



Diagramas de flujo

▼ Que es un diagrama de flujo?

Representa la **secuencia lógica o los pasos** que tenemos que dar **para realizar una tarea** mediante unos símbolos y dentro de ellos se describen los pasos a realizar

- Un diagrama de flujo debe proporcionar una **información clara, ordenada y concisa** de todos los pasos a seguir.
- Un diagrama de flujo es una **representación gráfica o simbólica de un proceso.**

▼ Formas

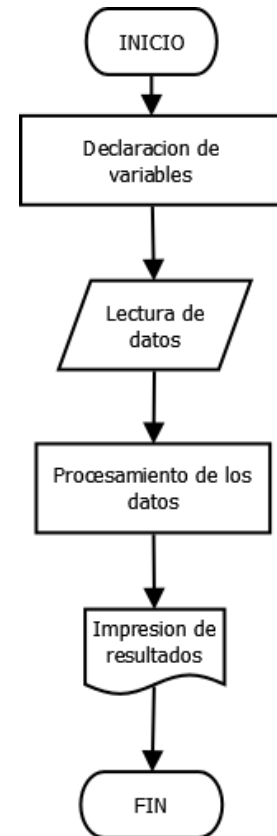
▼ Normas

- Todo diagrama de flujo **debe tener un inicio y un fin** → Elipse
- Las **líneas** utilizadas para indicar la **dirección del flujo del diagrama** deben ser rectas, **verticales u horizontales**, exclusivamente → Flecha
 - Todas las líneas utilizadas para indicar la dirección del flujo del diagrama deben estar conectadas a un símbolo.
- El diagrama debe ser construido **de arriba hacia abajo** (top-down) y de **izquierda a derecha** (left-right)
- La notación utilizada en el **diagrama de flujo** debe ser **independiente** del **lenguaje de programa en el que se va a codificar la solución.**
- Se recomienda **poner comentarios** que expresen o **ayuden a entender** un bloque de símbolos.
- Si la extensión de un diagrama de flujo ocupa más de una página, es necesario utilizar y enumerar los símbolos adecuados.
- A cada símbolo solo le puede llegar una línea de dirección de flujo.

▼ Símbolos

Los diagramas de flujo poseen símbolos que permiten estructurar la solución de un problema de manera gráfica. Por tanto, es fundamental conocer los elementos que conforman este lenguaje gráfico.

- **Elipse** → Representa el **inicio o el fin** del diagrama de flujo.
- **Romboide** → Datos de entrada. Expresa **lectura de datos**.
- **Rectángulo** → Proceso. En su interior se expresan **asignaciones u operaciones**.
- **Rombo** → **Decisión**. Válida una condición y toma uno u otro camino. (**Acción booleana**)
- **Rectángulo de base curva** → Escritura. **Impresión del resultado** o resultados.
- **Flecha** → **Dirección de flujo del diagrama**.
- **Círculo** → **Conexión dentro de la misma página**.
- **Cuadrado en punta** → **Conexión entre diferentes páginas**.
- **Hexágono** → **Decisión múltiple**. Almacena un selector que determina la rama por la que sigue el flujo.



El esquema para construir un diagrama de flujo

▼ Estructuras de control de flujo

Las estructuras de control de flujo permiten la ejecución **condicional** y la **repetición** de un conjunto de instrucciones.

Existen 3 estructuras de control: secuencial, condicional y repetitivas o iterativas.

1. Estructura de control secuencial

Las estructuras de control secuenciales son las **sentencias o declaraciones** que **se realizan una a una**.

continuacion de otra en el orden en el que estan escritas.

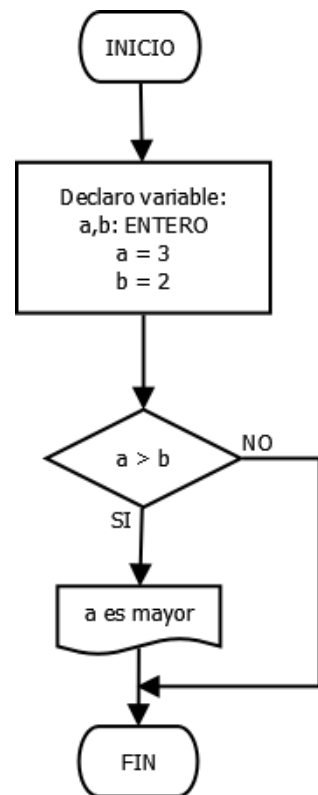
Declaro variable:
x: [tipo de variable]
x= [valor]

2. Estructuras de control condicionales (o selectivas)

Las estructuras de control condicionales permiten evaluar una expresion logica (condicion que puede ser verdadera o falsa) y, dependiendo del resultado, se realiza uno u otro de instrucciones. Estas estructuras son mutuamente excluyentes (o se ejecuta una accion o se ejecuta la otra).

▼ La estructura de control de flujo mas simple es la estructura condicional SI (IF)

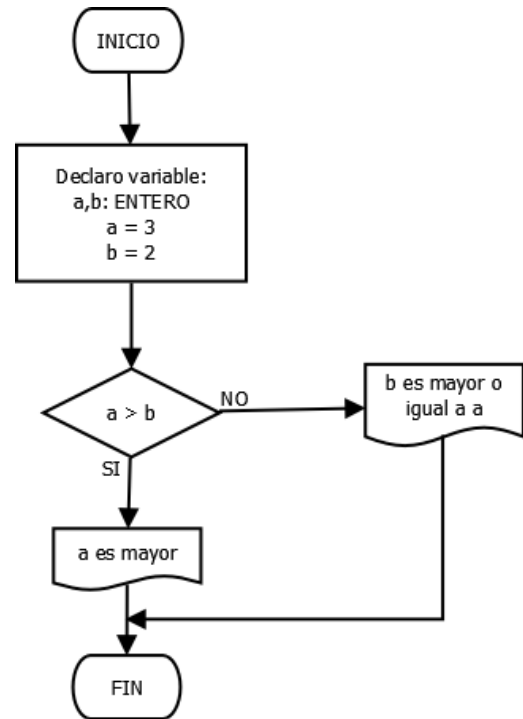
Se evalua la expresion logica y si se cumple (si la condicion es verdadera) se ejecutan las instrucciones del bloque [Acciones]. Si no se cumple la condicion, se continua con el flujo normal del programa.



▼ La estructura condicional completa es SI-DE LO CONTRARIO (IF-ELSE)

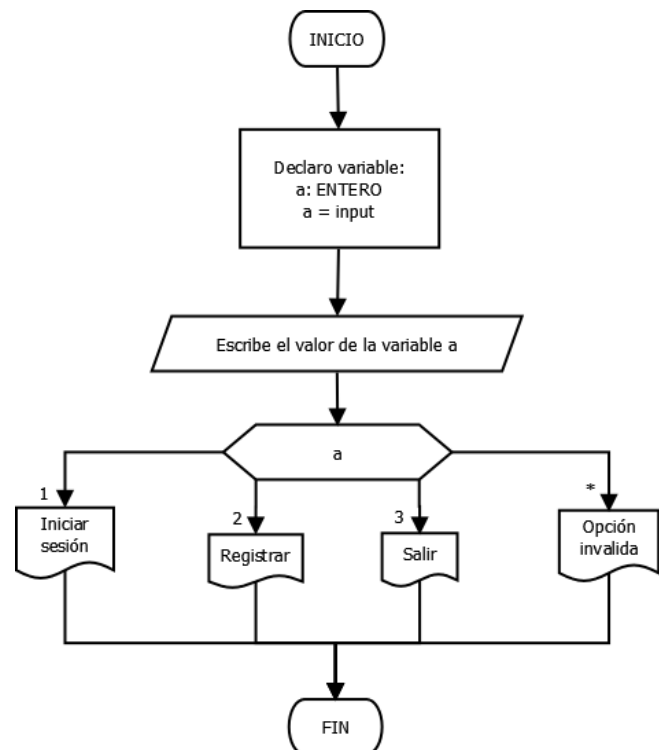
Se valua la expresion logica y si se cumple (si la condicion es verdadera) se ejecutan las

instrucciones del bloque Si, Si no se cumple la condicion se ejecutan las instrucciones del bloque No. Al final el programa sigue su flujo normal.



▼ La estructura condicional SELECCIONAR-CASO (CASE)

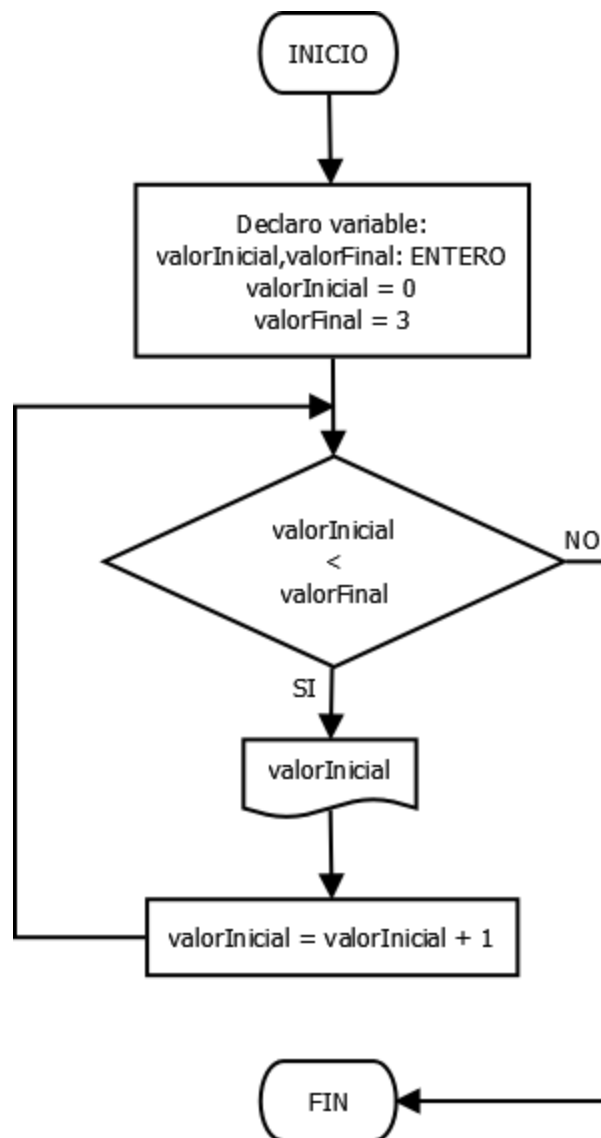
Valida el valor de la variable que esta en el hexagono y comprueba se es igual al valor que esta definido en cada caso (lineas que emanan del hexagono). Si la variable no tiene el valor de algun caso se va la instruccion por defecto (*).



3. Estructuras de control iterativas o repetitivas

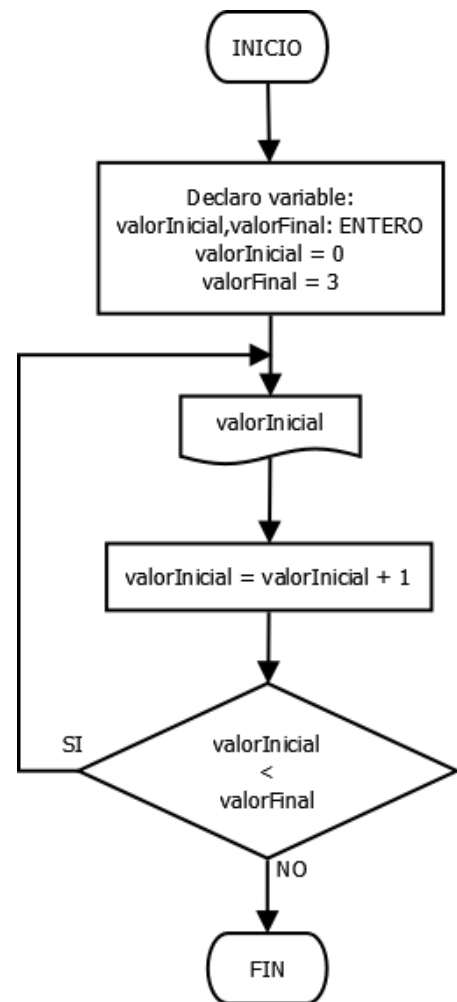
Las estructuras de control de flujo iterativas o repetitivas (también llamadas cíclicas) permiten ejecutar una serie de instrucciones mientras se cumpla la expresión lógica. Existen dos tipos de expresiones cíclicas MIENTRAS y HACER-MIENTRAS.

- ▼ La estructura MIENTRAS primero valida la condición y si esta es verdadera procede a ejecutar el bloque de instrucciones de la estructura, de lo contrario rompe el ciclo y continúa el flujo normal del programa.



▼ La estructura **HACER-MIENTRAS** primero ejecuta las instrucciones descritas en la estructura y al final valida la expresion logica.

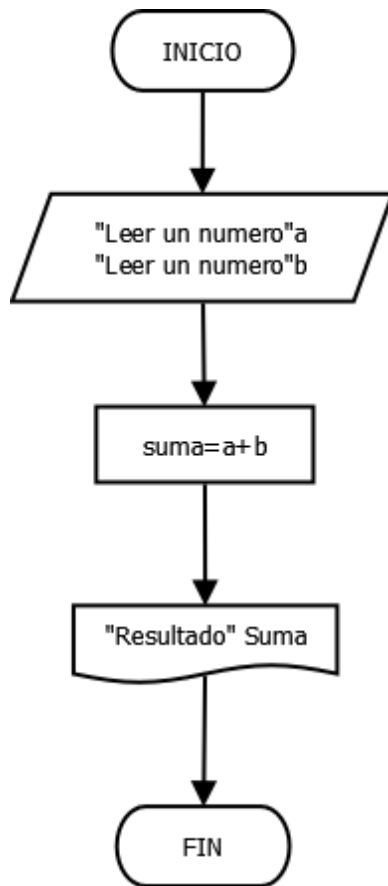
Si la condicion se cumple vuelve a ejecutar las instrucciones de la estructura, de lo contrario rompe el ciclo y sigue el flujo del algoritmo. Esta estructura asegura que, por lo menos, se ejecuta una vez el bloque de la estructura, ya que primero ejecuta y despues pregunta por la condicion.



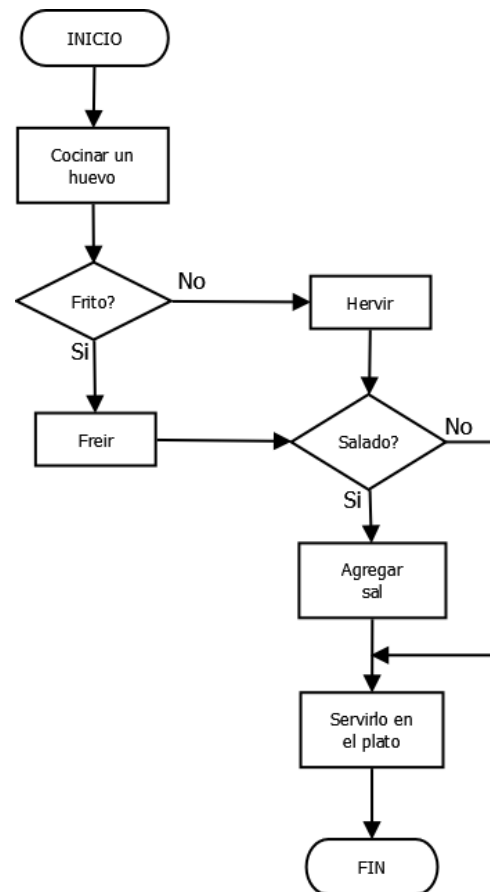
Programa que suma dos numeros y nos da el resultado en pantalla

▼ Ejemplos

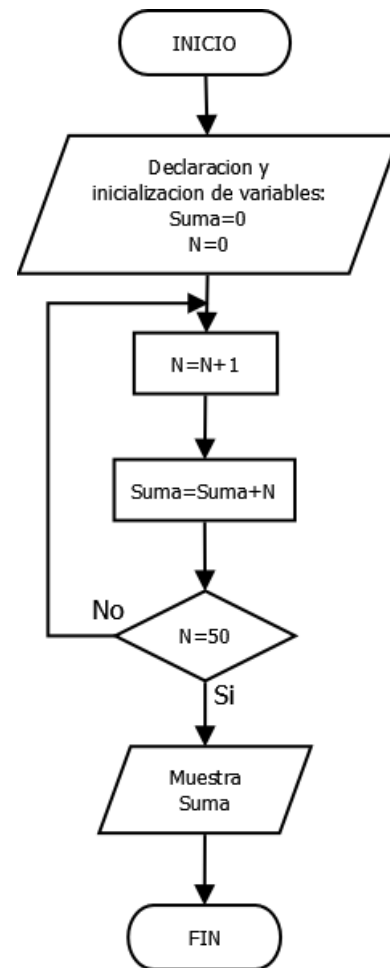
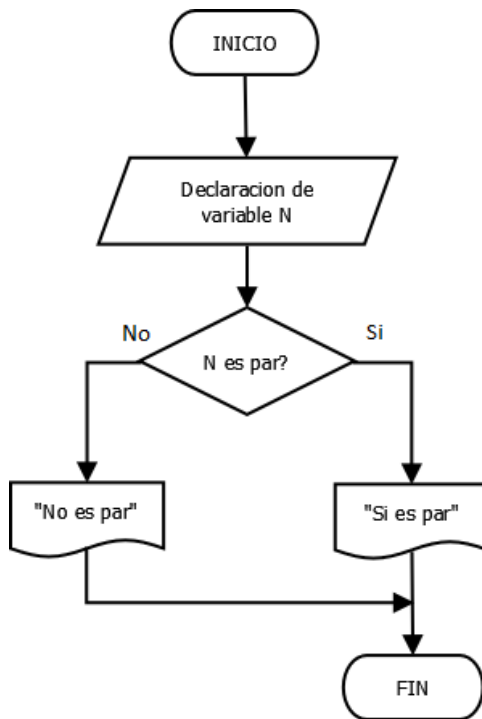
1. Programa informático que nos sume dos número y nos de el resultado en pantalla:
3. Además los diagramas de flujo no solo valen para informática, incluso podemos hacer uno para cocinar un huevo:



2. Programa que nos diga si un número es par o impar:



4. Un diagrama de flujo para mostrar la suma de los 50 primeros números:



Lo primero es poner a cero la suma y dar el primer número a sumar que será el 0. El diagrama acaba momento, es 50. Mientras no sea 50 el programa vuelve a la tercera secuencia que será sumarle un número al anterior $N=N+1$

Buscar un diagrama de flujo de datos (típico de hoja de producto o servicio contratado) y comentarlo