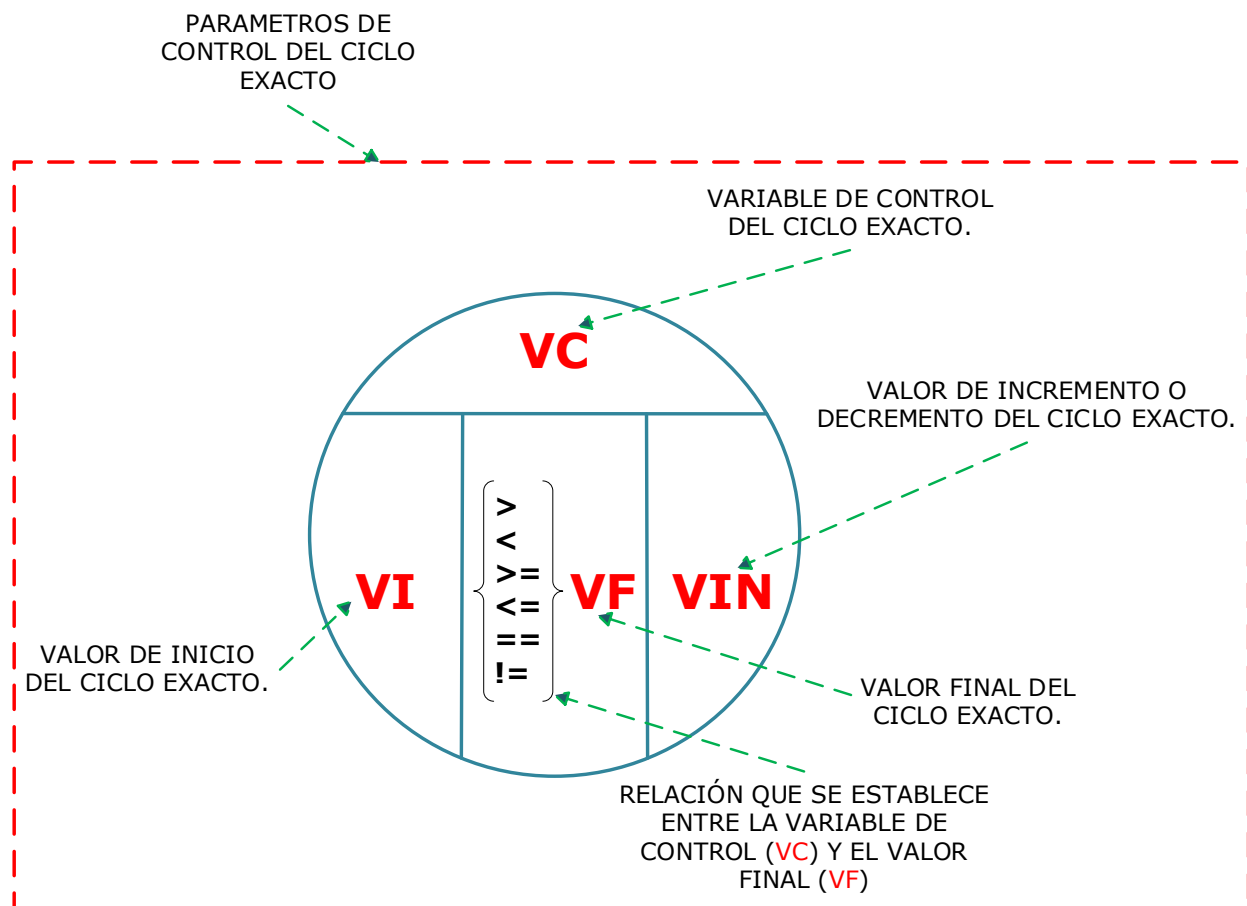
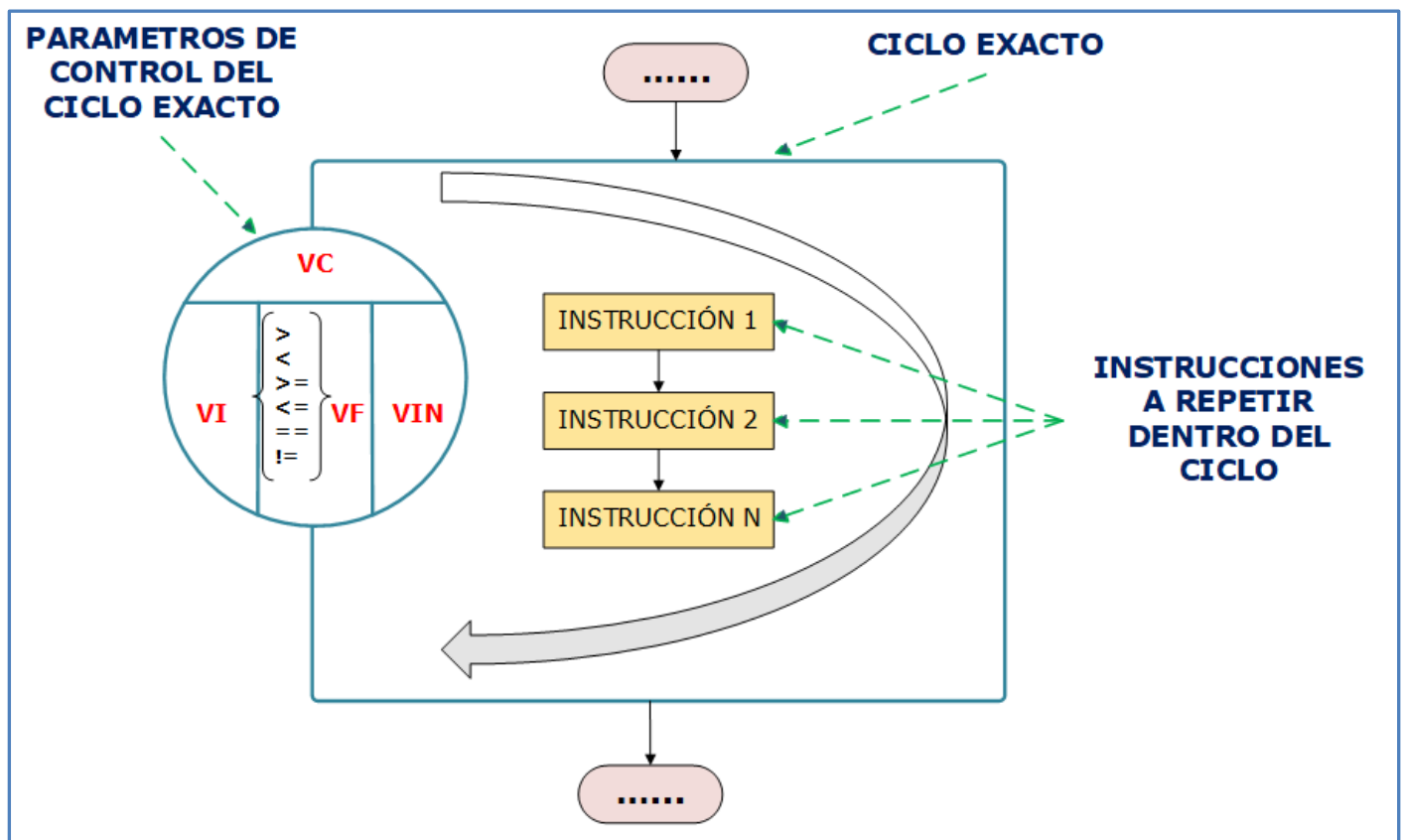


CICLO EXACTO

Un ciclo exacto es una estructura de control repetitiva utilizada en programación cuando se sabe de antemano el número exacto de iteraciones que se deben realizar.

Este tipo de ciclo ejecuta un bloque de instrucciones un número específico de veces, asegurando que la cantidad de repeticiones es fija y no depende de condiciones evaluadas durante la ejecución.

La principal característica es que el número de veces que el ciclo se repetirá es conocido antes de que el ciclo comience.





Un ciclo exacto generalmente consta de componentes clave para su funcionamiento:

Variable de Control (VC)

El ciclo exacto generalmente está controlado por una variable tipo contador, que se incrementa o decrementa en cada iteración hasta alcanzar un valor límite predefinido. Esta variable es la encargada de determinar cuándo debe finalizar la ejecución del ciclo.

Valor de Inicio (VI)

Asigna un valor inicial a la variable de control del ciclo, es decir, establece el valor inicial del contador.

Valor Final (VF)

Representa el valor final que la variable de control, es decir, el contador, debe alcanzar.

Operador Relacional ($>$, $<$, $>=$, $<=$, $==$, $!=$)

Representa el operador que establece la relación entre el valor de la variable de control (contador) y el valor final. Este operador permite comprobar en cada iteración si se ha alcanzado el valor final para detener el ciclo.

Valor de Incremento o Decremento (VIN)

Representa el valor por el cual se incrementa o decrementa la variable de control (contador) en cada iteración. Esto garantiza que la variable de control avance hacia el valor final del ciclo. Es fundamental que este valor permita alcanzar el valor final para evitar que el ciclo se ejecute indefinidamente, lo que produciría un ciclo infinito.

Se puede observar que los valores que va tomando la variable de control del ciclo (VC) se desplaza desde el valor inicial (VI) hasta el valor final (VF). En cada iteración del ciclo exacto, el valor de la variable de control (VC) se incrementa o decrementa según el valor de incremento (VIN), dependiendo de si el ciclo es ascendente o descendente.

Reglas para tener en cuenta:

- Los valores inicial y final pueden ser expresiones aritméticas, variables, constantes o valores literales.

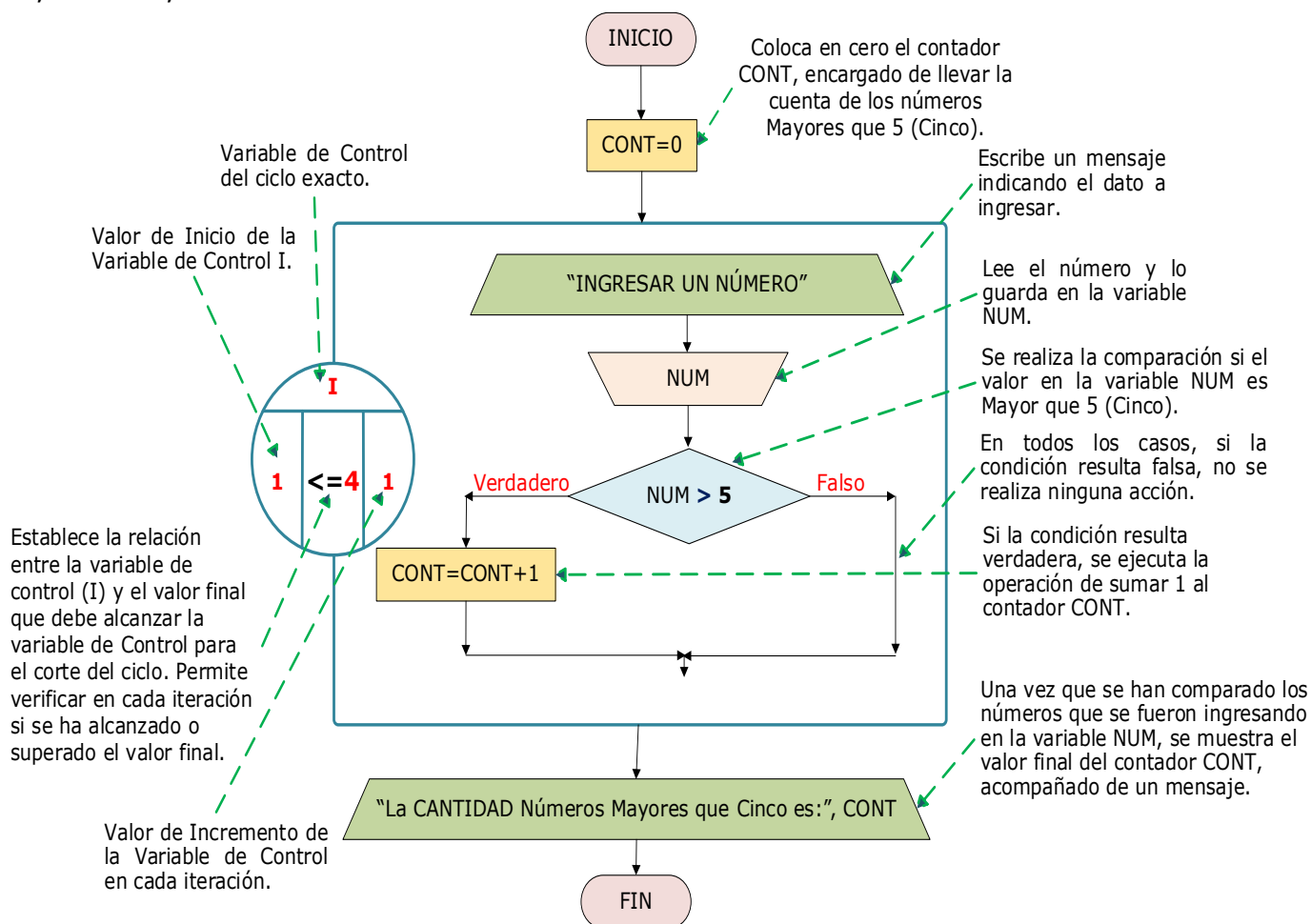
- Evitar modificar la variable de control, el valor inicial y el valor final dentro del bucle, a menos que el algoritmo lo requiera explícitamente.
- Al igual que en las estructuras selectivas, las estructuras repetitivas pueden anidarse.

Ejemplo 1:

Se requiere implementar un algoritmo que lea cuatro números enteros y, mediante un mensaje, informe cuántos de ellos son mayores que cinco. Representar el algoritmo utilizando un diagrama de flujo.

Este problema fue analizado al comienzo del tema, pero en esa ocasión se resolvió sin utilizar un ciclo exacto. Ahora, el objetivo es permitir el ingreso de cuatro números y contar cuántos de ellos son mayores que cinco.

Para lograr esto, utilizaremos un ciclo que se repetirá cuatro veces. En cada iteración, se solicitará un número y se comparará para determinar si es mayor que cinco; si lo es, se incrementará un contador.





Lógica del Algoritmo

Se inicializa el contador CONT en cero ($CONT = 0$) para llevar la cuenta de cuántos números son mayores que cinco. El ciclo exacto se utiliza para iterar exactamente cuatro veces, ya que sabemos que debemos leer y procesar cuatro números.

La variable de control I comienza en 1 y finaliza al alcanzar el valor 5. Esto garantiza que el ciclo se ejecute cuatro veces, ya que el ciclo se detiene cuando I llega a 5, habiendo completado cuatro iteraciones.

En cada iteración, se lee un número y se verifica si es mayor que cinco. Si es así, se incrementa el contador CONT. Luego, la variable de control I se incrementa en uno, y se comprueba si ha alcanzado o superado el valor final. Esta verificación se realiza evaluando la relación $I \leq 4$, asegurando que el ciclo continúe mientras el valor de I sea menor o igual a 4.

Al finalizar el ciclo, se muestra un mensaje indicando cuántos de los números ingresados son mayores que cinco.

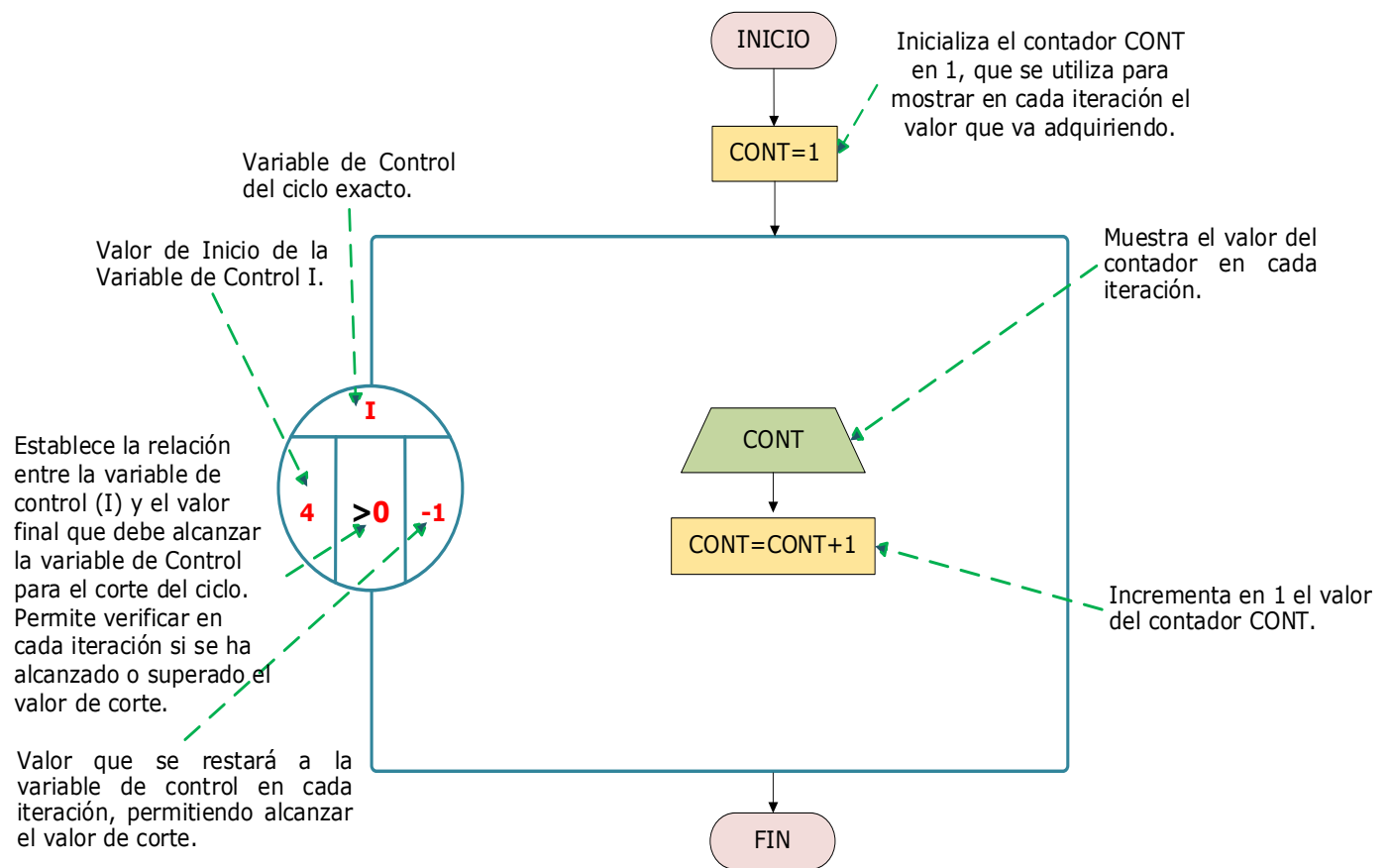
Este ejemplo permite entender claramente cómo utilizar un ciclo exacto para resolver un problema específico, asegurando que el ciclo se ejecute un número determinado de veces y se realicen las operaciones necesarias en cada iteración.

Ejemplo 2:

Se requiere implementar un algoritmo que muestre en pantalla los números del 1 al 4. Como restricción, el algoritmo no debe incluir la entrada de datos. Representar el algoritmo utilizando un diagrama de flujo.

El objetivo es mostrar en pantalla los números del 1 al 4, con la restricción de que el algoritmo no debe incluir entrada de datos por parte del usuario. Es decir, al simular la ejecución del algoritmo, los números se mostrarán automáticamente sin requerir ninguna acción del usuario.

Para lograrlo, utilizaremos un ciclo exacto que se repetirá cuatro veces. En cada iteración, se mostrará el valor de un contador que se incrementará para cumplir con este propósito.



Lógica del Algoritmo

Se inicializa el contador CONT en 1 ($CONT = 1$), ya que este contador se utilizará para mostrar los números del 1 al 4.

El ciclo exacto se utiliza para iterar exactamente cuatro veces. La variable de control I comienza en 4 y se decrece en cada iteración hasta alcanzar 0, momento en el cual el ciclo se detiene. Esto asegura que el ciclo se ejecute con los valores 4, 3, 2, y 1, evitando un bucle infinito.

En cada iteración, se muestra el valor actual de CONT en pantalla, después, se incrementa CONT en 1.

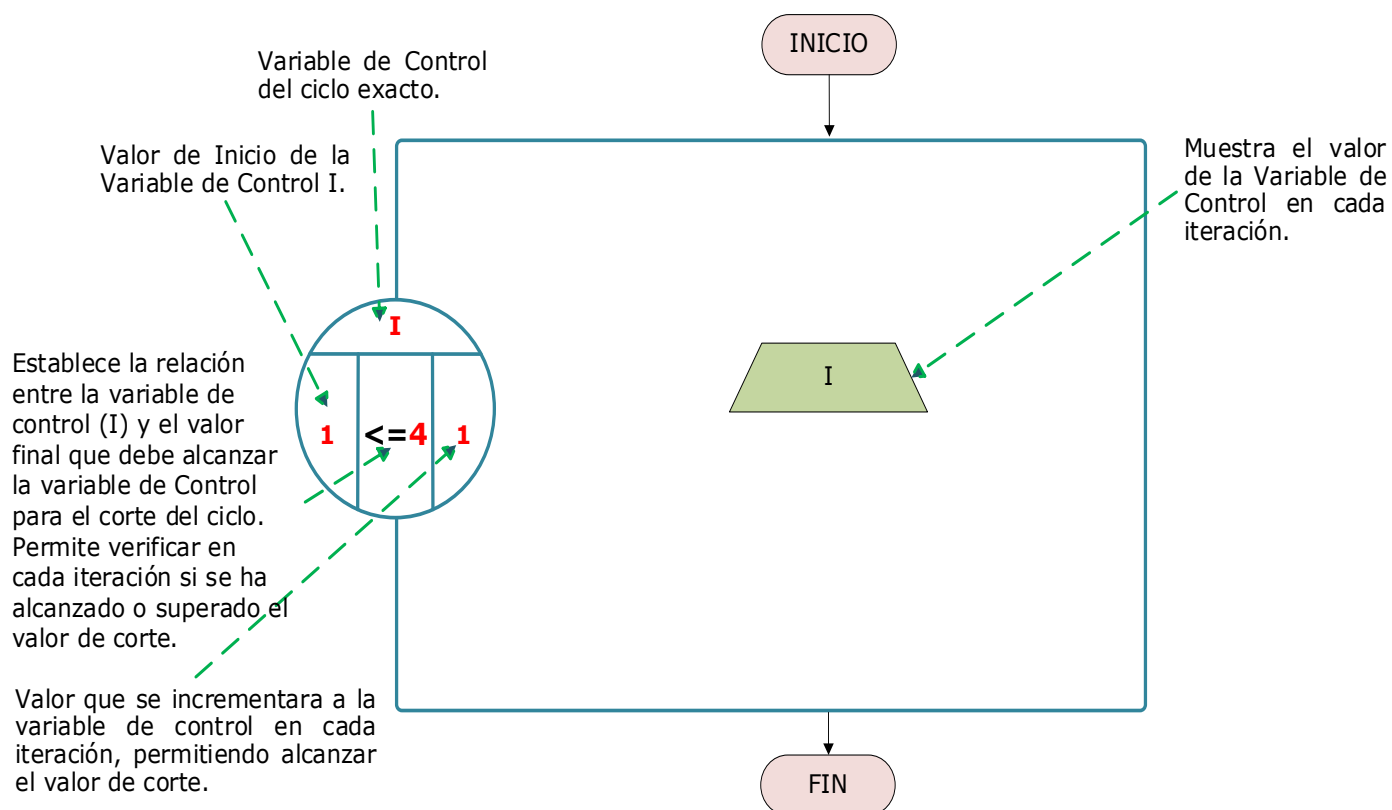
Una vez que se ejecutaron todas las instrucciones dentro del ciclo exacto, se decrementa la variable de control I en 1. El ciclo continúa mientras I sea mayor que 0, garantizando las cuatro iteraciones necesarias.

Al finalizar el ciclo, el algoritmo habrá mostrado en pantalla los números del 1 al 4.

Para resolver este problema, también podríamos prescindir del contador independiente CONT y utilizar únicamente la variable de control del ciclo, I. Esta variable es responsable de llevar la cuenta de las iteraciones, por lo que podríamos configurarla de manera que su valor inicial sea 1 y su valor final sea 4. El ciclo funcionaría mientras I tenga un valor menor o igual a 4, deteniéndose cuando I alcance 5.

En la solución anterior, la variable de control I iba de 4 a 1 y el ciclo se detenía al llegar a 0. Aunque también permitía cuatro iteraciones, los valores de I en ese caso no eran útiles para mostrar los números del 1 al 4 en pantalla.

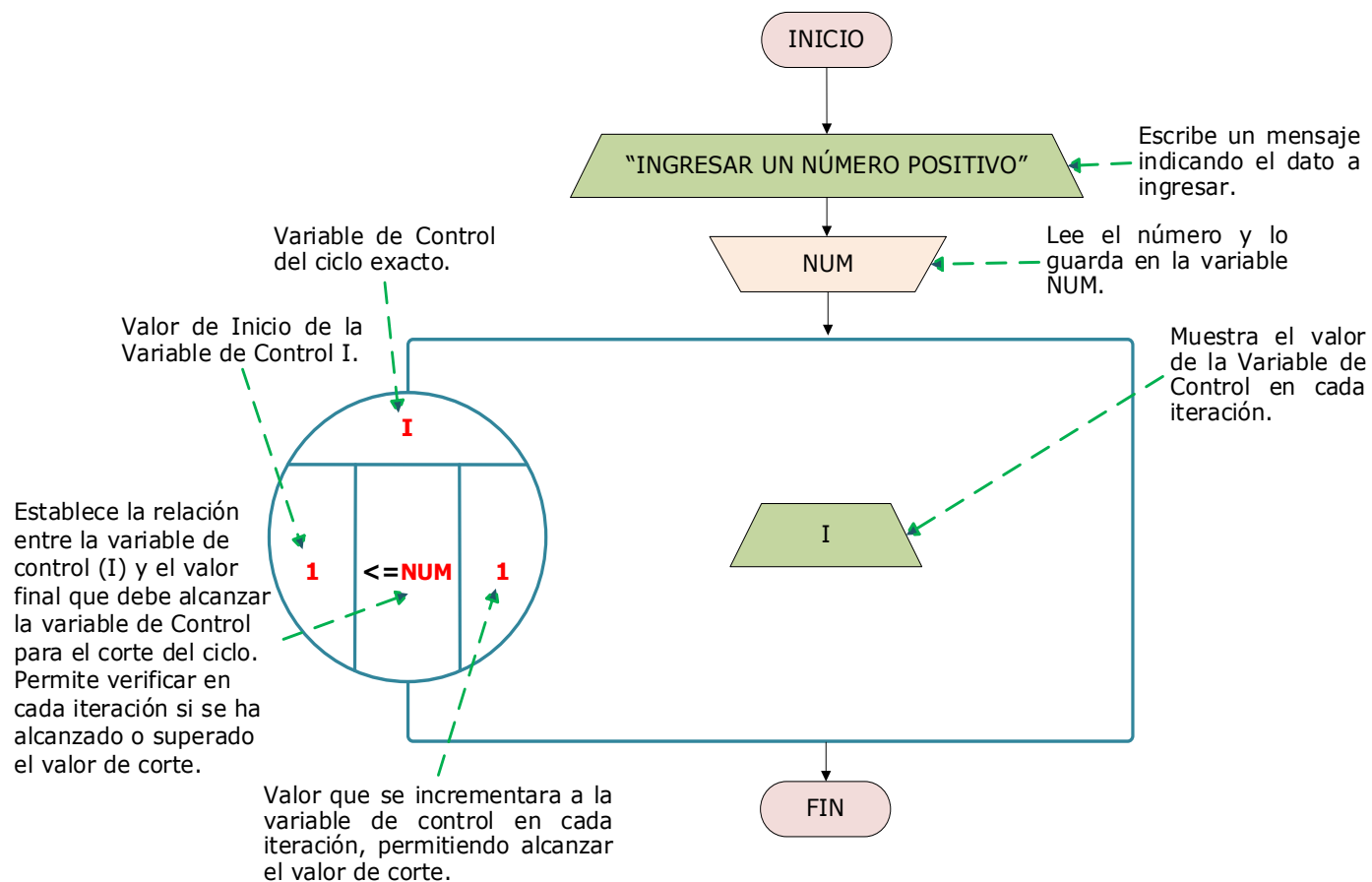
Esta alternativa, utilizando la variable de control I, permite que los valores de I (1, 2, 3, 4) se utilicen directamente para mostrar los números en pantalla, haciendo innecesario un contador independiente.



Ejemplo 3:

Se requiere implementar un algoritmo para que solicite el ingreso de un número positivo y luego se muestren por pantalla los números entre el 1 y el número ingresado. Representar el algoritmo utilizando un diagrama de flujo.

El objetivo es solicitar al usuario que ingrese un número positivo y luego mostrar en pantalla todos los números desde el 1 hasta el número ingresado.



El algoritmo comienza solicitando al usuario que ingrese un número, asumiendo que siempre será positivo, por lo que no se realiza ninguna validación.

Se utiliza un ciclo exacto que iterará desde 1 hasta el número ingresado. La variable de control **I**, que se inicializa en 1, será utilizada para contar y mostrar los números desde 1 hasta el número ingresado. En cada iteración, se muestra el valor de **I** en pantalla.

Después de mostrar el valor de **I**, y dado que no hay más instrucciones dentro del ciclo, **I** se incrementa automáticamente en 1.

El ciclo continuará hasta que **I** sea mayor que el número ingresado, momento en el cual finalizará.

Al concluir el ciclo, el algoritmo habrá mostrado todos los números desde 1 hasta el número ingresado por el usuario.



Ejemplo 4:

Se desea implementar un algoritmo que genere y muestre en pantalla la tabla de multiplicar del 1 al 10. El algoritmo debe funcionar sin requerir entrada de datos por parte del usuario. Representa el algoritmo mediante un diagrama de flujo.

El objetivo es mostrar la tabla de multiplicar para los números del 1 al 10, sin necesidad de que el usuario ingrese ningún dato. La salida debe presentar los productos de multiplicar los números del 1 al 10. Por ejemplo:

Multiplicaciones del 1:

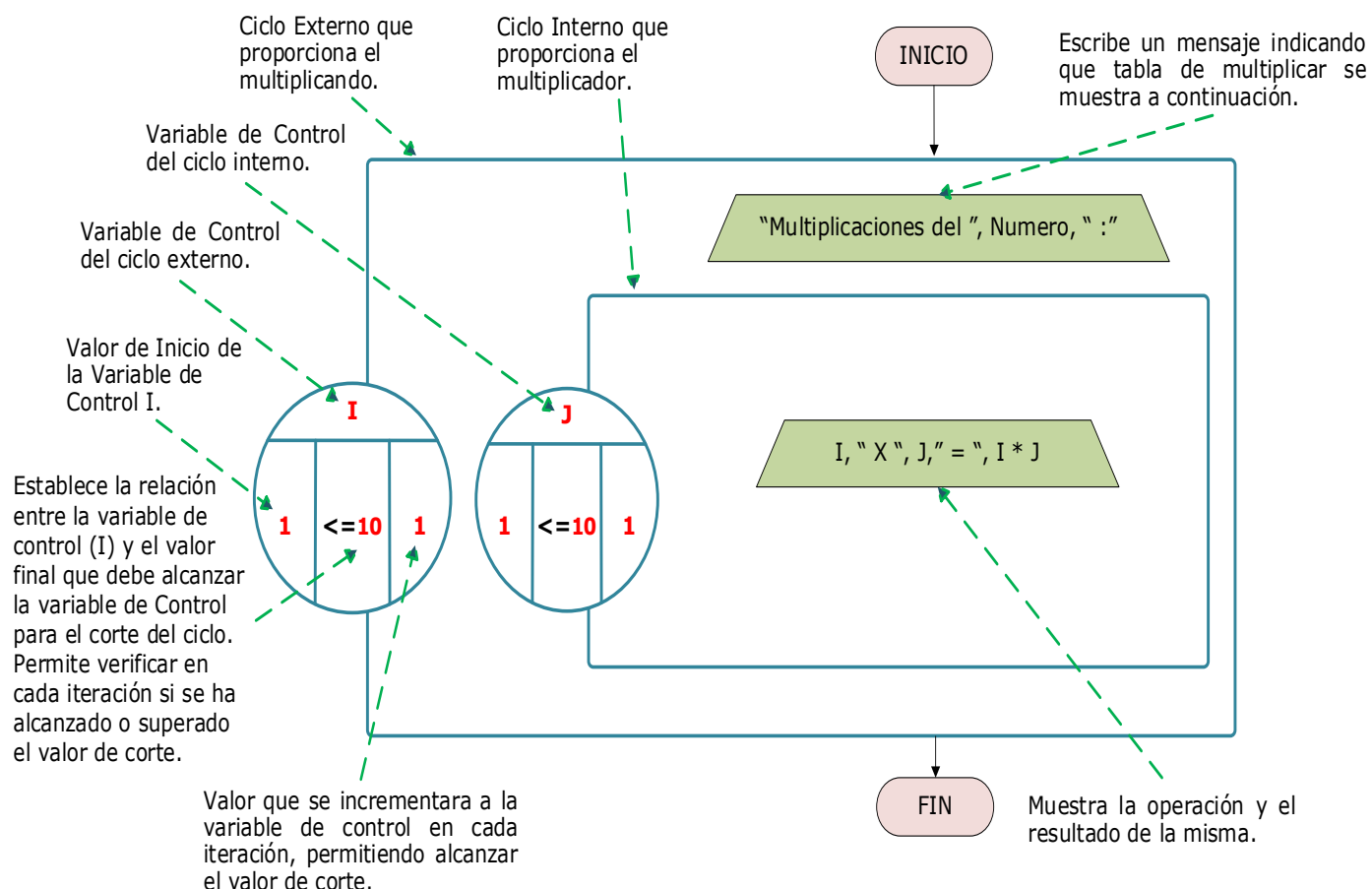
$1 \times 1 = 1$
 $1 \times 2 = 2$
 $1 \times 3 = 3$
 $1 \times 4 = 4$
 $1 \times 5 = 5$
 $1 \times 6 = 6$
 $1 \times 7 = 7$
 $1 \times 8 = 8$
 $1 \times 9 = 9$
 $1 \times 10 = 10$

Multiplicaciones del 2:

$2 \times 1 = 2$
 $2 \times 2 = 4$
 $2 \times 3 = 6$
 $2 \times 4 = 8$
 $2 \times 5 = 10$
 $2 \times 6 = 12$
 $2 \times 7 = 14$
 $2 \times 8 = 16$
 $2 \times 9 = 18$
 $2 \times 10 = 20$

Multiplicaciones del 3:

.....
.....
.....



El algoritmo inicia sin requerir datos de entrada del usuario. El ciclo exterior itera sobre los números del 1 al 10, representando el número base de la tabla de multiplicar. La variable de control de este ciclo, llamada I, se inicializa en 1. Dentro de cada iteración del ciclo exterior, se muestra un mensaje que indica el número para el cual se generará la tabla de multiplicar.

A continuación, se ejecuta un ciclo interior que también itera sobre los números del 1 al 10, representando el multiplicador. La variable de control de este ciclo, llamada J, también se inicializa en 1. En cada iteración del ciclo interior, se calcula el producto de I por J ($I * J$) y se muestra en pantalla.

Una vez que el ciclo interior completa todas sus iteraciones para el valor actual de I, el ciclo exterior incrementa I, muestra nuevamente un mensaje indicando el número sobre el cual se generará la siguiente tabla de multiplicar, y el ciclo interior comienza de nuevo para el siguiente valor de I. Este proceso se repite hasta que ambos ciclos han completado todas sus iteraciones, cubriendo así las tablas de multiplicar del 1 al 10.

Las variables de control de ambos ciclos (I y J) terminan con el valor 11, que es el valor de corte para ambos ciclos. Esto ocurre porque ambos ciclos funcionan mientras las variables de control I y J tengan un valor menor o igual a 10. Una vez que



alguna de estas variables alcanza el valor 11, el ciclo correspondiente se detiene, ya que ha completado todas las iteraciones previstas.