

INSTITUTO SUPERIOR DE FORMACIÓN TÉCNICA N° 177



Algoritmos y Estructuras de Datos 1

Diagramación Lógica

Prof. Lic. Walter Carnero
profewaltercarnero@gmail.com



APUNTE: AyEDD02 – DIAGRAMACIÓN LÓGICA

INTRODUCCIÓN

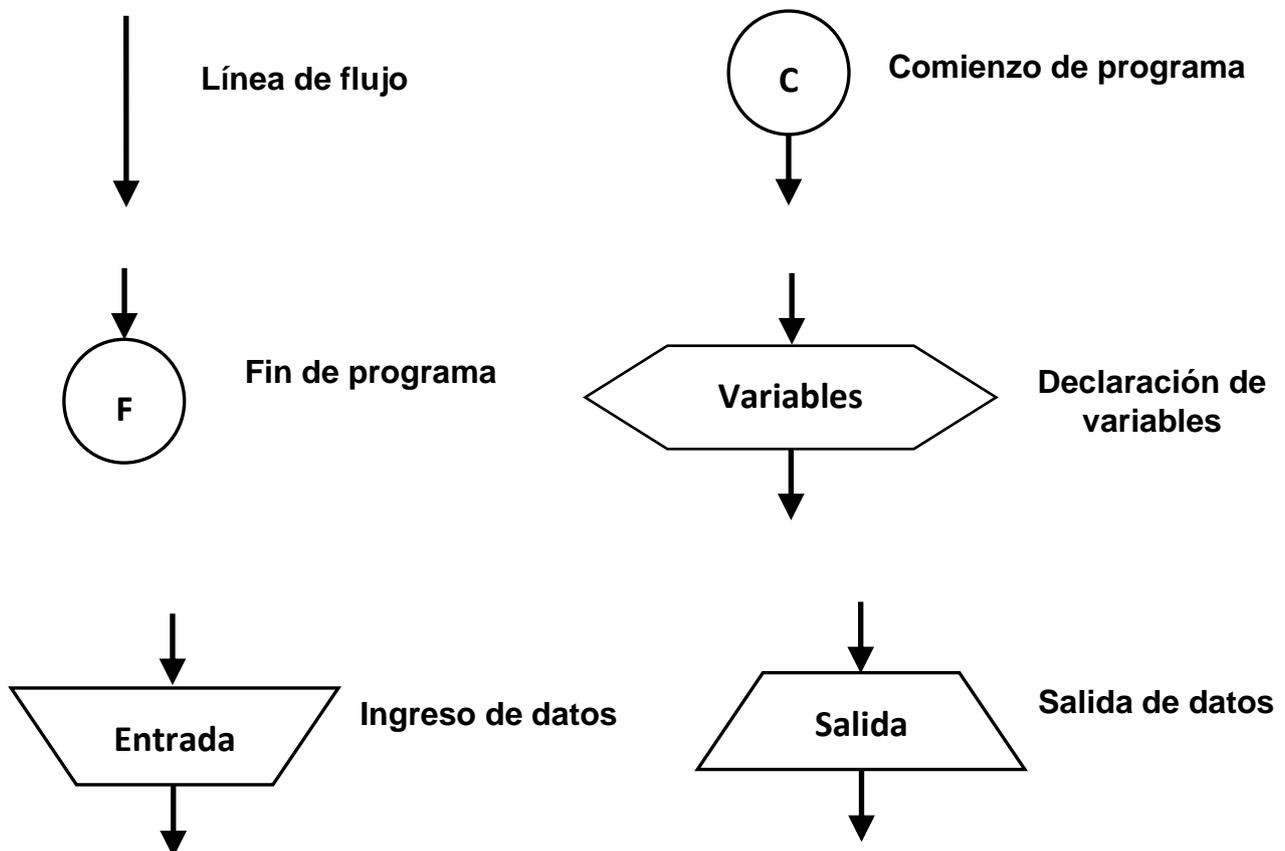
Los programadores cuentan con una variedad de herramientas para solucionar los problemas que luego serán, en muchos casos, sistematizados. Una de esas herramientas es la diagramación lógica.

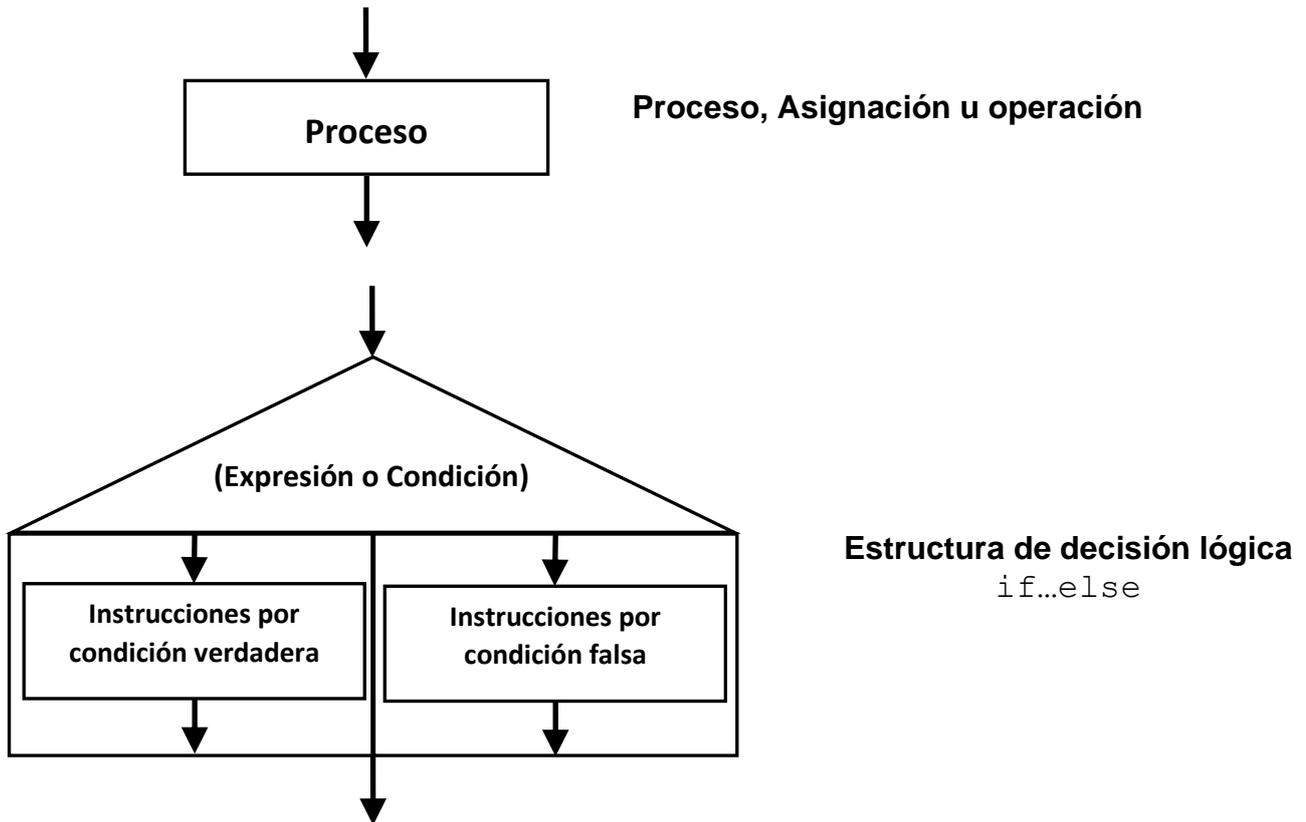
La diagramación lógica facilita el diseño de los algoritmos, mediante el uso de diagramas estandarizados que permiten indicar y visualizar el flujo de datos dentro de la solución que se pretende desarrollar.

Los diagramas son muy útiles a la hora de aprender a programar, pero se vuelven difíciles de aplicar cuando la complejidad y extensión de los programas se incrementa. Debido a ello, solo se suelen realizar diagramas del tipo “Top down”, y no con el detalle que en el presente apunte se muestra.

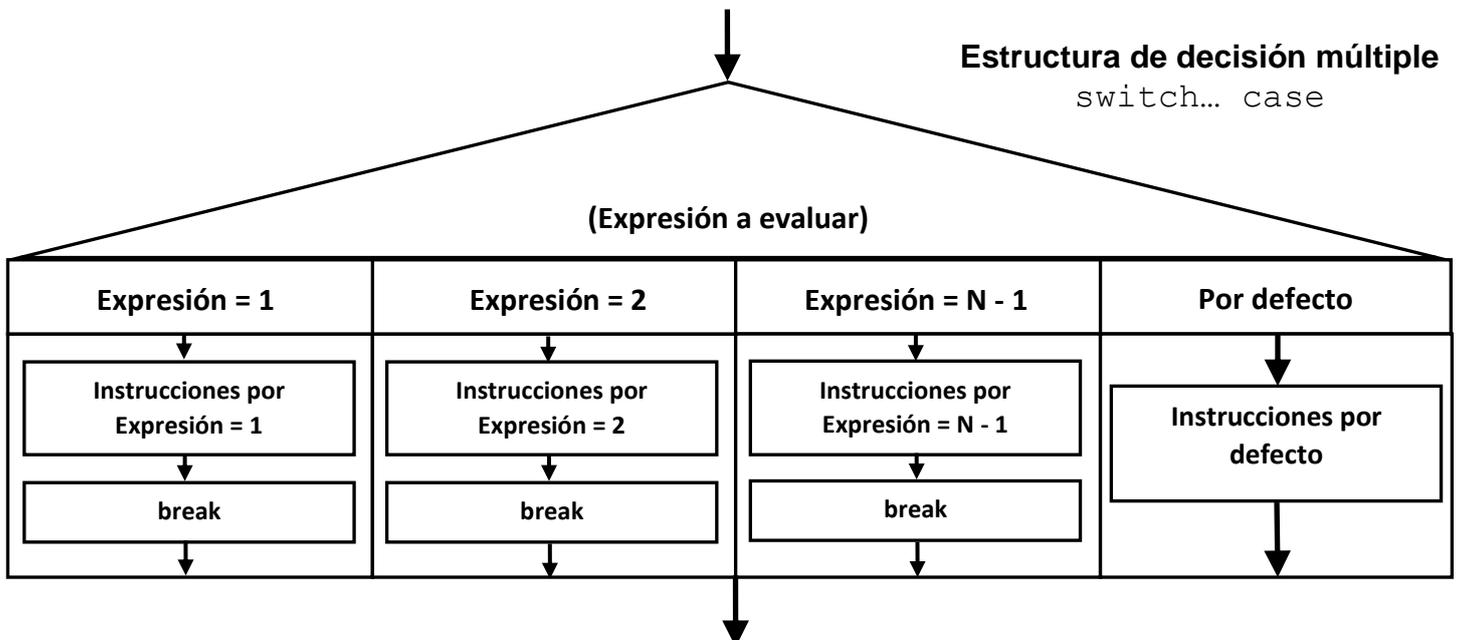
Al finalizar el algoritmo mediante la utilización de los diagramas que aquí se presentan, el mismo puede ser programado mediante cualquier lenguaje, ya que cada símbolo, cada estructura, tiene su correspondencia en la sintaxis del lenguaje elegido para escribir el código fuente.

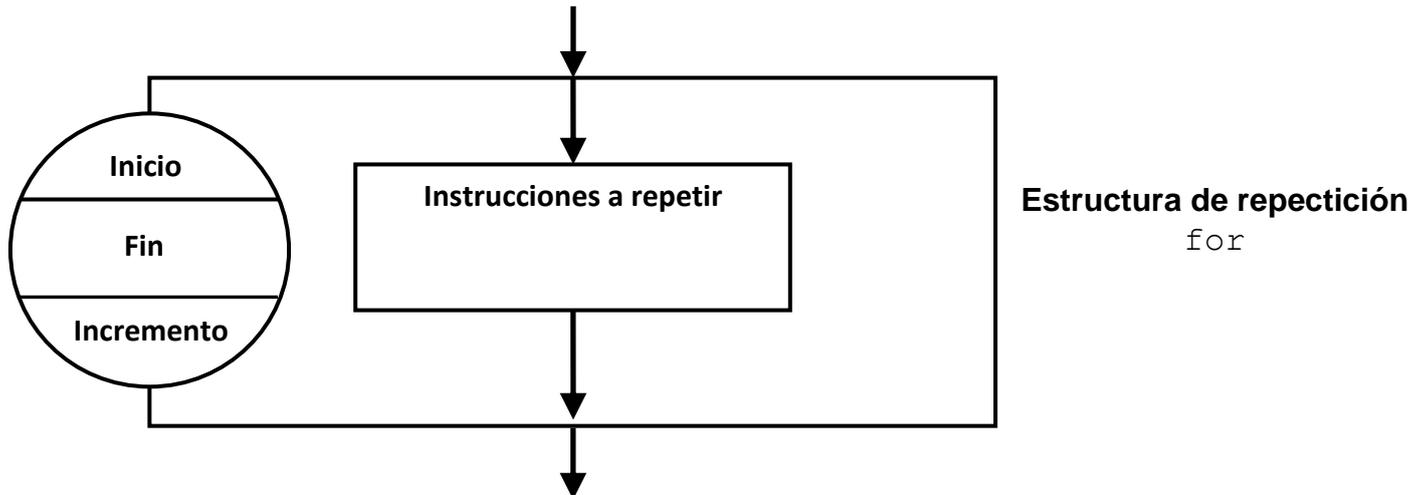
A continuación, se exponen los diagramas que utilizaremos en Algoritmos y Estructuras de Datos, así como lo que dicho diagrama representa.



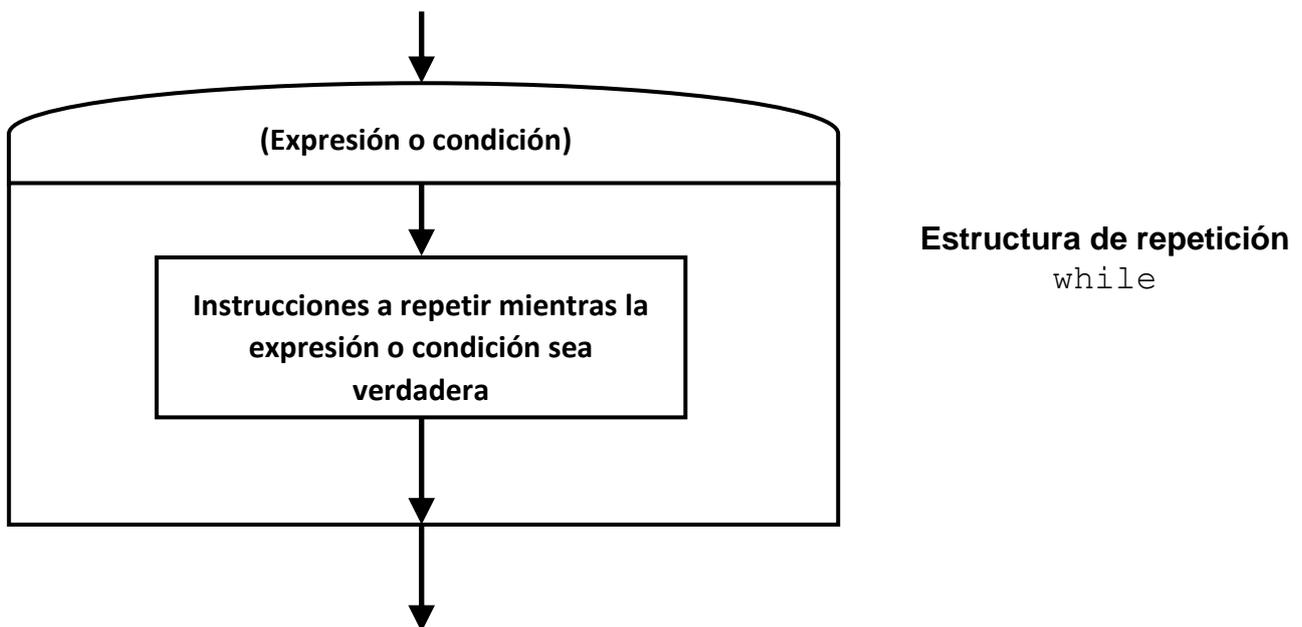


El diagrama que se muestra a continuación es de decisión o selección múltiple, se utiliza cuando se requieren varios `if` anidados, en este caso permite simplificar el algoritmo que requiere varias estructuras de decisión lógica. En programación se denomina `switch`

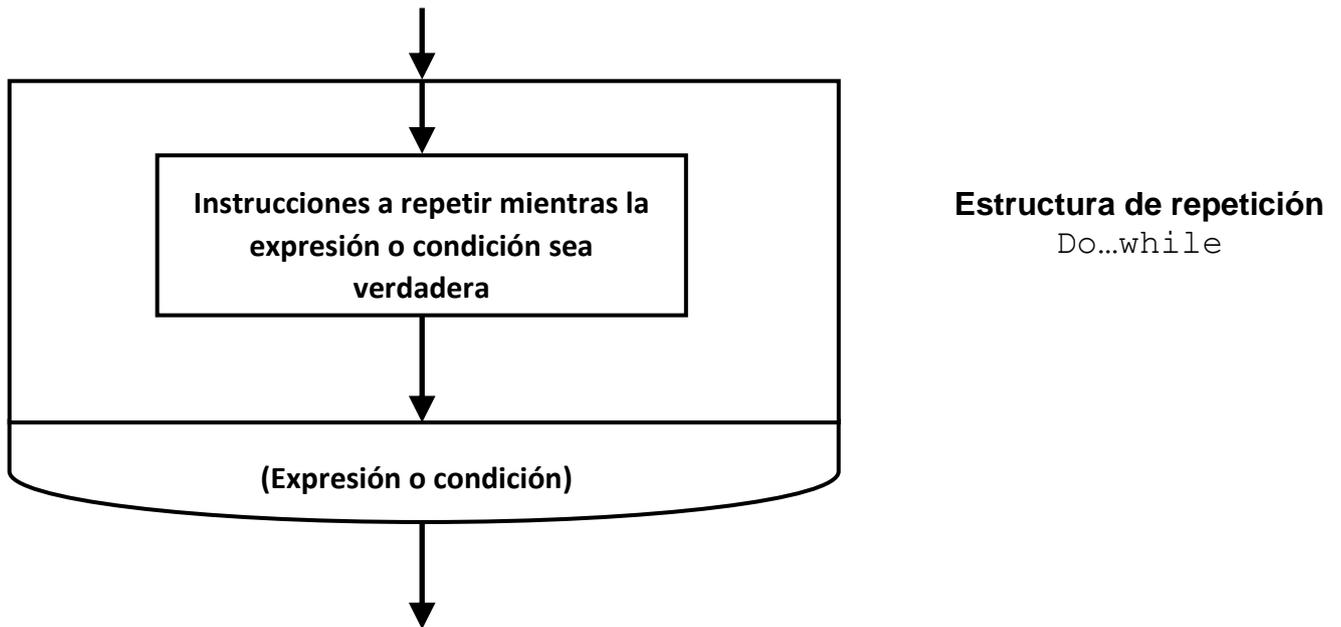




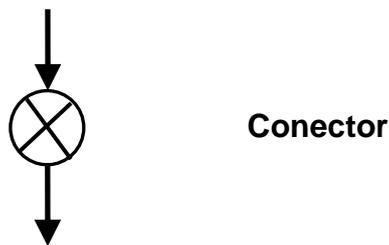
El diagrama sobre este párrafo se denomina `for`, representa una estructura de repetición con iteraciones finitas, es decir, al usarla, conocemos donde inicia y finaliza la repetición. Observar que el círculo que forma parte de la estructura se encuentra dividido en tres, en la parte superior se indica el inicio de la repetición, por ejemplo $X=1$, en la división central se indica la condición de ejecución (fin del ciclo), por ejemplo mientras $X \leq 100$, y por último en la división inferior se indica el incremento que se realiza sobre la variable X en cada pasada por el ciclo, por ejemplo $X=X+1$, indica que el incremento se realizará de uno en uno; con lo cual el ciclo de repetición se ejecutará 100 veces.



Estructura de repetición `while`, la misma se ejecuta mientras la expresión o condición que se encuentra al principio sea verdadera, observar que como dicha condición es evaluada al principio de la estructura si la misma no es verdadera nunca se ejecutará el ciclo. Además dicha expresión deberá tener alguna modificación dentro de la estructura para las iteraciones finalicen en algún momento.



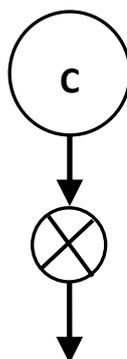
La estructura de repetición `do...while` es similar al `while` con la diferencia de que la expresión o condición se evalúa al final, de esta manera, si dicha condición es falsa desde el principio el código dentro de la estructura se ejecutará por lo menos una vez.

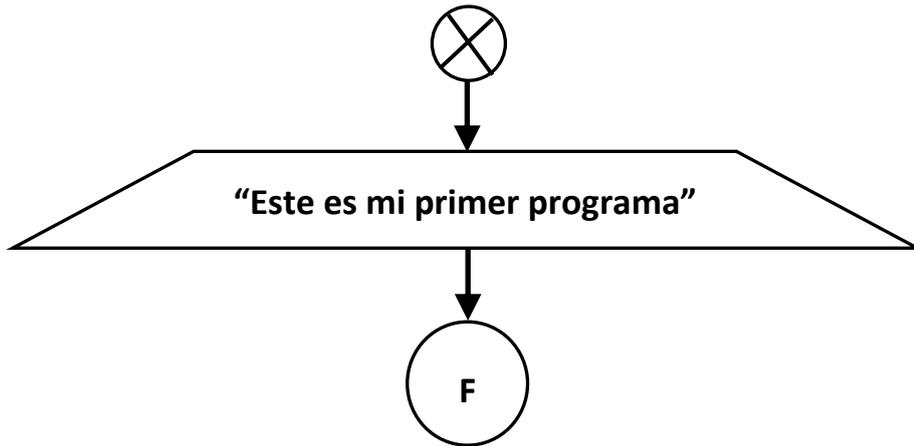


Cuando estamos utilizando diagramación lógica y el algoritmo que desarrollamos no entra en la hoja, para expresar que dicho algoritmo continúa en la hoja siguiente, se utilizan conectores, los cuales permiten establecer la unión de ambos diagramas.

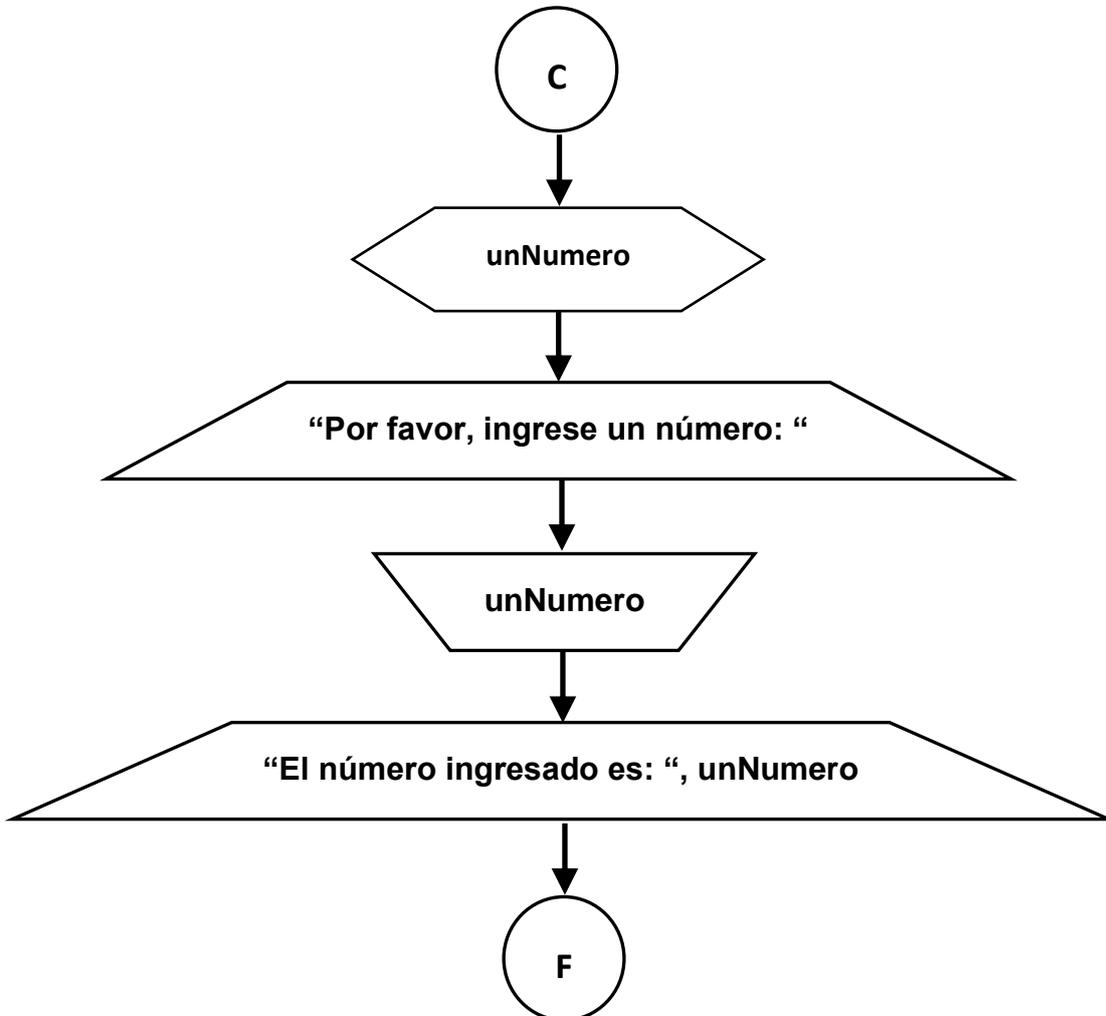
Ejemplos de aplicación para algunos algoritmos

1. Diseñar un algoritmo que permita visualizar en pantalla la leyenda "Este es mi primer programa"

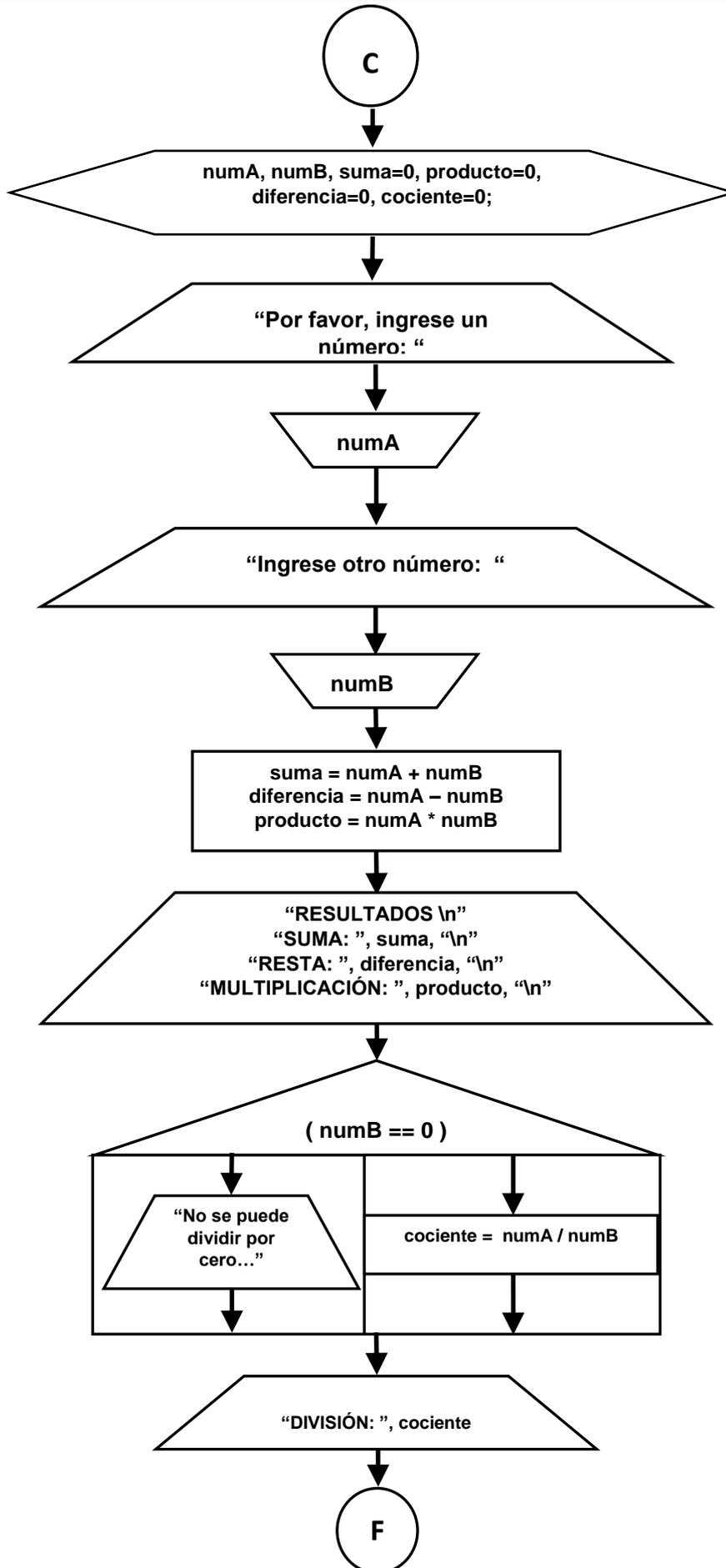




2. Diseñar un algoritmo que permita al usuario ingresar un número con la leyenda "Ingrese un número usando el teclado:" y que al informarlo se pueda visualizar "El número ingresado es:".



3. Diseñar un algoritmo que permita ingresar dos números, para luego informar la suma, diferencia, producto y cociente entre ambos.





4. Diseñar un programa que informe la edad de una persona luego de que la misma haya ingresado el año de nacimiento.

