

# CULTURA DIGITAL I



PENSAMIENTO ALGORITMICO

# Problema

# CONCEPTO DE PROBLEMA

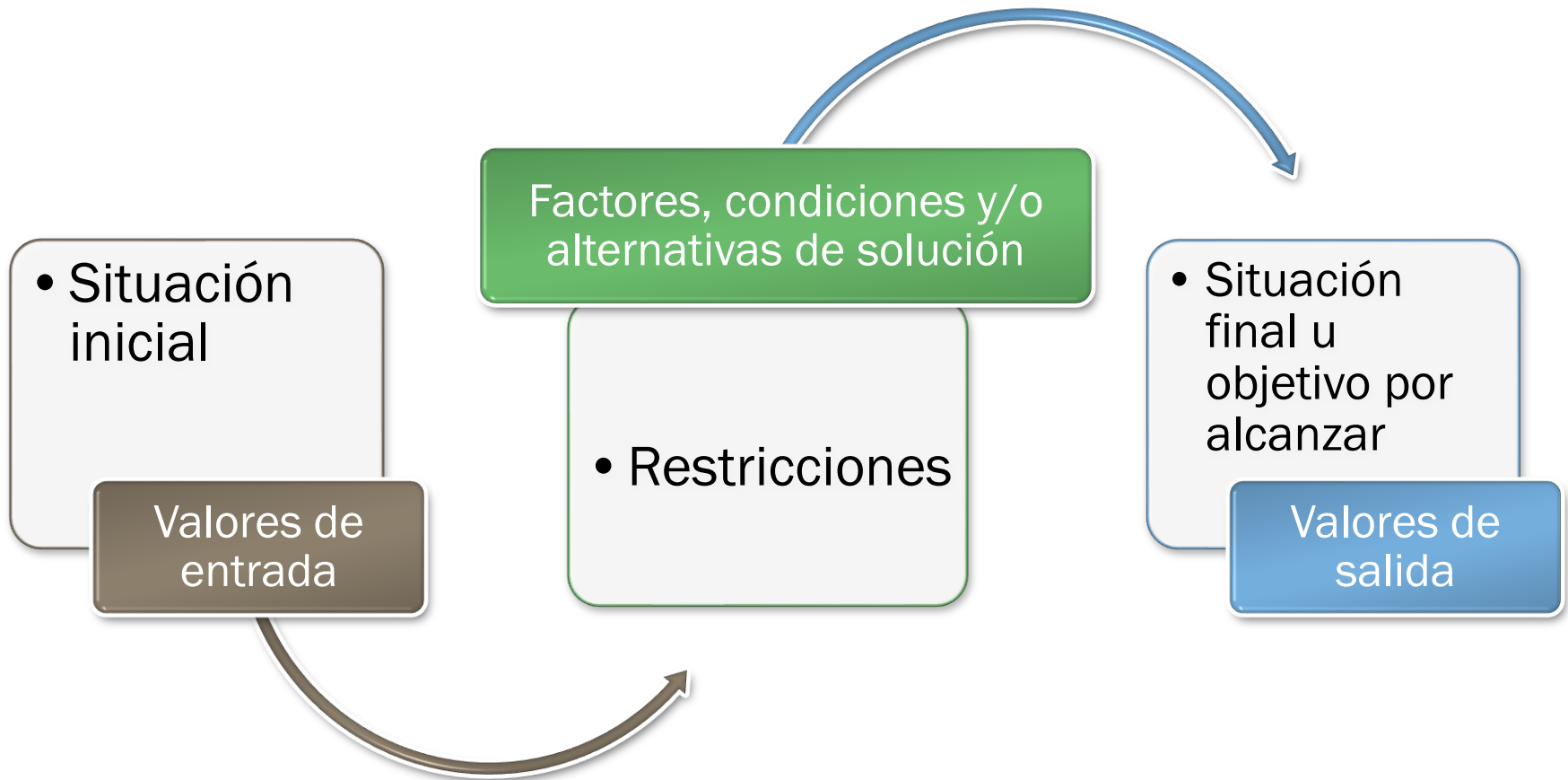
Un problema es un asunto o conjunto de cuestiones que se plantean para ser resueltas.

Asunto o cuestión por tratar

Algo que hay que resolver

*Cuestión o situación inconveniente que obstaculiza el logro de algún fin u objetivo*

# Un problema existe cuando hay tres elementos claramente definidos:



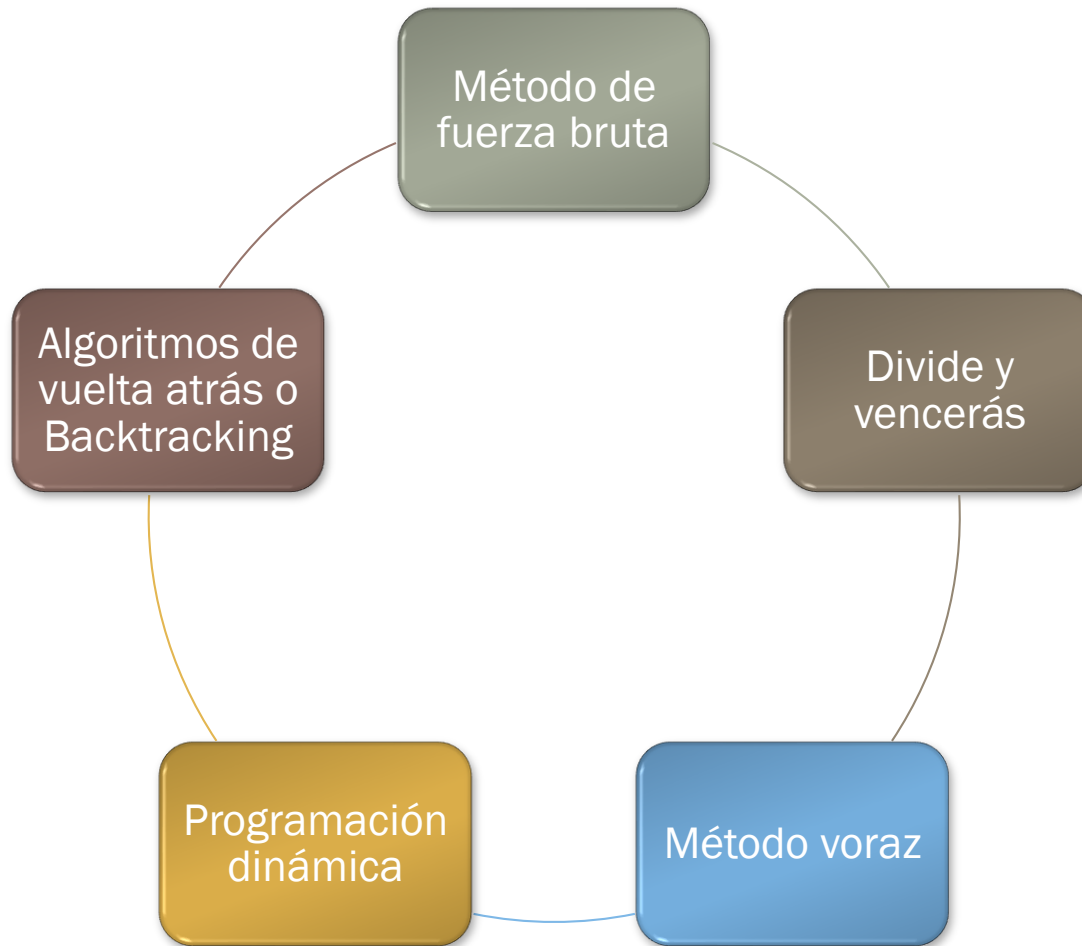
# Fases para la solución de problemas

**Definición del problema:** Es el enunciado del problema, el cual debe ser claro y completo. Es fundamental conocer y delimitar por completo el problema, saber qué es lo que se desea que realice la computadora.

**Análisis del problema:** El problema se analiza teniendo presente la lista de los requisitos dados por el cliente de la empresa o por la persona que encarga el programa.

**Diseño del algoritmo:** Una vez analizado el problema, se diseña una solución que conducirá a un algoritmo que resuelva el problema.

# Métodos de solución de problemas con algoritmos



# Algoritmo

# Definición de algoritmo

Serie de pasos organizados que describe el proceso que se debe seguir para dar solución a un problema específico.

Conjunto de pasos nos permite obtener la solución de un problema.

*Conjunto de pasos ordenados, finitos y precisos que conduce a la resolución de un problema.*



# Los algoritmos se pueden expresar de dos formas:

**Gráfica:** con símbolos, utilizando diagramas de flujo.

**No gráfica:** describiendo las operaciones que llevará a cabo, utilizando un pseudocódigo.

# Los algoritmos también pueden ser:

**Deterministas:** en cada paso del algoritmo se determina, de forma única, el siguiente paso.


**No deterministas:** deben decidir en cada paso de la ejecución entre varias alternativas y agotarlas todas antes de encontrar la solución.

# Las características fundamentales que debe cumplir todo algoritmo son:

**Finito:** el algoritmo debe acabar tras un número finito de pasos.

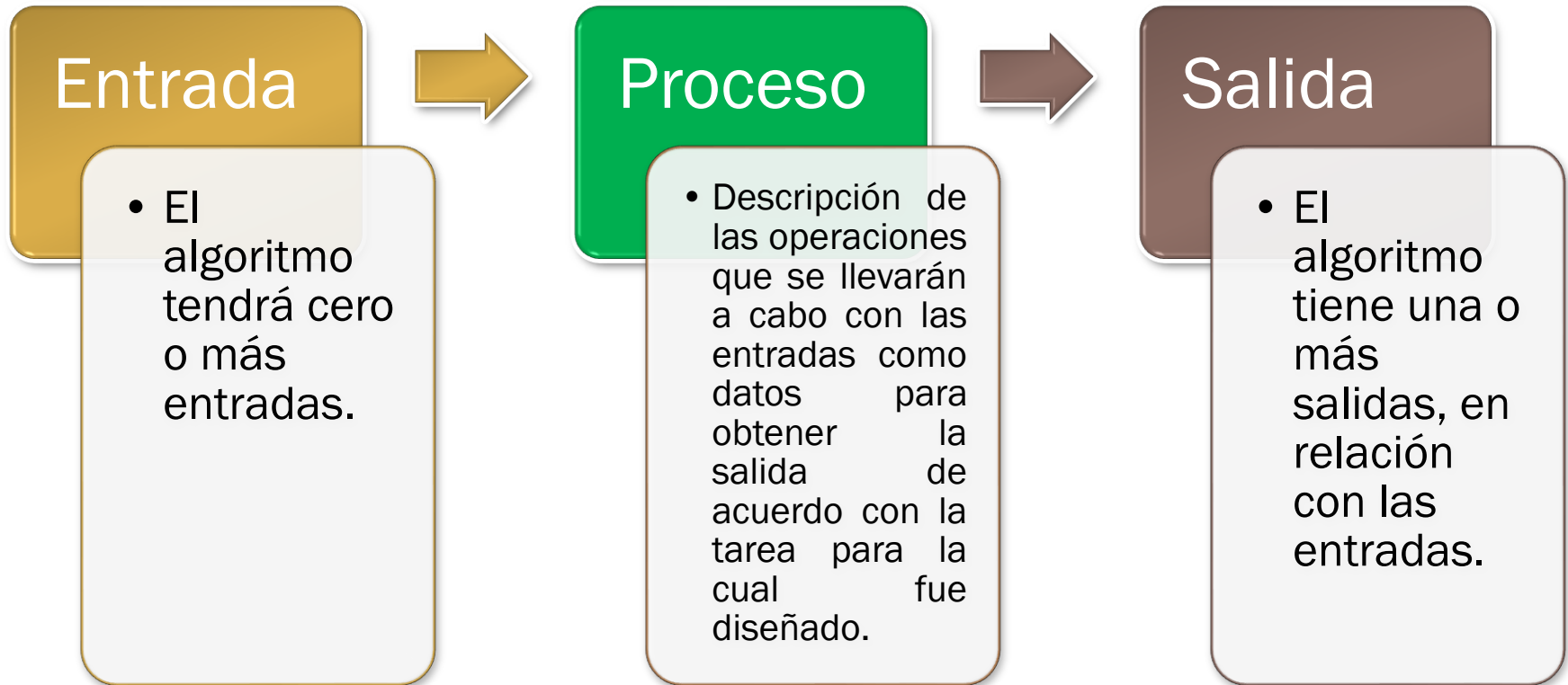


**Definido:** el algoritmo debe definirse de forma precisa para cada paso, es decir, si se sigue dos veces, se debe obtener el mismo resultado.

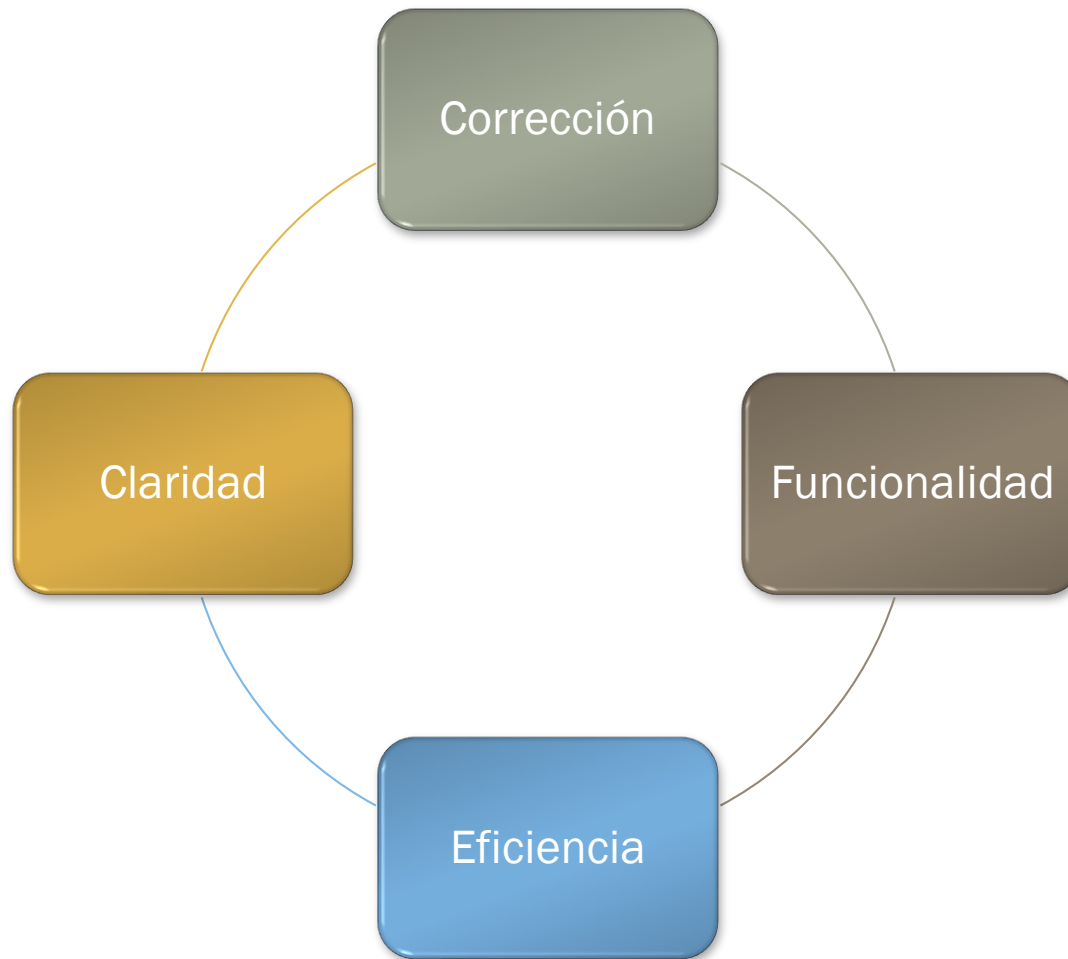


**Preciso:** todas las operaciones que el algoritmo realizará deben ser lo suficientemente claras, es decir, hay que evitar toda ambigüedad al definir cada paso, de modo que puedan ser llevadas a cabo en forma exacta y en un tiempo finito por una persona.

# En la definición de un algoritmo se deben determinar tres partes:



# Factores que identifican la calidad de los algoritmos





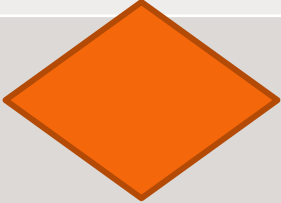

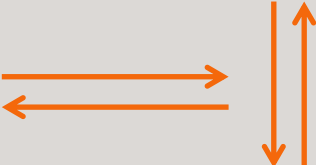
# Diagramas de flujo

# CONCEPTO DE DIGRAMAS DE FLUJO

Son esquemas que sirven para expresar de manera gráfica los pasos para la solución de un problema utilizando una serie de símbolos que ilustran las operaciones y la secuencia de los datos.

Técnica que representa gráficamente un algoritmo. Se utiliza para resolver problemas mediante diagramas a través del uso de la computadora.

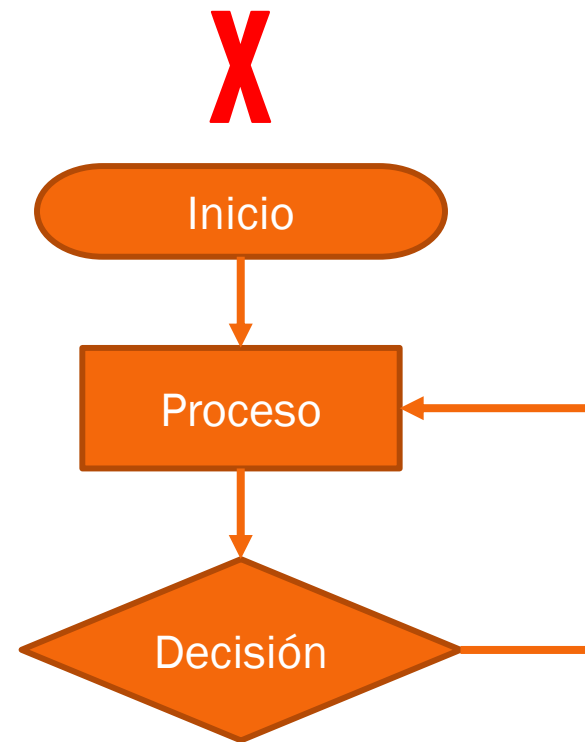
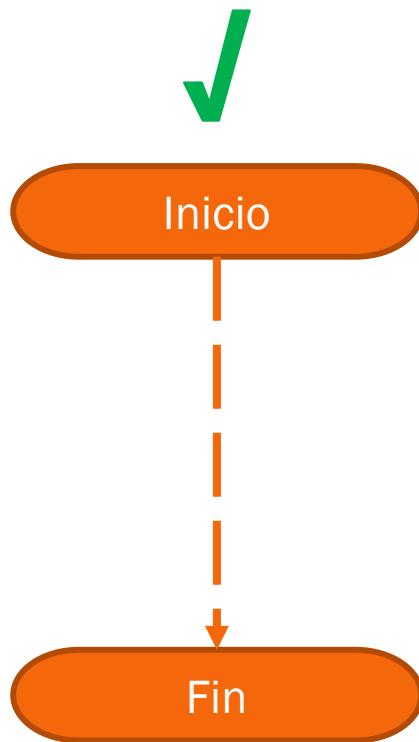
# SIMBOLOS DEL DIAGRAMA DE FLUJO

Simbolo	Nombre	Descripción
	Inicio o Terminador	Este símbolo inicia y termina el diagrama de flujo.
	Entrada y salida de datos	Representa la entrada y salida de datos.
	Decisión	Representa una decisión, comparación, evaluación, validación, etc. de las variables o datos de entrada y salida.
	Proceso	Este símbolo contiene una expresión que procesa los datos, es decir, contiene las instrucciones para procesar los datos de entrada.
	Flechas o dirección del flujo	Indican la dirección del flujo, es decir, la secuencia de la realización de las instrucciones del diagrama.



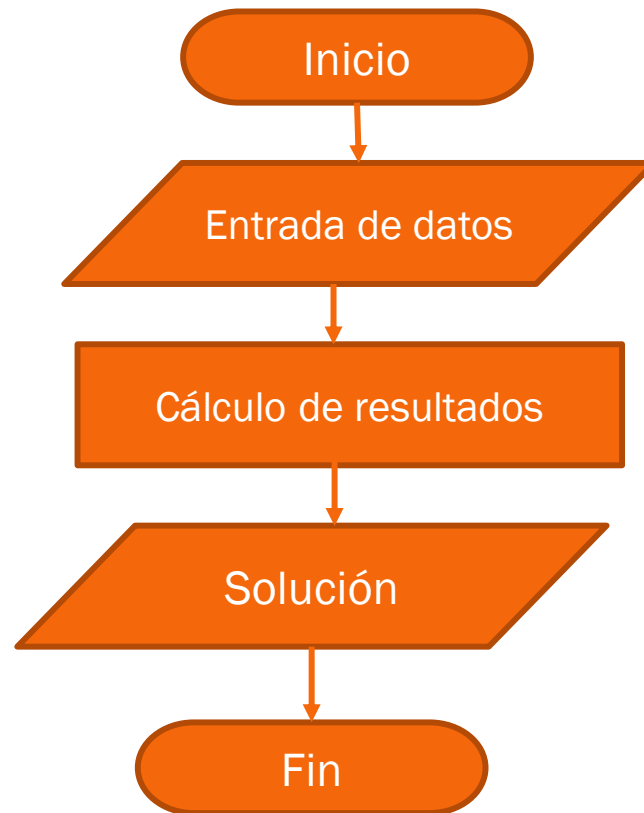
# CONSIDERACIONES AL CONSTRUIR DIAGRAMAS DE FLUJO

- Se debe tener un inicio y un fin, y tener cuidado de no caer en redundancias o procesos cíclicos que nunca terminen.



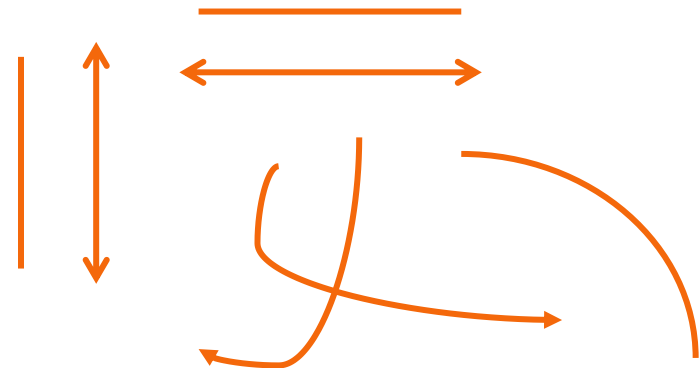
# CONSIDERACIONES AL CONSTRUIR DIAGRAMAS DE FLUJO

- ☞ Tiene que existir una ruta para lograr la solución del problema



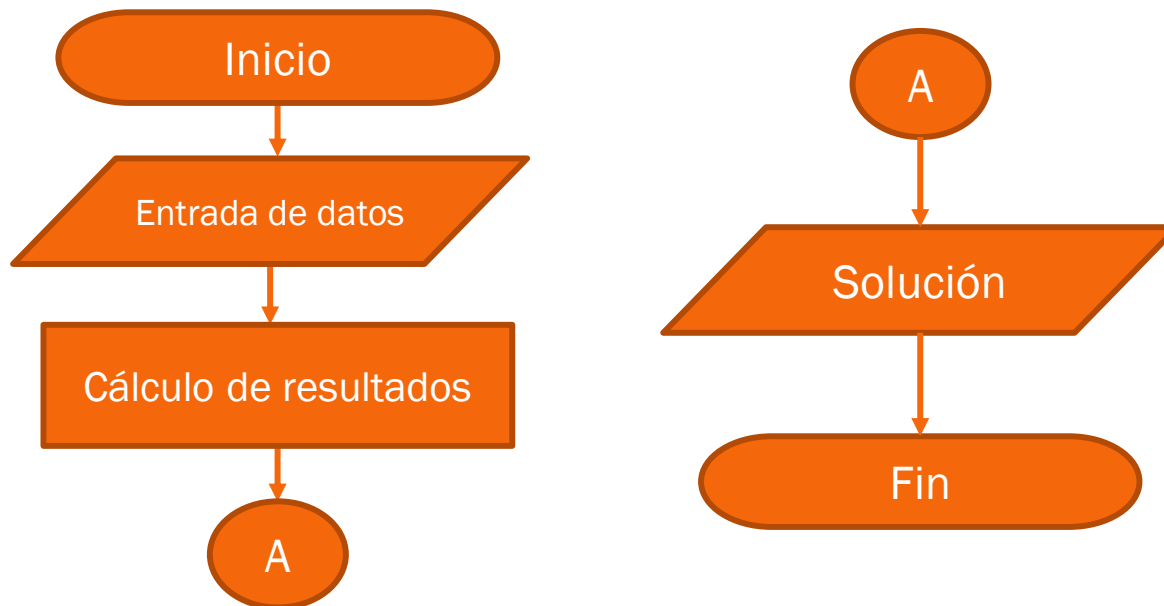
# CONSIDERACIONES AL CONSTRUIR DIAGRAMAS DE FLUJO

- ∞ El flujo o camino del proceso del diagrama se señala con flechas rectas de un solo sentido en las direcciones arriba-abajo o izquierda-derecha. Ninguna línea debe quedar sin conectar o cruzarse con otra



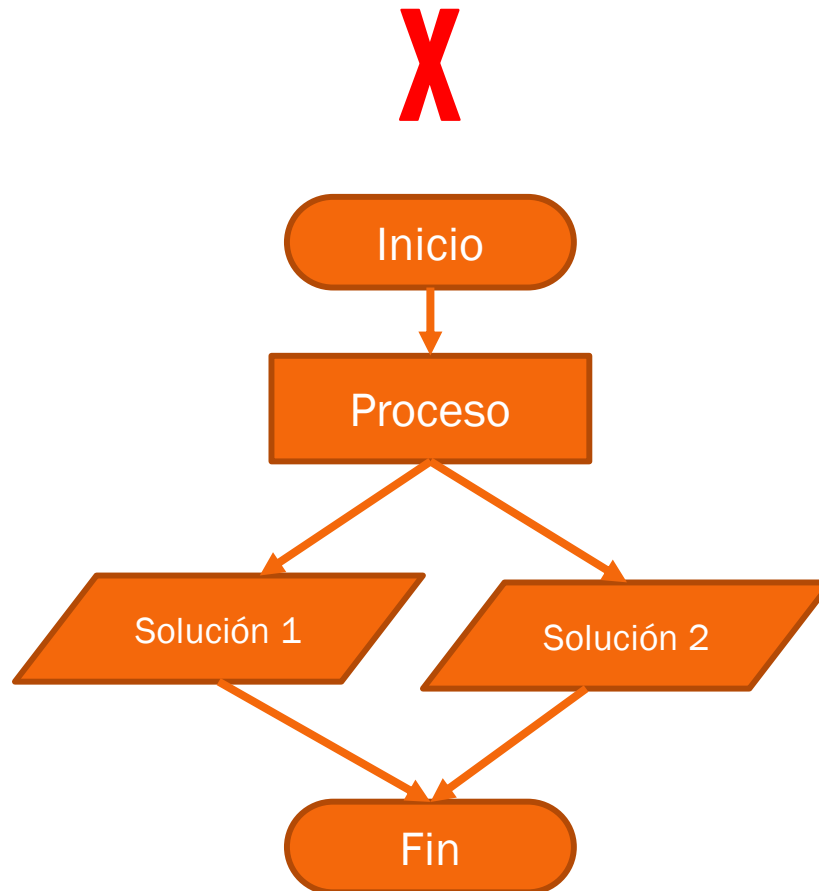
# CONSIDERACIONES AL CONSTRUIR DIAGRAMAS DE FLUJO

- Los enunciados de entrada y salida, así como las operaciones y sentencias de decisión, deben estar correctamente redactadas para evitar confusiones en el flujo del diagrama.
- En caso de que el diagrama sea extenso, se utilizan conectores para indicar la continuidad del desarrollo del proceso.



# CONSIDERACIONES AL CONSTRUIR DIAGRAMAS DE FLUJO

- ⌘ No se puede obtener de un mismo proceso dos resultados diferentes.



# Pseudocódigo y estructuras de control

# CONCEPTO DE PSEUDOCÓDIGO

∞ Es un lenguaje que se asemeja a los lenguajes de programación en lo que se crea el software para computadora. Utiliza un lenguaje no formal para definir la secuencia.

∞ Ejemplo:

∞ Leer a

∞ Leer b

∞  $e \leftarrow a + b$

∞ Mostrar e

# Características de los pseudocódigos

Emplea palabras normales de un idioma, por ejemplo: inicio, leer, escribir, imprimir, calcular, etc.

No existe vocabulario obligado aunque por el uso frecuente y la comodidad se han establecido estándares.

Las instrucciones deben ser escritas con precisión, sin ambigüedades como podría suceder con el método de texto narrativo.

Las instrucciones aunque escritas con precisión no requieren la rigurosidad de las empleadas en un lenguaje de programación.

Un algoritmo representado en pseudocódigo puede convertirse con facilidad en código de cualquier lenguaje de programación



# ESTRUCTURAS DE CONTROL

En los algoritmos, diagramas de flujo y pseudocódigos es común utilizar las estructuras de control, las cuales permiten, como su nombre lo indica, controlar el flujo de los datos durante su ejecución o proceso.

Las estructuras son de tipo secuencial, condicionales ó ciclicas.

# Ejemplos de estructuras de control

## ∞ Secuencial

- Inicio
  - Levantar la bocina del teléfono
  - Esperar tono de marcado
  - Marcar número telefónico
  - Esperar a que contesten
  - Hablar con la otra persona
  - Colgar la bocina
- Fin

# Ejemplos de estructuras de control

## ∞ Condicional

- Inicio
  - Solicitar calificación entre 0 y 10
  - Si calificación  $\geq 6$  entonces:
    - Mostrar “aprobado”
  - Si no:
    - Mostrar “Reprobado”
- Fin

# Ejemplos de estructuras de control

## ∞ Cíclica

- Inicio
  - Suma = 0
  - Repite
  - $\text{Suma} = \text{Suma} + 1$
  - Hasta Suma = 200
  - Suma
- Fin