



***Técnico en***  
**< DESARROLLO DE SOFTWARE >**

***Lógica de Programación I***

(CC BY-NC-ND 4.0)  
International

Attribution-NonCommercial-NoDerivatives 4.0



## **Atribución**

Usted debe reconocer el crédito de una obra de manera adecuada, proporcionar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que tiene el apoyo del licenciante o lo recibe por el uso que hace.



## **No Comercial**

Usted no puede hacer uso del material con fines comerciales.



## **Sin obra derivada**

Si usted mezcla, transforma o crea un nuevo material a partir de esta obra, no puede distribuir el material modificado.

No hay restricciones adicionales - Usted no puede aplicar términos legales ni medidas tecnológicas que restrinjan legalmente a otros hacer cualquier uso permitido por la licencia.

<http://creativecommons.org/licenses/by-nc-nd/4.0/>



# *Lógica de Programación I*

## *Unidad III*

### 1. Algoritmos Simbólicos


Este tipo de algoritmos proporcionan una visión general de la solución del problema. Se representan gráficamente con un conjunto de símbolos y reglas para que sea fácil su comprensión.







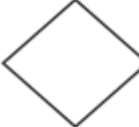

#### Diagramas de flujo



Los diagramas de flujo son representaciones gráficas de un algoritmo o de un fragmento del mismo. Están formados por una secuencia de símbolos, cada símbolo representa una acción distinta. Estos símbolos están conectados entre sí por flechas denominadas líneas de flujo.

#### *Símbolos del diagrama de flujo*

Los símbolos han sido normalizados por el Instituto Norteamericano de Normalización (ANSI, por sus siglas en inglés).

Símbolo	Nombre	Descripción
	Terminador	Indica el inicio o salida del algoritmo

	Proceso	Son procesos o acciones que el programa tiene que realizar.
	Subrutina	Es un marcador que representa un segmento de código que está formalmente declarado en otro lugar. Por ejemplo procedimientos o funciones.
	Entrada/Salida de datos	Indica la entrada o salida de datos
	Desplegar	Indica la salida de datos en pantalla.
	Salida en documento	Indica la salida de datos o información en un documento. Por ejemplo la salida de información por impresora.
	Solicitar dato de entrada	Indica la solicitud al usuario de información que debe ser ingresada de forma manual.
	Decisión	Indica operaciones lógicas o de comparación.
	Preparación	Indica un ciclo de preparación.

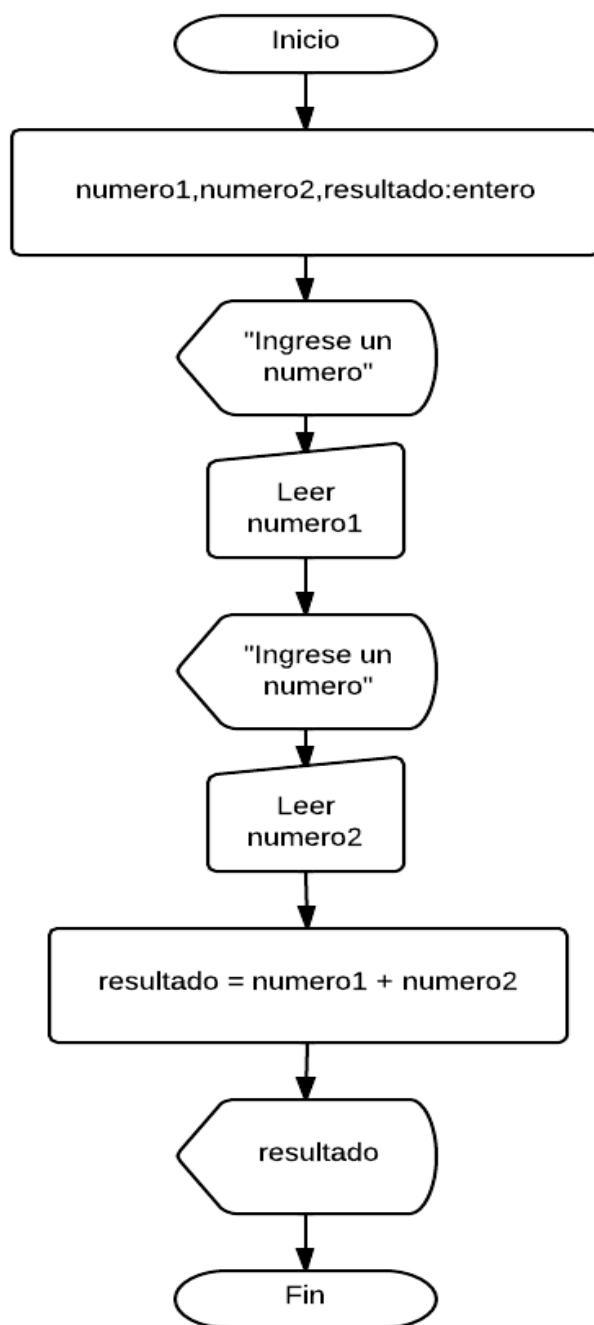
	Conector	Enlaza dos partes de un algoritmo en la misma página.
	Conector fuera de página	Enlaza dos partes de algoritmo representadas en páginas diferentes.

A continuación mostraremos algunos ejemplos de cómo se realizan los diagramas de flujo:

## *Estructura secuencial*

Ejemplo:

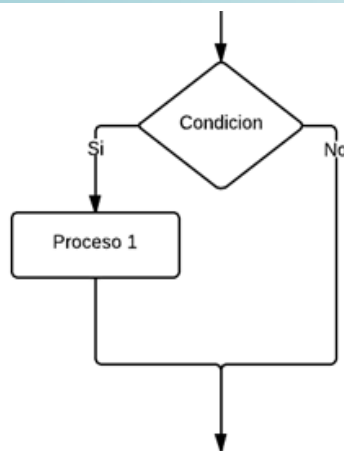
Diseñar un algoritmo que despliegue el resultado de la suma de dos números.



## Estructura condicional

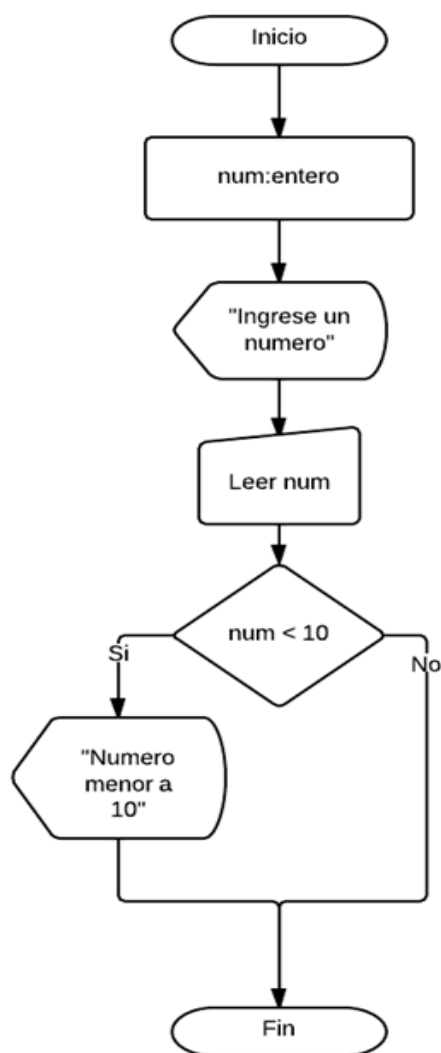
Estructura condicional simple

Representación



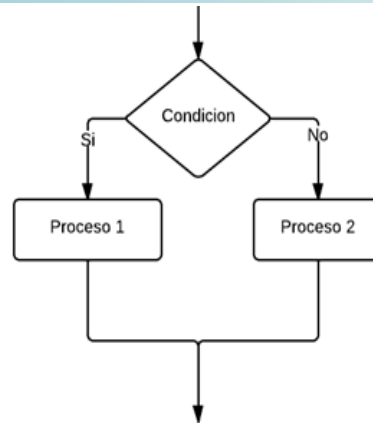
Ejemplo:

Diseñar un algoritmo que solicite el ingreso de un número, si el número es menor a diez que despliegue el mensaje “Número menor a 10”.



Estructura condicional doble

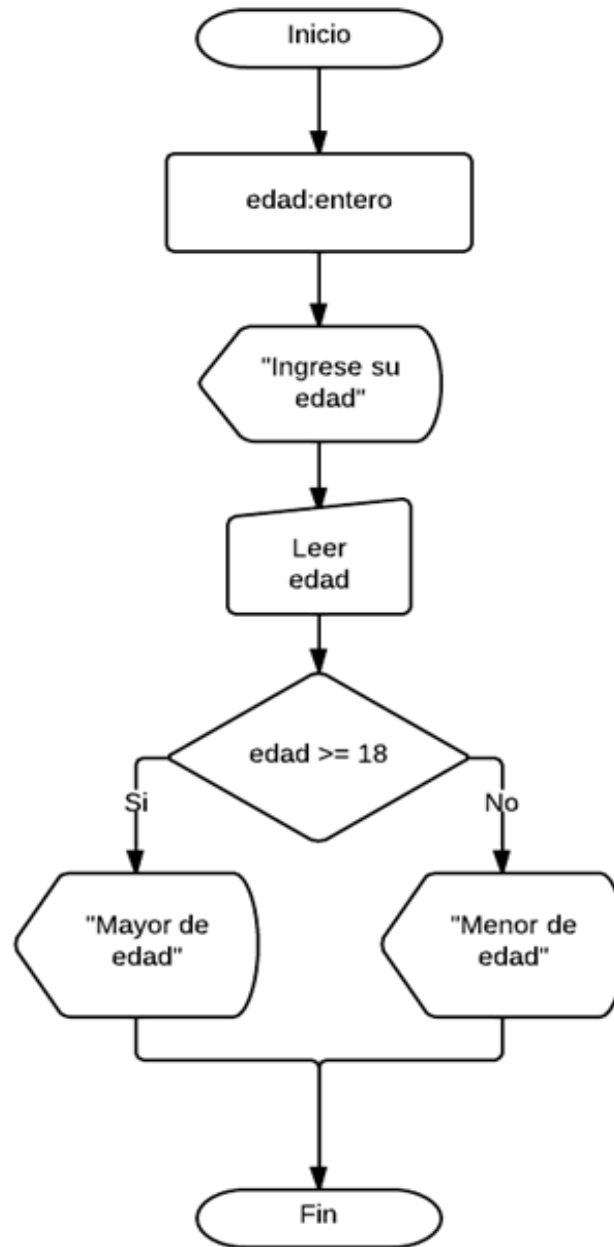
Representación



Ejemplo

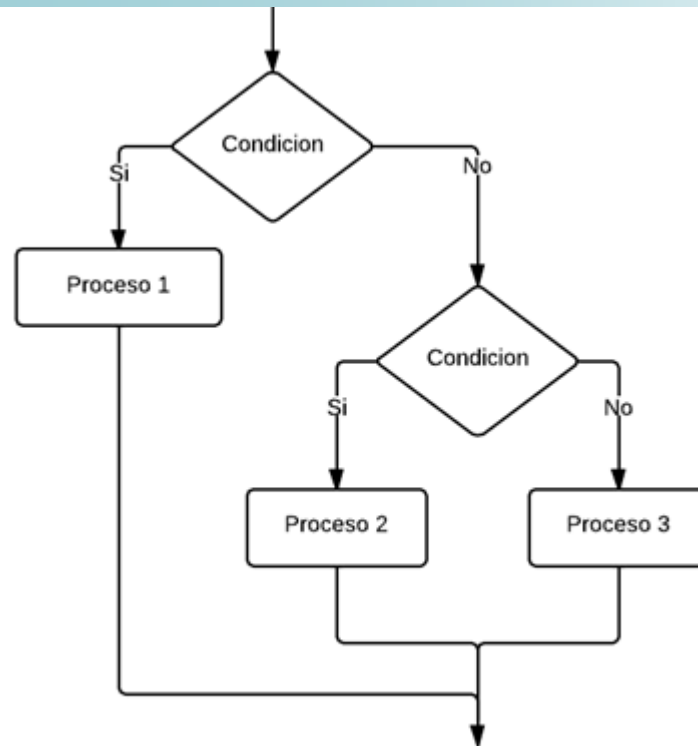
Diseñar un algoritmo que solicite la edad de una persona, si la persona es mayor de edad (mayor o igual a dieciocho años) que despliegue “Mayor de edad” y si la persona es menor de edad que despliegue “Menor de edad”





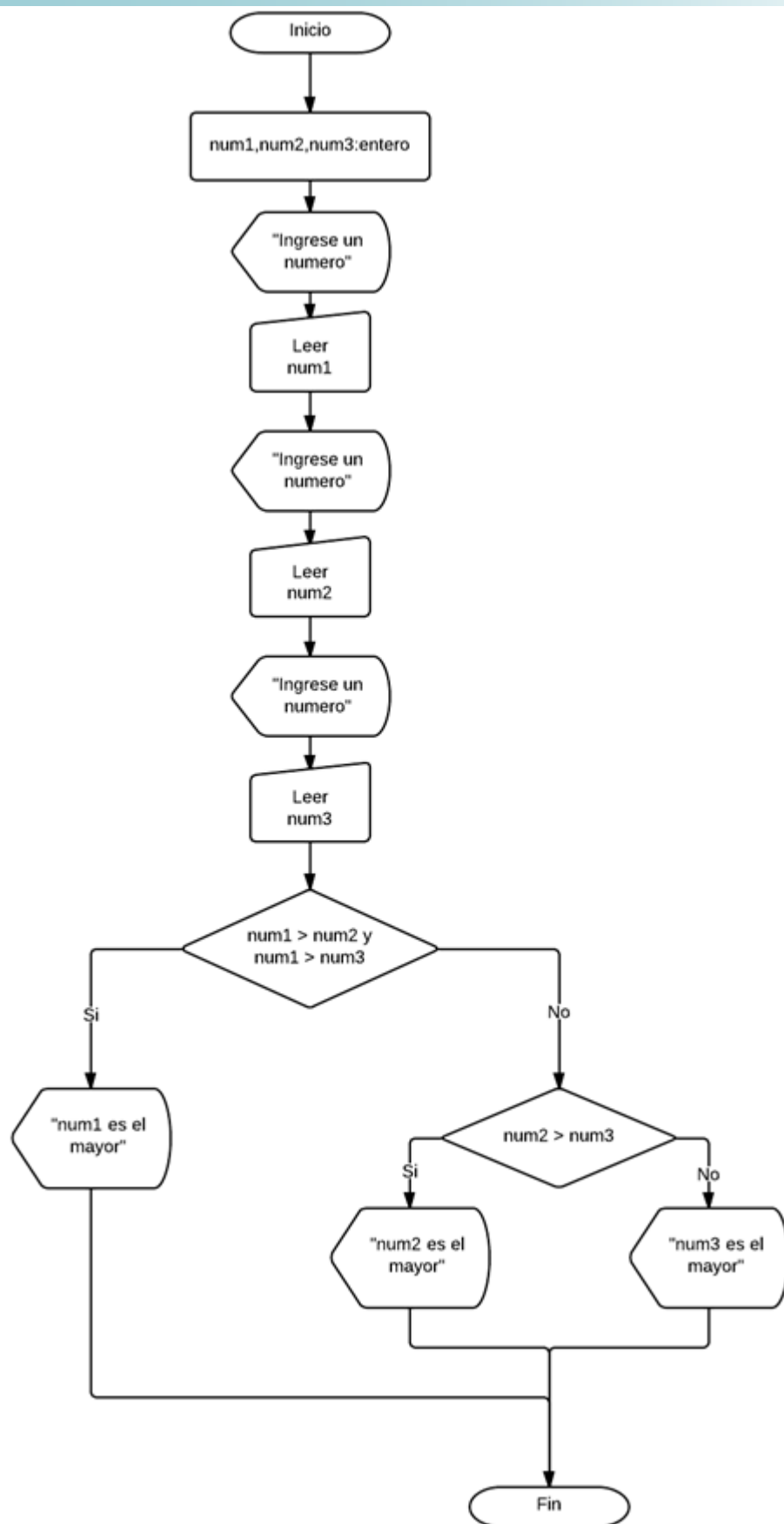
Estructura condicional anidada

Representación



Ejemplo

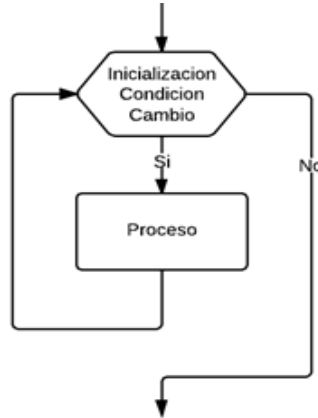
Diseñar un algoritmo que solicite el ingreso de tres números y despliegue cual es el mayor.



## Estructura iterativa

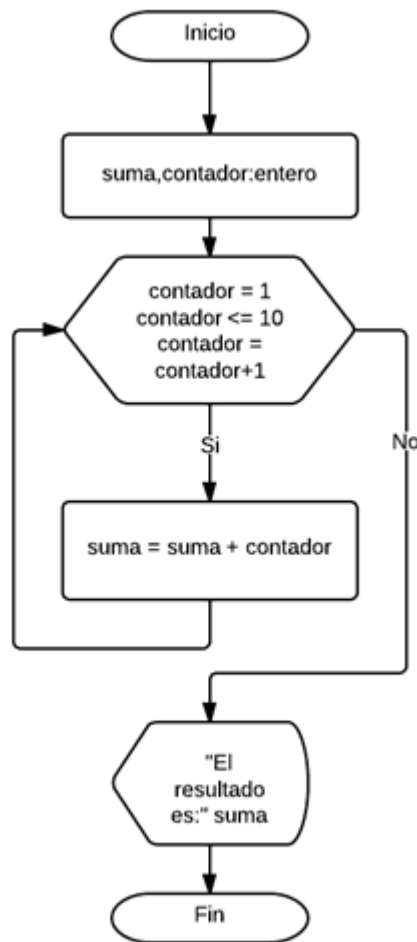
For

Representación



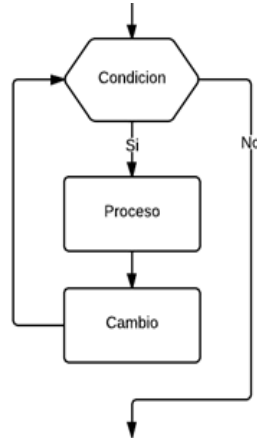
Ejemplo

Diseñar un algoritmo que calcule la suma de los números de 1 a 10.



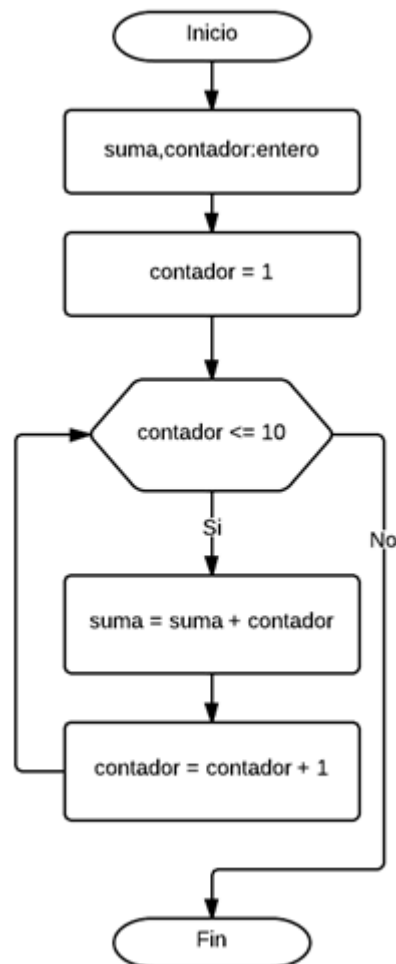
While

Representación

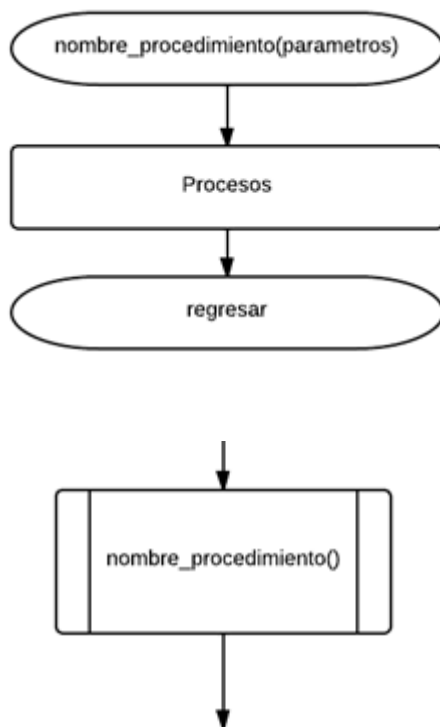


Ejemplo

Diseñar un algoritmo que calcule la suma de los números de 1 a 10.

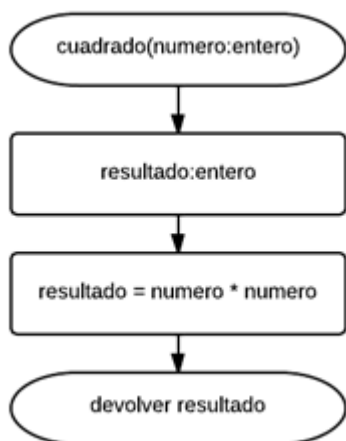


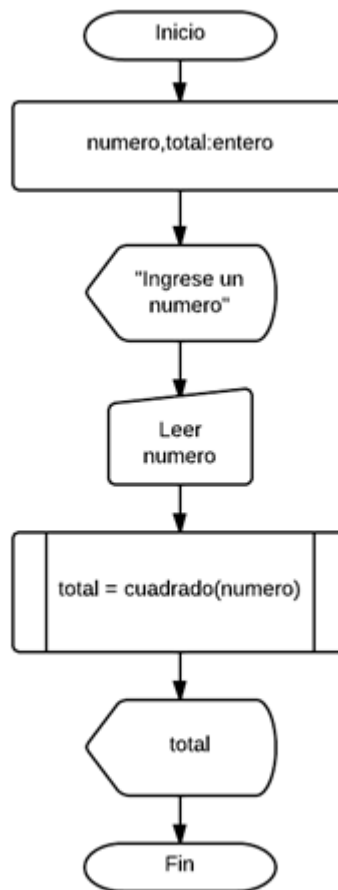
## Representación de un procedimiento



Ejemplo:

Diseñar un algoritmo que calcule el cuadrado de un número y despliegue el resultado.

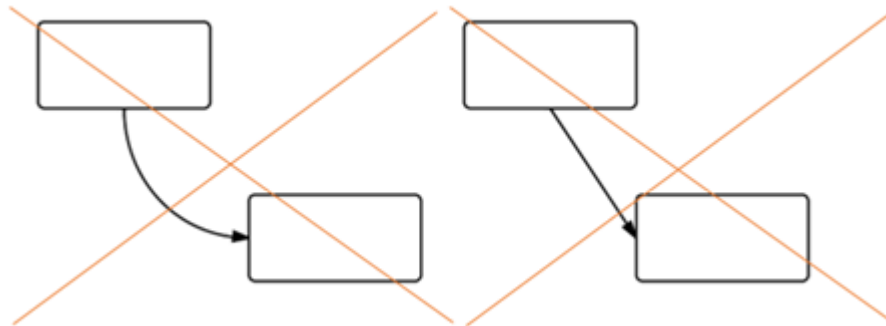




## Reglas para elaborar un diagrama de flujo

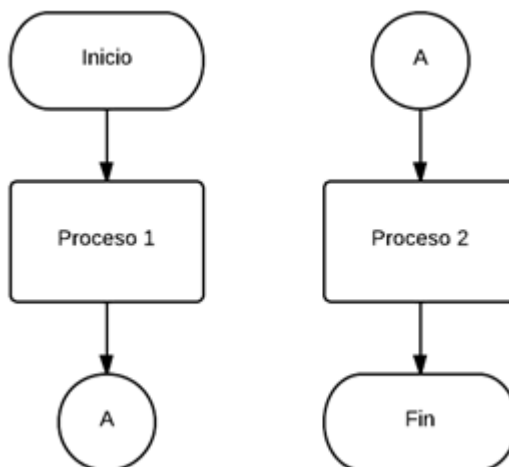
Para dibujar un diagrama de flujo debemos tomar en cuenta las siguientes reglas:

1. El diagrama debe ser construido de arriba hacia abajo y de izquierda a derecha.
2. El diagrama debe tener un inicio y un fin.
3. Las líneas de flujo deben ser rectas, no pueden ser inclinadas o curvas ni tampoco cruzarse entre sí.



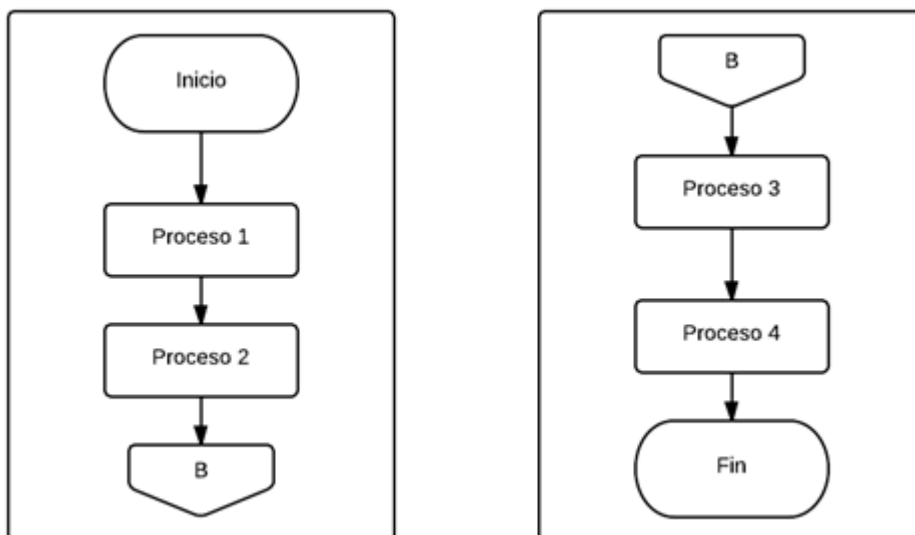
1. Todas las líneas de flujo deben estar conectadas entre sí. Solo puede llegar una línea de flujo a un símbolo.
2. Si el diagrama de flujo requiriera mucho espacio o más de una hoja, se debe utilizar los conectores adecuados y enumerar las páginas para mantener un orden.

## Representación conector en la misma página





## Representación conector en diferente página



## *Descargo de responsabilidad*

La información contenida en este documento descargable en formato PDF o PPT es un reflejo del material virtual presentado en la versión online del curso. Por lo tanto, su contenido, gráficos, links de consulta, acotaciones y comentarios son responsabilidad exclusiva de su(s) respectivo(s) autor(es) por lo que su contenido no compromete al área de e-Learning del Departamento GES o al programa académico al que pertenece.

El área de e-Learning no asume ninguna responsabilidad por la actualidad, exactitud, obligaciones de derechos de autor, integridad o calidad de los contenidos proporcionados y se aclara que la utilización de este descargable se encuentra limitada de manera expresa para los propósitos educacionales del curso.

