

**UTNIFRGP**

UNIVERSIDAD TECNOLÓGICA NACIONAL  
FACULTAD REGIONAL GENERAL PACHECO

Apuntes de clase de la asignatura

# **INTRODUCCIÓN A LA PROGRAMACIÓN**

**DIAGRAMA DE FLUJOS**

CURSO DE NIVELACIÓN – TECNICATURA UNIVERSITARIA EN  
PROGRAMACIÓN

**2024**







## DIAGRAMA DE FLUJOS

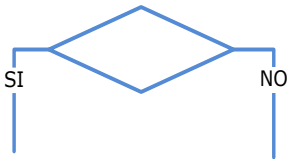
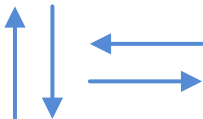
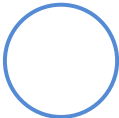

Un diagrama de flujo representa la esquematización gráfica de un algoritmo.

En realidad, muestra gráficamente los pasos o procesos a seguir para alcanzar la solución de un problema. Su correcta construcción es sumamente importante porque a partir del mismo se escribe un programa en algún lenguaje de programación.

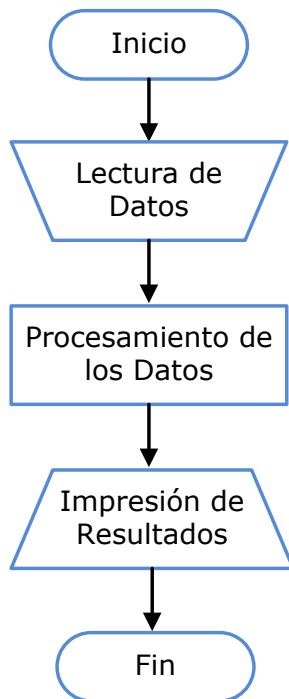
Si el diagrama de flujo está completo y correcto, el paso de este a un lenguaje de programación es relativamente simple y directo. A continuación, presentamos los símbolos que utilizaremos, y una explicación de estos. Estos satisfacen las recomendaciones de la "International Organization for Standardization" (ISO) y la "American National Standards Institute" (ANSI).

SÍMBOLOS UTILIZADOS EN LOS DIAGRAMAS DE FLUJOS		
SÍMBOLO	FUNCION	DESCRIPCION
	Inicio / Fin	<i>Símbolo utilizado para marcar el <b>Inicio</b> y el <b>Fin</b> del diagrama de flujo.</i>
	Entrada de Datos	<i>Símbolo utilizado para Introducir los <b>Datos de Entrada</b>. Expresa <b>Lectura</b>.</i>
	Imprimir Resultados	<i>Símbolo utilizado para representar la <b>Impresión</b> de un resultado. Expresa <b>Escritura</b>. (<b>Salida</b>)</i>
	Proceso	<i>Símbolo utilizado para representar un <b>Proceso</b>. En su interior se expresan asignaciones, operaciones aritméticas, cambios de valor de celdas en memoria, etc.</i>



	Decisión	<i>Símbolo utilizado para representar una <b>Decisión</b>. En su interior se almacena una condición, y dependiendo del resultado de la evaluación de la misma se sigue por una de las ramas o caminos alternativos. Este símbolo se utiliza en la <b>Estructura Selectiva Si / Entonces</b>.</i>
	Flujo de Datos	<i>Símbolos utilizados para expresar la <b>Dirección del Flujo del Diagrama</b>.</i>
	Conector	<i>Símbolo utilizado para expresar <b>Conexión</b> dentro de una <b>Misma Página</b>.</i>
	Conector	<i>Símbolo utilizado para expresar <b>Conexión</b> entre <b>Páginas Diferentes</b>.</i>

Estas son algunas de las etapas que debemos seguir en la construcción de un diagrama de flujo.



**Nota:**

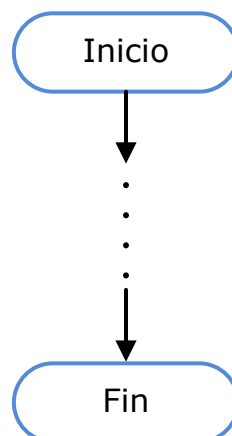
*Estas son algunas de las etapas que se presentan en la mayoría de los Diagramas de Flujo. Tener en cuenta que a veces se pueden presentar en diferente orden y repitiendo algunas de ellas. También es frecuente tomar decisiones o repetir una serie de pasos un número de veces determinado.*

## Reglas para la construcción de diagramas de flujo

Debemos recordar que un diagrama de flujo debe ilustrar gráficamente los pasos o procesos a seguir para alcanzar la solución de un problema. Los símbolos presentados, colocados adecuadamente, permiten crear una estructura gráfica flexible que ilustra los pasos a seguir para alcanzar un resultado específico.

El diagrama de flujo facilitará más tarde la escritura del programa en algún lenguaje de programación.

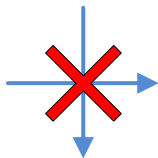
1. Todo diagrama de flujo debe tener un Inicio y Fin.



2. Las líneas utilizadas para indicar la dirección del flujo del diagrama deben ser rectas, verticales y horizontales.



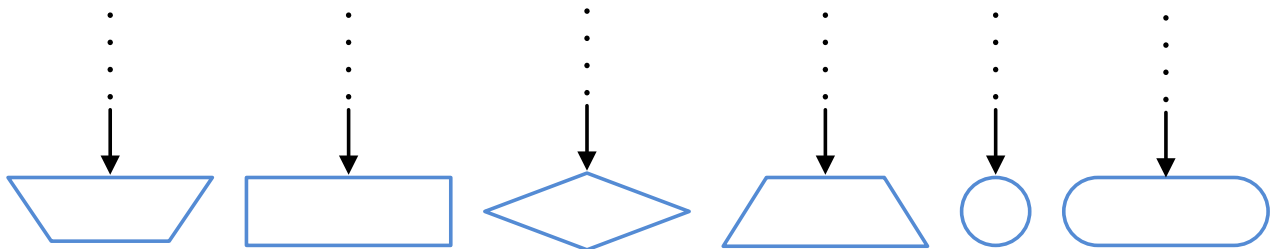
No deben ser inclinadas.



Tampoco debemos cruzarlas.

**3.** No deben quedar líneas de flujo sin conectar.

**4.** Todas las líneas utilizadas para indicar la dirección del flujo del diagrama deben estar conectadas. La conexión puede ser a un símbolo que exprese lectura, proceso, decisión, Impresión, conexión o fin de diagrama.



**5.** El diagrama de flujo debe ser construido de arriba hacia abajo (top-down) y de izquierda a derecha (right to left).

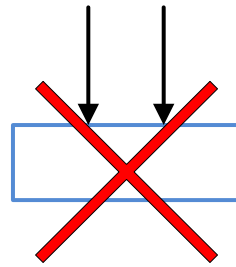
**6.** La notación utilizada en el diagrama de flujo debe ser independiente del lenguaje de programación. La solución presentada en el diagrama puede escribirse posteriormente y fácilmente en diferentes lenguajes de programación.

**7.** Todo texto escrito dentro de un símbolo deberá ser escrito claramente, evitando el uso de muchas palabras.

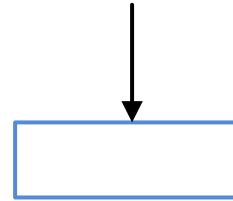
**8.** Es conveniente cuando realizamos una tarea compleja poner comentarios que expresen o ayuden a entender lo que hicimos.

**9.** Si el diagrama de flujo requiriera más de una hoja para su construcción, debemos utilizar los conectores adecuados y enumerar las páginas convenientemente.

**10.** No puede llegar más de una línea a un símbolo.



No Valido



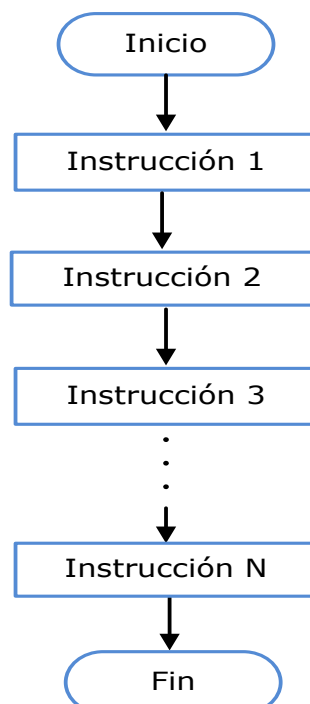
Valido

## ESTRUCTURAS DE CONTROL (FLUJO DE CONTROL)

Sin importar qué herramienta o técnica se utilice para la solución de un problema dado, ésta tendrá una estructura, que se refiere a la secuencia en que se realizan las operaciones o acciones para resolver el problema; esas estructuras pueden ser: secuenciales, de decisión y de ciclo o repetición.

### ESTRUCTURA SECUENCIAL

La estructura secuencial es aquella en la que una instrucción sigue a otra en secuencia. Las instrucciones se ejecutan de arriba hacia abajo, de tal modo que la salida de una es la entrada de la siguiente, en el orden en que se las haya escrito y así sucesivamente hasta el fin del proceso. Se puede decir que el flujo normal de un algoritmo es secuencial.



**EJEMPLO1**



Construya un Algoritmo que solicite el Ingreso de tres números Enteros, los guarde en las variables A, B y C, imprima (Escribir en Pantalla) los mismos en orden inverso.

### **DEFINICIÓN DEL PROBLEMA** (¿Qué debo hacer?)

*Mostrar tres números enteros de manera inversa a la que se ingresaron.*

### **ANÁLISIS DEL PROBLEMA** (¿Qué datos, Que Proceso....?)

**Datos:** *Tres números enteros.*

**Resultados:** *Imprimir los tres números de manera inversa.*

**Proceso:** *No debo realizar ninguno.*

### **DISEÑO DE UNA ESTRATEGIA** (¿Como lo voy a hacer?)

- 1. Imprimir un cartel que solicite el Ingreso de los tres Números.*
- 2. Leer los tres Números y Almacenarlos en sus respectivas Variables.*
- 3. Imprimir los tres Números de manera inversa.*

### **CONSIDERACIONES**

*Para el inicio y fin del diagrama de flujo se utiliza el símbolo:*



*Para lectura se utiliza el símbolo:*



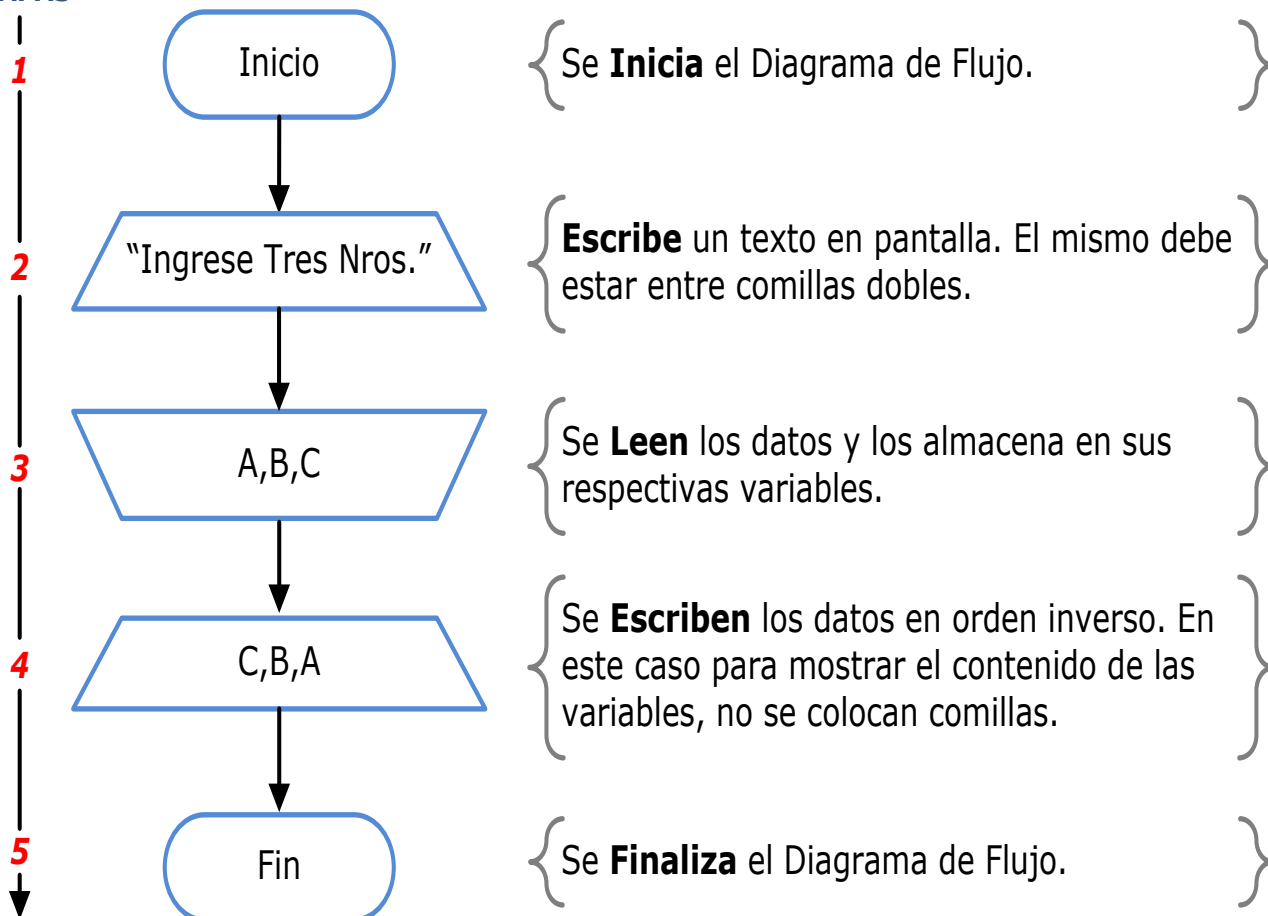
*Para Imprimir (escritura) se utiliza el símbolo:*



### **ALGORITMO** (Diagrama de Flujo)



ETAPAS



Suponiendo que los **Datos** (Números) que se ingresan son: **5, 17, 53**, la **Impresión** produce lo siguiente: **53, 17, 5**.

Todos los ingresos de datos deben ir acompañados previamente por impresiones (Escritura) que guíe al usuario que datos debe ingresar. Así mismo durante el curso prescindiremos de ellos con motivo de hacer más rápido los Diagramas de Flujos y ocupar menos espacio.

## PRUEBA DE ALGORITMOS

Una vez construido el algoritmo que resuelve el problema, es importante hacer algunas pruebas para comprobar su comportamiento y, si es el caso, realizar las respectivas correcciones.

Cada uno puede ingeniar su propio método de comprobación de acuerdo con sus necesidades y de los elementos que disponga para tal fin.





Uno de los métodos más utilizados para verificar un algoritmo, son las *pruebas de escritorio*, que consisten en utilizar nuestro cerebro como si fuera el procesador que va a ejecutar el algoritmo. Se asignan diferentes valores a las variables de entrada y se realiza un recorrido por los diferentes pasos del algoritmo, ejecutando todas las instrucciones y tomando nota de los valores que van adquiriendo las variables, hasta llegar al final.

Después del recorrido se verifica que las salidas sean correctas y de ser necesario se repite el proceso con otros datos. Para esto, por lo general, se utiliza una tabla en la cual se ubican las variables del algoritmo y los valores que toman en las diferentes instrucciones.

INSTRUCCIÓN	VARIABLES				
	Entrada1	Entrada2	....	Salida1	....
Instrucción 1					
Instrucción 2					
<b><i>Repetir con otros datos</i></b>					
Instrucción 1					
Instrucción 2					

## EJEMPLO 2

Construya un Algoritmo que solicite el Ingreso de dos números Enteros y calcule e imprima (Escritura en Pantalla) la suma de los mismos.

### **DEFINICIÓN DEL PROBLEMA** (¿Qué debo hacer?)

*Mostrar el resultado de la suma de dos números.*

### **ANÁLISIS DEL PROBLEMA** (¿Qué datos, Que Proceso....?)

**Datos:** *Dos números enteros.*

**Resultados:** *Imprimir el Resultado de la Suma.*

**Proceso:** *La Suma de dos números enteros.*

### **DISEÑO DE UNA ESTRATEGIA** (¿Como lo voy a hacer?)

**1.** *Leer los dos Números y Almacenarlos en sus respectivas Variables.*



2. Hacer la Suma de los dos Números Enteros.
3. Imprimir el resultado de la Suma.

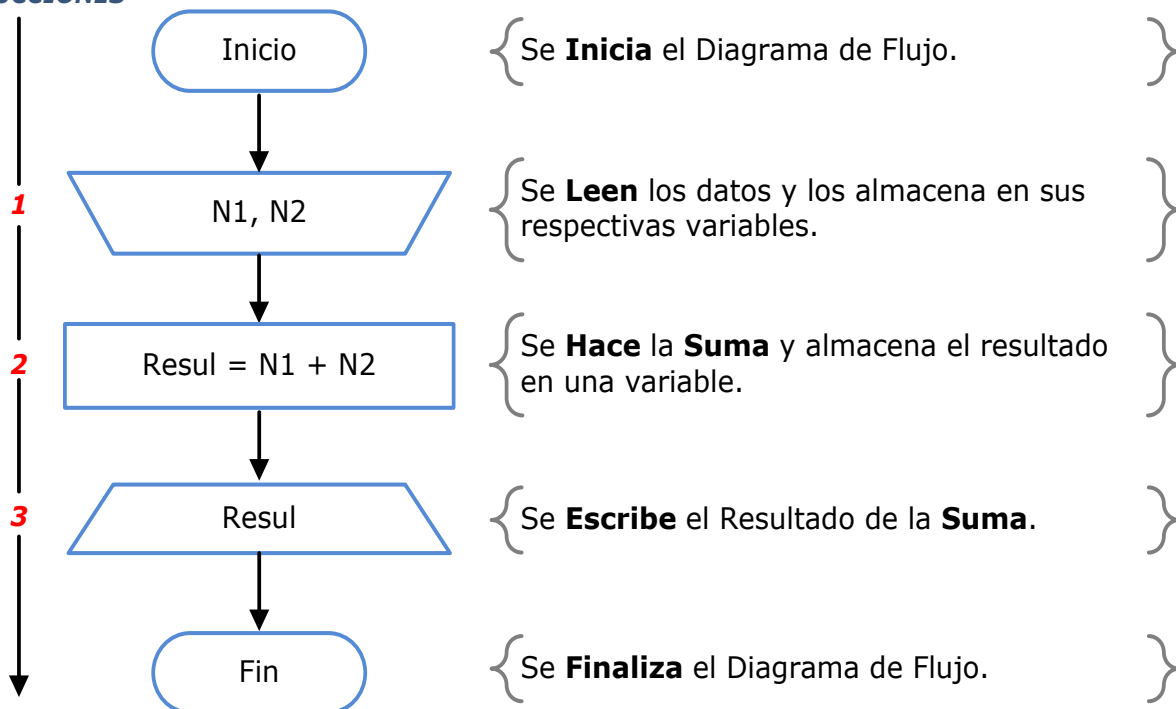
## CONSIDERACIONES

Para realizar la Suma se utilizará el símbolo de proceso, de esta manera asignaremos el resultado de una expresión aritmética (La Suma) a una variable que nos guarde el resultado. Esto se conoce como asignación.



## ALGORITMO (Diagrama de Flujo)

### INSTRUCCIONES



## PRUEBA DEL ALGORITMO

En la primera prueba suponemos que se ingresaron los Datos (Números) 5 y 3, en sus respectivas variables N1 Y N2. El Resultado seria 8.

En la segunda prueba suponemos que se ingresaron los Datos (Números) 15 y 12, en sus respectivas variables N1 Y N2. El Resultado seria 27.



INSTRUCCIÓN	VARIABLES		
	N1	N2	Resul
<i>Instrucción 1</i>	<b>5</b>	<b>3</b>	
<i>Instrucción 2</i>	<b>5</b>	<b>3</b>	<b>8</b>
<i>Instrucción 3</i>	<b>5</b>	<b>3</b>	<b>8</b>
<b><i>Repetir con otros datos</i></b>			
<i>Instrucción 1</i>	<b>15</b>	<b>12</b>	
<i>Instrucción 2</i>	<b>15</b>	<b>12</b>	<b>27</b>
<i>Instrucción 3</i>	<b>15</b>	<b>12</b>	<b>27</b>



## BIBLIOGRAFIA

---

- ❖ Osvaldo Cairó Battistutti (2005). Metodología de la Programación. Editorial Alfaomega.
- ❖ Pedro Vicente Rosero Montaña (2006). Introducción a la Programación de Computadores.
- ❖ Luis Joyanes Aguilar (2008). Fundamentos de Programación. Cuarta edición. Editorial McGRAW-HILL.
- ❖ Francisco Javier Pinales Delgado, César Eduardo Velázquez Amador. Problemario de algoritmos resueltos con diagrama de flujos y pseudocódigo. Universidad Autónoma de Aguascalientes.