

TECNICATURA UNIVERSITARIA EN

Programación I

Estructuras Algorítmicas Repetitivas:
ciclo FOR



Índice

Objetivos:.....	2
Contenidos:	2
Introducción	3
La estructura repetitiva CICLO EXACTO (CICLO FOR)	7
Bibliografía.....	11



Objetivos:

- Conocer la terminología relacionada con los algoritmos de estructuras repetitivas, así como la importancia de aplicar técnicas adecuadas de programación.
- Conocer la metodología de las estructuras repetitivas (Ciclo Exacto).

Contenidos:

- Introducción a las estructuras repetitivas
- Estructura repetitiva Ciclo Exacto

Introducción

Antes de avocarnos de lleno al uso de estructuras repetitivas, tomaremos un problema de ejemplo correspondiente al Tema 2, para reflejar de esta manera las ventajas de la utilización de estructuras repetitivas.

Suponiendo que tuviéramos que resolver el siguiente ejercicio:

Hacer un programa que permita ingresar 3 números enteros y que luego se informe por pantalla aquellos que sean mayores a cinco.

Con los conocimientos adquiridos hasta el momento, el Diagrama de Flujo sería el siguiente:

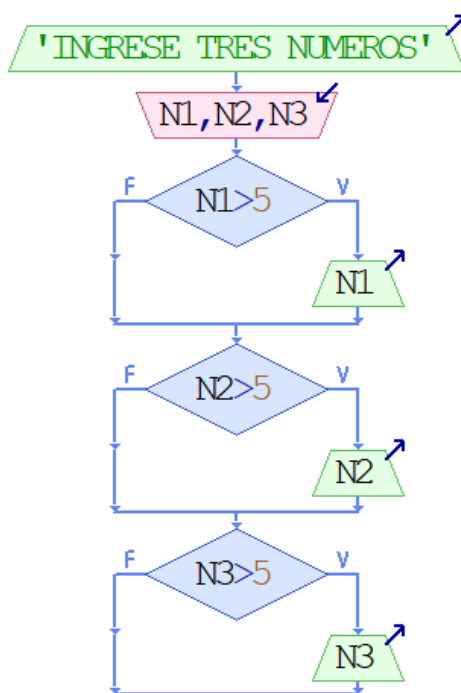


Figura 1

En el diagrama de flujo anterior para la resolución del problema, utilizamos 3 variables para alojar los números enteros que solicitara el programa (N1, N2 y N3), luego realiza la comparación del primer número alojado en la variable N1, evaluando la relación si el valor de la variable es mayor a 5, en caso de que así sea, muestra el número por pantalla, caso contrario no hace nada y continua con el siguiente número a comparar, así hasta el tercer número.

Otra forma de resolver el problema sería el representado por la siguiente figura:

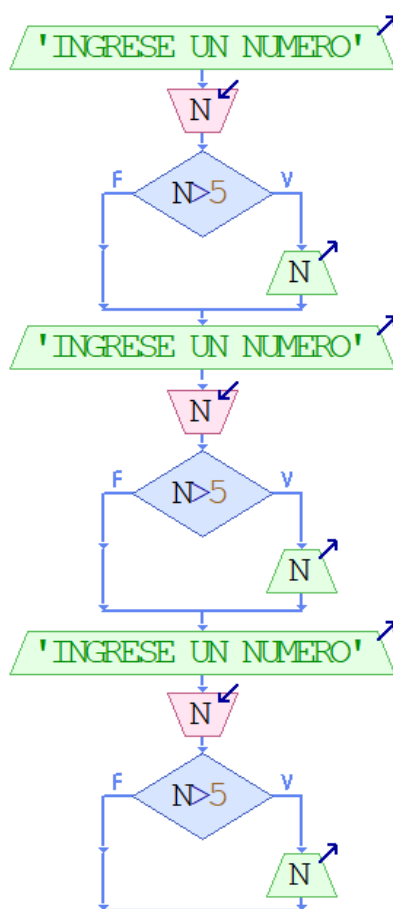


Figura 2

En este caso el programa solicita que se ingrese el primer número y se guarda en la variable de nombre N. Luego realiza la comparación del primer número alojado en la variable N, evaluando la relación si el valor de la variable es mayor a 5, en caso de que así sea, muestra el número por pantalla, caso contrario no hace nada.

Luego solicita el ingreso del segundo número y se guarda en la variable de nombre N, borrando de esta manera el valor anterior de la variable. Luego realiza la comparación del segundo número alojado en la variable N, evaluando la relación si el valor de la variable es mayor a 5, en caso de que así sea, muestra el número por pantalla, caso contrario no hace nada.

Se realizan los mismos pasos para el tercer número.

De esta manera utilizamos una sola variable (N) para alojar los tres números enteros que solicitara el programa.

En el diagrama de flujo anterior, vemos que la solicitud de ingreso de un número, la captura de dicho número y la estructura de decisión, se REPITE para cada uno de los números que se ingresan. De esta manera podríamos tomar una parte del Diagrama de Flujo anterior (Figura 2) y repetirla la cantidad de veces que deseemos (Para 3, 10, 50, . . . , N números). **Así es como aparece la necesidad de utilizar estructuras repetitivas.**

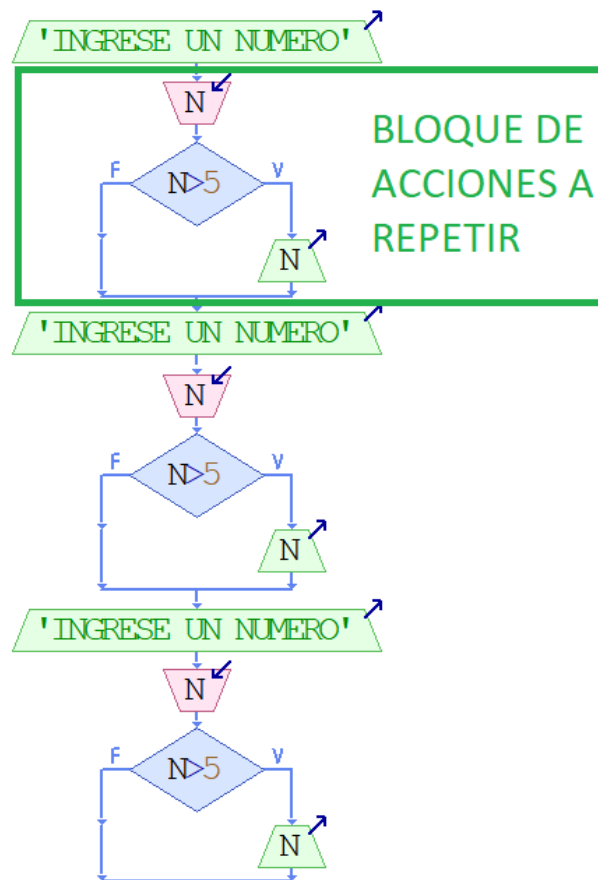


Figura 3

En la práctica, 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**.

Todo ciclo debe terminar luego de repetirse un número finito de veces. Dentro del conjunto de instrucciones siempre debe existir una condición de parada o fin de ciclo. En cada



iteración de este son evaluadas las condiciones necesarias para decidir si se debe seguir ejecutando o si debe detenerse.

En algunos algoritmos podemos establecer de antemano el número de veces que se debe repetir el ciclo. En este caso, el número de repeticiones no depende de las proposiciones dentro de ciclo. **La estructura algorítmica repetitiva CICLO EXACTO se utiliza para resolver problemas en los que conocemos el número de veces que se puede repetir el ciclo.**

Por otra parte, en algunos algoritmos no podemos establecer de antemano el número de veces que se debe repetir el ciclo. Este número depende de las condiciones que contenga el mismo. La estructura repetitiva CICLO INEXACTO se utiliza para resolver este tipo de problemas y estudiaremos su uso en el tema próximo.

En este tema, estudiaremos una de las estructuras algorítmicas repetitivas:

— Estructura Repetitiva CICLO EXACTO.

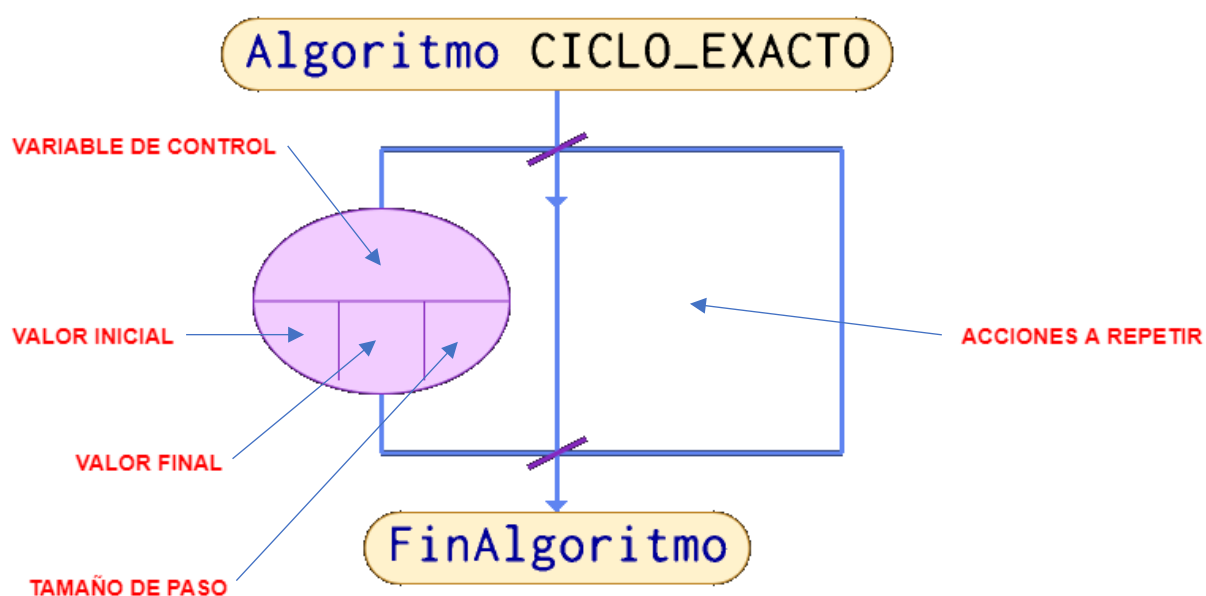
La estructura repetitiva CICLO EXACTO (CICLO FOR)

Esta es la estructura algorítmica utilizada para **repetir un conjunto de instrucciones un número finito de veces**. Es decir que el programador, al momento de programar **conoce de antemano el número exacto de veces que se repetirá el ciclo**.

Reglas para tener en cuenta

1. Las variables de control, valor inicial y valor final deben ser todas del mismo tipo, pero el tipo real no está permitido. Los valores inicial y final pueden ser tanto expresiones como constantes.
2. No intentar modificar el valor de la variable de control, valor inicial y el valor final dentro del bucle (ciclo), salvo que el programa específicamente lo requiera.

Al igual que en las sentencias de selección, las sentencias repetitivas se pueden anidar. La representación del **CICLO EXACTO** en el diagrama de flujo es el siguiente:



Donde:

Variable de Control: Representa la **variable de control** del ciclo.

Valor inicial: Representa el **valor inicial** del ciclo.

Valor final: Representa el **valor final** del ciclo.

Tamaño de paso: Representa el valor de incremento o decremento.

Acciones a repetir: se trata del bloque de instrucciones que se repiten mientras dure el ciclo.

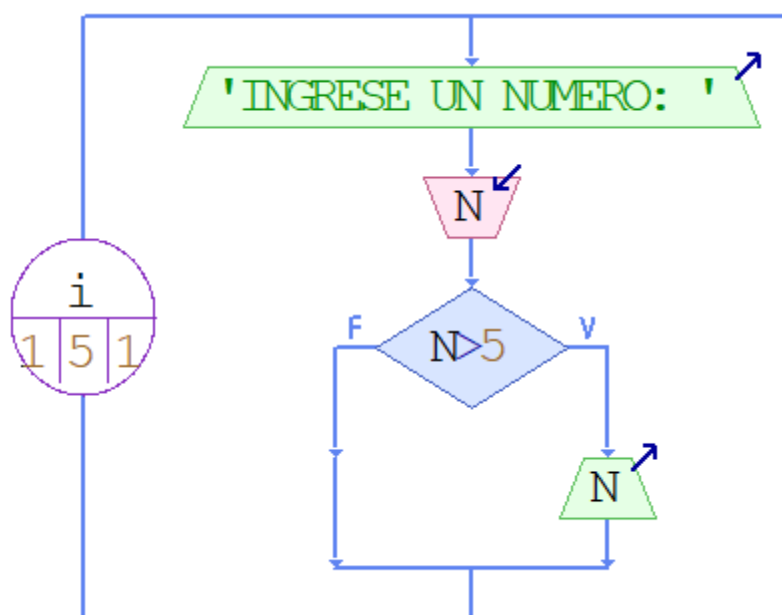
Durante la ejecución del CICLO EXACTO:

- El primer paso es asignarle a la **variable de control** el **valor de inicio**.
- El segundo paso es comparar **el valor de la variable** de control con el **valor final**.
- En esta **comparación** se establece la relación entre el valor de la variable de control y el valor final. El ciclo se ejecuta mientras el valor de la variable de control sea mayor, menor, mayor igual o menor igual al valor final. **NOTA: En PSeInt la evaluación que se hace es IF (valor variable de control <= valor final).**
- El tercer paso es ejecutar **las acciones a repetir** dentro del ciclo si este no se pasa del valor final.
- El cuarto paso es que, una vez ejecutadas todas las acciones dentro del ciclo, es realizar el incremento o decremento de la variable de control de acuerdo si el ciclo es ascendente o descendente. El valor de incremento o decremento en cada iteración esta dado por el valor del **tamaño de paso**.

Ejemplo

Hacer un programa que permita ingresar 5 números enteros y que luego se informe por pantalla aquellos que sean mayores a cinco.

Diagrama de flujo



Análisis del diagrama de flujo

Para la resolución de este problema debemos realizar un ciclo exacto que permita dar 5 iteraciones o ciclos, y así poder ingresar en cada iteración un número y evaluar si el mismo es mayor a cinco o no.

Por lo tanto, declaramos a **i** como nuestra **variable de control** del CICLO EXACTO.

El ciclo exacto comienza con un **valor de inicio 1** y se establece **valor final** en 5: La relación entre el valor de la **variable de control i** y el **valor final** es de \leq [lo hace internamente PSeInt, por eso no lo vemos]. Quiere decir que se van a permitir iteraciones mientras el valor de la variable de control **i** tenga un valor menor o igual al valor final.

Por cada iteración (vuelta) del ciclo exacto la variable de control se incrementará en 1, que es el valor del **tamaño de paso**. Esto posibilita que la variable de control **i** en algún momento supere al valor final establecido y el ciclo exacto termine.

Dentro del grafico del ciclo exacto, ingresamos las **acciones a repetir**: solicitamos al usuario un valor **N** e inmediatamente lo evaluamos. Si el número ingresado es mayor a 5 lo mostramos por pantalla. Caso contrario no se hace nada.

Ejemplo tabla de datos

Vuelta del ciclo	i (variable de control)	Resultado de la decisión ($i \leq 5$) (donde 5 es el valor final)	Tamaño de paso (incremento i en 1)
1	1	VERDADERO	$i=i+1$ (i vale 2)
2	2	VERDADERO	$i=i+1$ (i vale 3)
3	3	VERDADERO	$i=i+1$ (i vale 4)
4	4	VERDADERO	$i=i+1$ (i vale 5)
5	5	VERDADERO	$i=i+1$ (i vale 6)
CORTE	6	FALSO (TERMINA EL CICLO)	-

Hay que recordar que este ciclo corta cuando i tenga un valor 6. En la iteración que i tiene valor 5 al finalizar la ejecución de las acciones dentro del ciclo exacto, i incrementa su valor en 1 y luego compara si su valor es menor o igual a 5, resultando esta comparación falsa por lo tanto corta el ciclo exacto.



Bibliografía

- Lic. Gladys Fernández y Prof. Abel Faure. Programación I. Estructuras algorítmicas repetitivas. Ciclo FOR. UTN FRGP.
- Ejemplos creados por la cátedra y dirección de la carrera.