



<b>MATERIA:</b>	PROGRAMACIÓN I - DIAGRAMAS		
<b>ÁREA TEMÁTICA:</b>	RESOLUCIÓN DE PROBLEMAS – CICLO INEXACTO		
<b>TRABAJO PRACTICO:</b>	IGUALES	<b>NIVEL:</b>	MEDIO
<b>OBJETIVOS:</b>	<ul style="list-style-type: none"><li>▪ COMPRENDER EL PROBLEMA.</li><li>▪ ELABORAR UNA ESTRATEGIA.</li><li>▪ DISEÑAR EL ALGORITMO.</li></ul>		

## PROBLEMA

Se necesita implementar un algoritmo que permita ingresar una lista de números, finalizando cuando se ingresen dos números iguales. Al concluir, el algoritmo debe mostrar el porcentaje de números positivos y negativos ingresados en total, sin contar el último número repetido. Representar el algoritmo mediante diagrama de flujos.

Para resolver este problema, es esencial leer detenidamente el enunciado y entender claramente lo que se busca solucionar.

En este caso, se requiere desarrollar un algoritmo que permita ingresar una lista de números y, una vez finalizado el ingreso, calcular y mostrar el porcentaje de números positivos y negativos ingresados.

A primera vista, el problema puede parecer similar a otros ejercicios resueltos con ciclos exactos (for).

Sin embargo, la complejidad radica en cómo se detiene el ciclo inexacto (while).

El ciclo debe finalizar cuando se ingresan dos números iguales consecutivamente, excluyendo el segundo número igual del cálculo de los porcentajes de positivos y negativos.

Para ilustrarlo mejor, utilizaremos una lista de números como ejemplo.

## Lista de Números

Si el número ingresado es cero, se debe incrementar solamente el contador total de números ingresados.

12

0

-5

10

3

-17

0

6

0

9

9

De esta manera se procederá con el resto de los números de la lista.

Si el número ingresado es positivo, se debe incrementar tanto el contador de números positivos como el contador total de números ingresados.

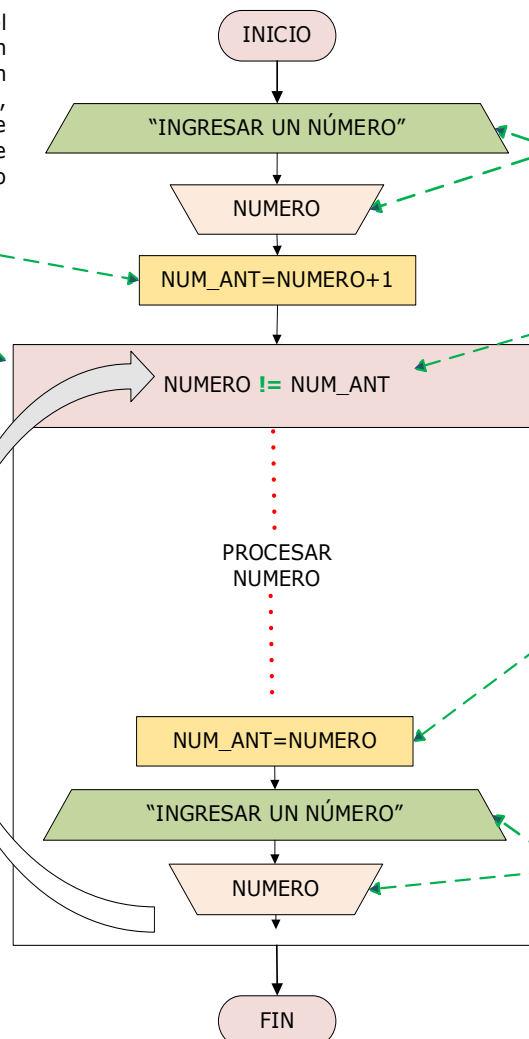
Si el número ingresado es negativo, se debe incrementar tanto el contador de números negativos como el contador total de números ingresados.

Observamos que el último número de la lista (9) es igual al anterior, por lo que el ciclo inexacto (while) debería detenerse y excluir este último número tanto del contador de positivos como del total de números ingresados.

Ahora podemos plantear el ingreso de la lista de números y el corte del ciclo inexacto (while)

Dado que no es posible comparar el primer número ingresado con un número anterior, pero la condición lógica está diseñada para hacerlo, asignamos a NUM\_ANT el valor de NUMERO más 1. Esto garantiza que se ejecute la primera iteración del ciclo correctamente.

CICLO INEXACTO  
(while)



El primer número de la lista se solicita fuera del ciclo inexacto (while). Esto asegura que, al evaluar la condición lógica del ciclo, la variable NUMERO ya tenga un valor asignado. De lo contrario, la condición lógica generaría un error.

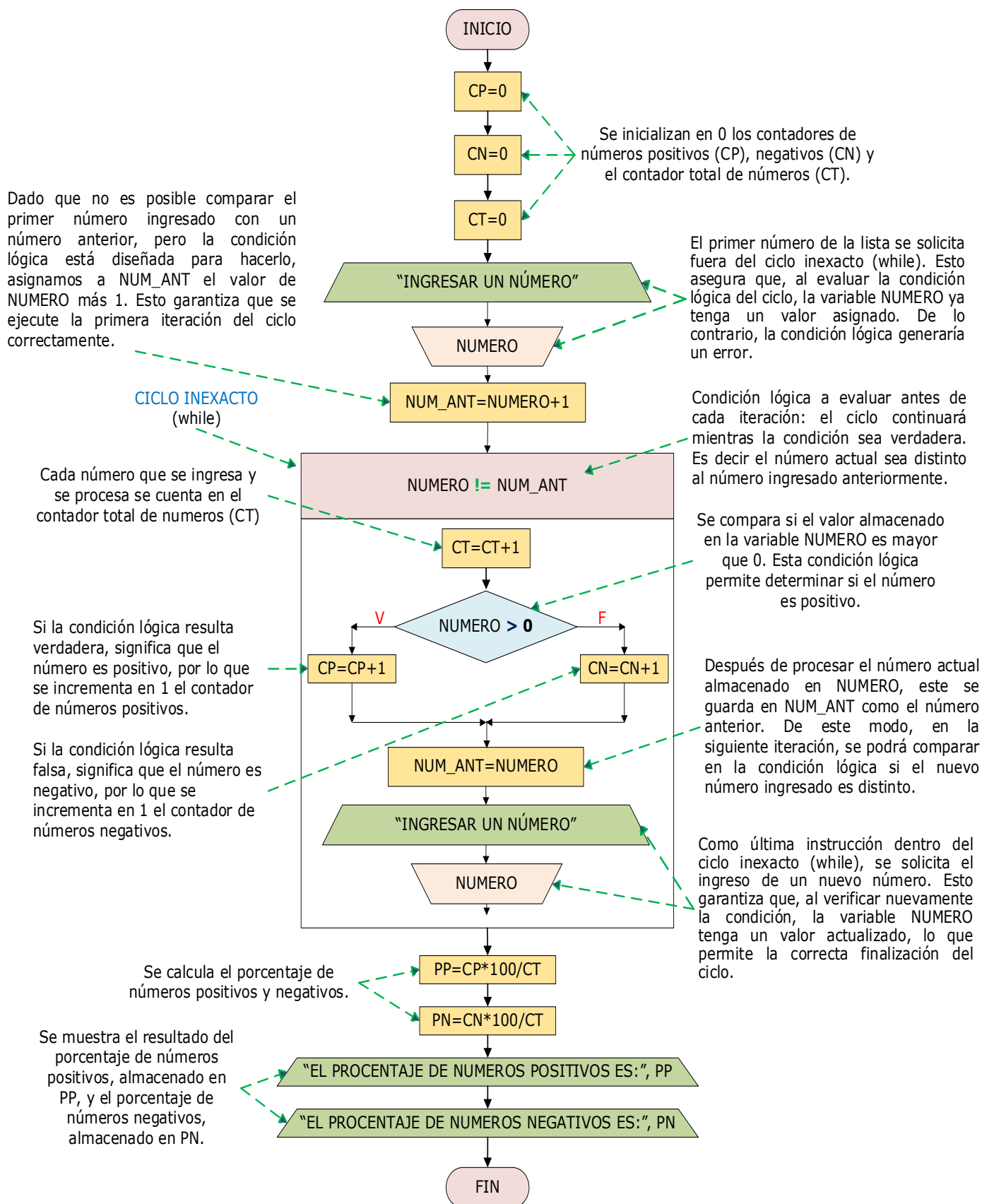
Condición lógica a evaluar antes de cada iteración: el ciclo continuará mientras la condición sea verdadera. Es decir el número actual sea distinto al número ingresado anteriormente.

Se ingresa el nuevo número y vuelve a comprobar la condición lógica.

Después de procesar el número actual almacenado en NUMERO, este se guarda en NUM\_ANT como el número anterior. De este modo, en la siguiente iteración, se podrá comparar en la condición lógica si el nuevo número ingresado es distinto.

Como última instrucción dentro del ciclo inexacto (while), se solicita el ingreso de un nuevo número. Esto garantiza que, al verificar nuevamente la condición, la variable NUMERO tenga un valor actualizado, lo que permite la correcta finalización del ciclo.

Una vez definidos los pasos para ingresar los números y cómo se detiene el ciclo inexacto (while), procedemos a resolver lo solicitado en el enunciado: calcular el porcentaje de números positivos y negativos. Para ello, es necesario conocer la cantidad de números positivos, negativos y el total de números ingresados, recordando que el 0 no se considera ni positivo ni negativo, pero debe contarse como un número más. Para este cálculo, utilizaremos tres contadores, que inicializaremos en 0.



Este problema puede tener varias soluciones, y esta es una de ellas. Si consideramos que el enunciado especifica que el ingreso de números finaliza cuando se ingresan dos números iguales, significa que, como mínimo, se ingresarán dos números. Por lo tanto, se podría utilizar un ciclo inexacto do-while, el cual permite que en la primera iteración no se evalúe la condición de corte. De esta manera, podríamos eliminar la instrucción de la solución anterior ( $NUM\_ANT = NUMERO + 1$ ), que permitía ingresar al ciclo por primera vez.

Con este enfoque, se procesa el primer número, se guarda como número anterior en  $NUM\_ANT$  y luego se solicita un nuevo número. A continuación, se evalúa la condición lógica del ciclo do-while: si el nuevo número es distinto al anterior. Si esta condición resulta falsa, el ciclo se detiene, excluyendo el último número del cálculo de los porcentajes. Esta sería la solución utilizando un ciclo inexacto do-while.

