

# UNIDAD 1. COMPRESION DE PROBLEMAS

## ESTRATEGIAS DE RESOLUCION

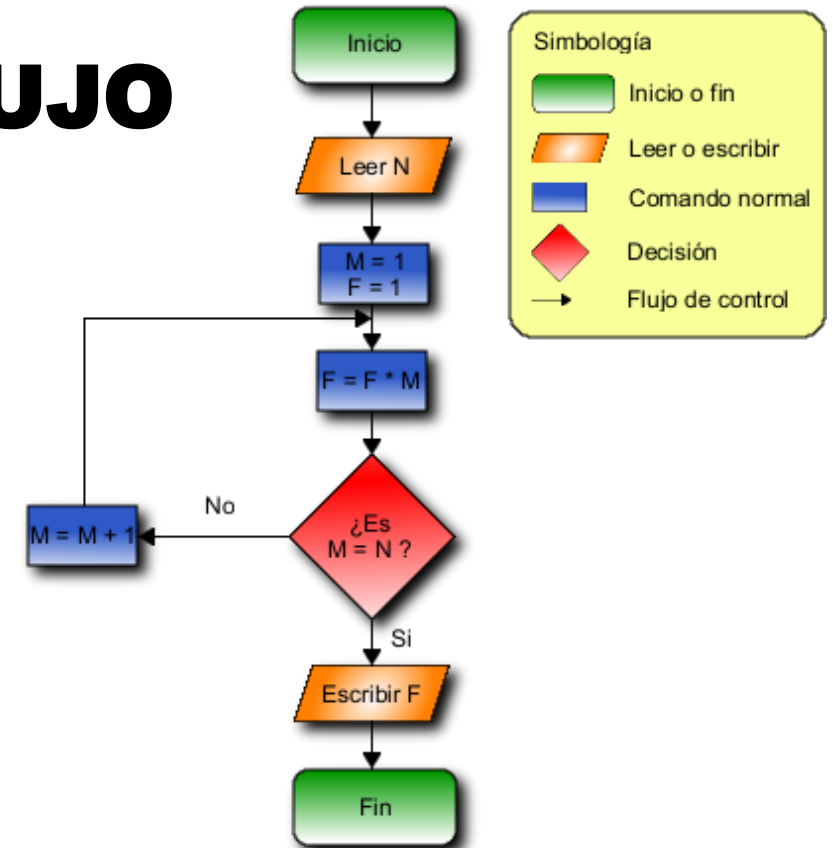
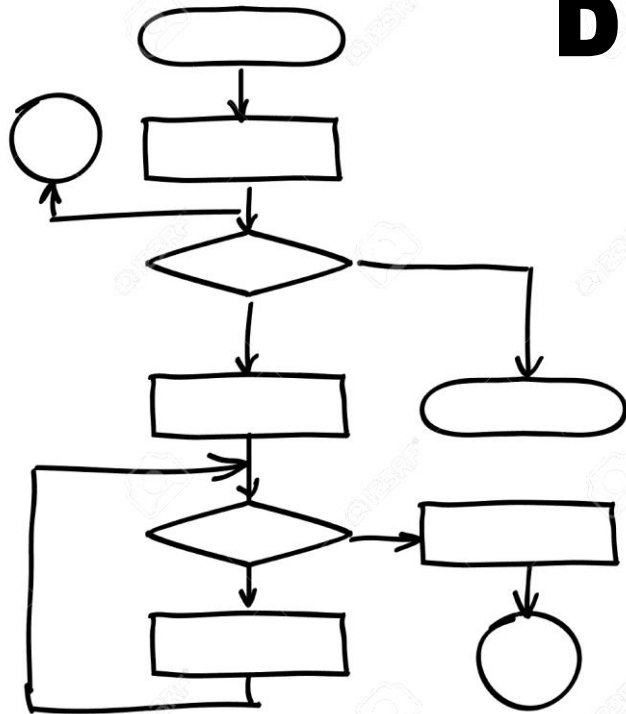
- Clasificación de los tipos de datos
  - Simples (Numéricos y No Numéricos)
  - Estructurados (Estático y Dinámico)
- Expresiones:
  - Aritméticas (cuadrado =  $x * x$ ;
  - de relación (PrecioNeto \* 1.19 > 45000)
  - Lógicas (Edad > 18 AND Edad < 25)
  - Compuestas
- Clasificación de los problemas elementales.
  - Problemas de Evaluación.
  - Problemas de Decisión.
  - Problemas Compuestos.
- Las partes principales de un problema:
  - Resultados
  - Datos
  - Condiciones Vinculantes.
- Diccionarios. Constantes y Variables. Identificadores

TEMA COMPRENDIDO  
PPT 01

# ESTRUCTURA ELEMENTAL DE DATOS

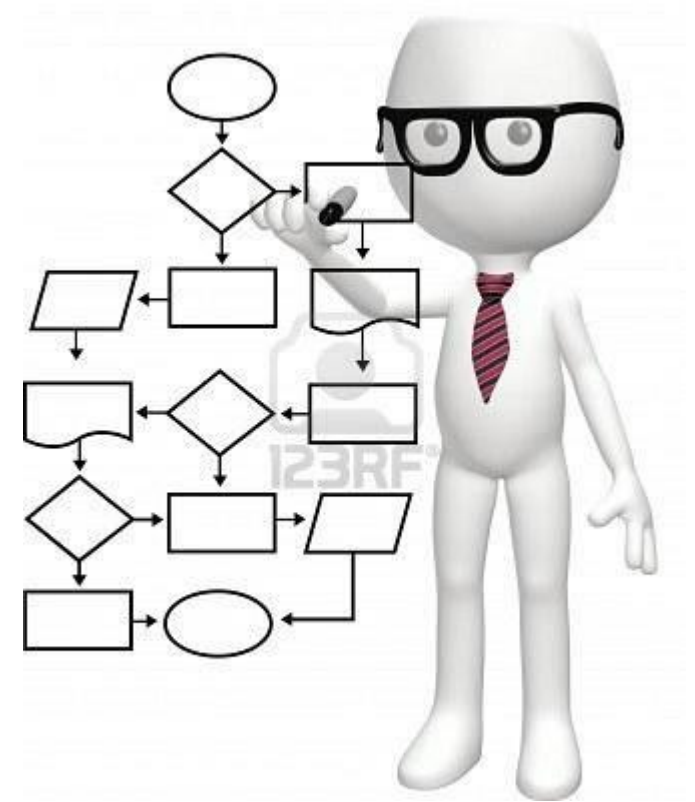
## DISEÑO DE ALGORITMOS

### DIAGRAMAS DE FLUJO



## TEMAS:

1. Concepto y definición de algoritmo.
2. Su representación gráfica: el diagrama de flujo lógico.
3. Símbolos utilizados.
4. Ventajas de la diagramación.
5. Prueba de escritorio.
6. Pautas básicas para el diseño general de un algoritmo.
7. El diseño descendente.
8. El teorema fundamental de la programación estructurada.
9. Estructuras: secuencial, de selección y repetición.



# Que es un Algoritmo

Es un conjunto de pasos para conseguir un objetivo *o bien como un conjunto de instrucciones que realizadas en orden conducen a obtener la solución de un problema.*

*Para realizar un programa es conveniente el diseño o definición previa del algoritmo.*

Los algoritmos son independiente de los lenguajes de programación.

En cada problema el algoritmo puede escribirse y luego ejecutarse en un lenguaje diferente de programación.

# Nuestra vida está llena de Algoritmo, Por ejemplo el Algoritmo de Tomar una Clase Virtual sería:

1. Controlar el horario de clase
2. Prender la computadora
3. Iniciar el programa de videoconferencia
4. Conectarse
5. Tomar la clase
6. Entendió?
7. Si es afirmativo, realizar los prácticos
8. Si es negativo, preguntar o asistir a consulta

*Por lo tanto Diseñar un algoritmo es una labor inmensamente creativa.*

Según **Donald Knuth** (científico de computación), para que sea considerado un **algoritmo informático** debe cumplir las siguientes condiciones:

- 1) **SECUENCIAL:** *“Los pasos deben tener un orden perfectamente definido”.*
- 2) **DEFINIDO:** *“Que las operaciones a llevar a cabo deben ser especificadas de manera exacta y precisa, y no confusa para cada caso”. Por ej. División por cero*
- 3) **GENERAL:** *“Al encararse la solución de un problema, ésta debe desarrollarse para la generalidad de los casos que pueden presentarse dentro del ámbito de validez de esa solución”. Por ej. Evitar de hacer algoritmo usando valores constantes sino con variables.*
- 4) **FINITO:** *"Un algoritmo siempre debe terminar después de un número finito de pasos".*

## Ejemplo 1:

Realice el algoritmo para el siguiente enunciado:

1. SECUENCIAL
2. DEFINIDO
3. GENERAL
4. FINITO:

*Ingresar tres valores enteros, calcule y muestre el promedio.*

1. Inicio
2.  $N=3$
3. Ingresar A, B, C
4.  $S = A + B + C$
5.  $P = S/N$
6. Mostrar P
7. Fin

## Ejemplo 2:

Indique si el siguiente algoritmo es o no un Algoritmo Informático:

### Algoritmo Informático

1. **SECUENCIAL**
2. **DEFINIDO**
3. **GENERAL**
4. **FINITO:**

*Ingresar dos valores enteros, calcule y muestre el cociente entre ambos valores.*

1. Inicio
2. Ingresar A, B
3.  $C = A/B$
4. Mostrar C
5. Fin

*El algoritmo anterior no es un algoritmo informático porque no cumple las condiciones de ser:*

**General:** no cumple para la generalidad del universo de datos ( $B=0$ ).

**Definido:** cuando B toma el valor cero, el resultado no está definido.

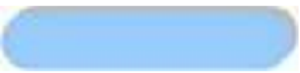
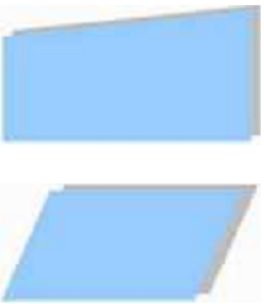



# **DIAGRAMAS DE FLUJO**


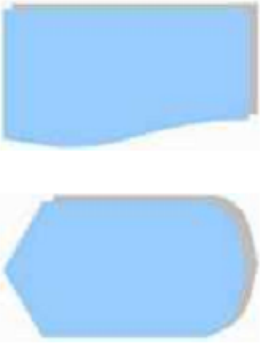
Es un esquema para representar gráficamente un algoritmo. Se basan en la utilización de diversos símbolos para representar operaciones específicas.

Se les llama diagramas de flujo porque los símbolos utilizados se conectan por medio de flechas para indicar la secuencia de operación.



Los símbolos usados son Universales

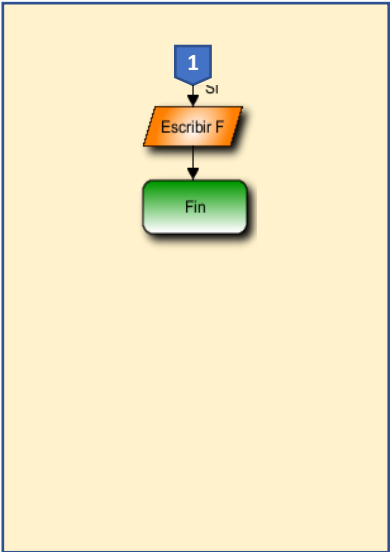
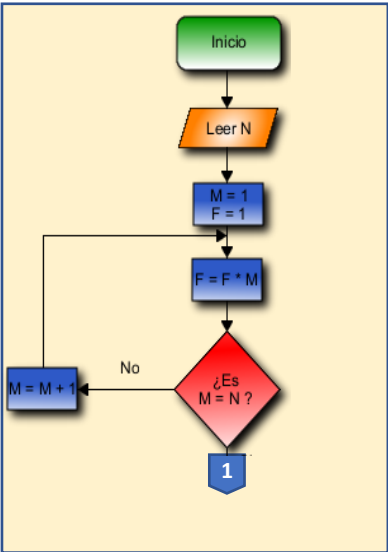
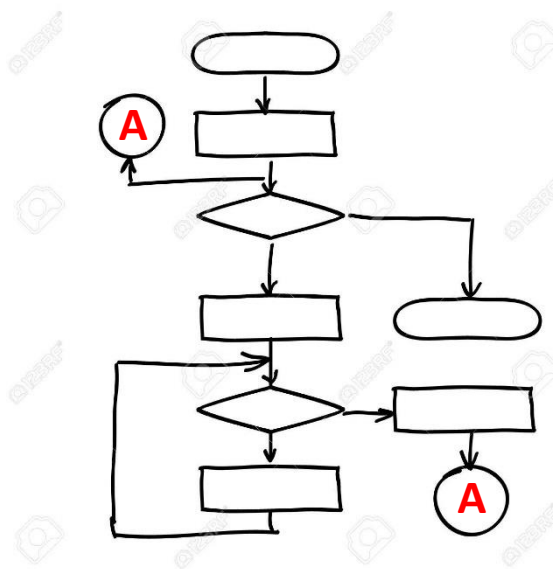
Símbolo	Descripción
	<b>Inicio / Terminación.</b> Este símbolo se utiliza para señalar el comienzo así como el final de un diagrama. Tradicionalmente se colocan las palabras “INICIO” ó “FIN” dentro de la figura para hacerlo más explícito. Es el único símbolo que solamente tiene una conexión (flecha) ya sea de salida, en el de inicio, o de entrada, para el de fin.
	<b>Entrada de datos.</b> En este símbolo se indican la lectura de las variables primarias, dándole los valores iniciales que deberán recibir en el proceso. Esto se hace colocando los identificadores de dichas variables en el interior de la figura. El trapecio normalmente se utiliza para entrada por teclado y el paralelogramo para cualquier dispositivo de entrada. Este símbolo siempre deberá tener al menos una conexión entrante (generalmente del inicio) y una de salida.
	<b>Proceso de datos.</b> Este símbolo lo utilizaremos para señalar operaciones matemáticas, aritméticas o procesos específicos que se realicen con nuestros datos. La manera de anotar dichos procesos, puede ser mediante una descripción breve de la operación o mediante una asignación de dicha operación hacia una variable como por ejemplo: $R = A + B$ Este símbolo siempre deberá tener al menos una conexión de entrada y una de salida.

# DIAGRAMAS DE FLUJO

	<p><b>Decisión.</b> Este símbolo nos representa una decisión. En su interior se anota una expresión lógica o pregunta que pueda ser evaluada como cierta o falsa y que determinará el flujo del programa.</p> <p>Este símbolo es el único que puede contener dos salidas y en cada una de las salidas se suele poner un rótulo de “si/no” indicando con esto cual de ellas se tomará según el resultado de la evaluación de la decisión.</p> <p>Es una buena práctica de diagramación utilizar siempre el mismo lado para el “si” siempre que esto sea posible, normalmente a la derecha.</p>
	<p><b>Desplegado de información.</b> Este símbolo se utiliza para mostrar un resultado, el cual puede representar la solución al problema que se pretende resolver y que fue conseguida a través del resto del diagrama.</p> <p>Dentro de su interior se anotará el identificado del resultado o un valor constante que represente el mismo, acompañado de un mensaje aclaratorio.</p> <p>El primer símbolo se usa habitualmente para salida impresa y el otro para salida por monitor o pantalla. Este símbolo siempre deberá tener al menos una conexión de entrada y una de salida.</p>

En la diagramación, también contamos con una serie de símbolos auxiliares que no intervienen en el proceso del algoritmo, pero que pueden ser útiles para ayudarnos a dar claridad a nuestros diagramas, algunos de ellos son los siguientes:

Símbolo	Descripción
	<p><b>Conector.</b> Este símbolo se utiliza para indicar un salto dentro del diagrama. Se utiliza con el propósito de facilitar la disposición plana de un diagrama y evitar el cruce excesivo de líneas a través del mismo.</p> <p>Este conector va asociado a un conector “gemelo” y junto con él, representa una puerta de entrada y de salida para el flujo del diagrama, es decir que cuando una flecha termina en un conector marcado con la letra “A”, se continuará el diagrama a partir de otro conector marcado con la misma letra tal como si se tratara de una línea continua ininterrumpida.</p>
	<p><b>Conector de página.</b> Este conector es idéntico en funcionamiento que el anterior, pero su forma pentagonal lo distingue y nos indica que debemos buscar el “gemelo” en una página distinta de la actual. Este conector lleva asociado una especie de salto entre páginas.</p>



Ejemplo de Diagrama de Flujo: Realice el diagrama de flujo para el siguiente enunciado:

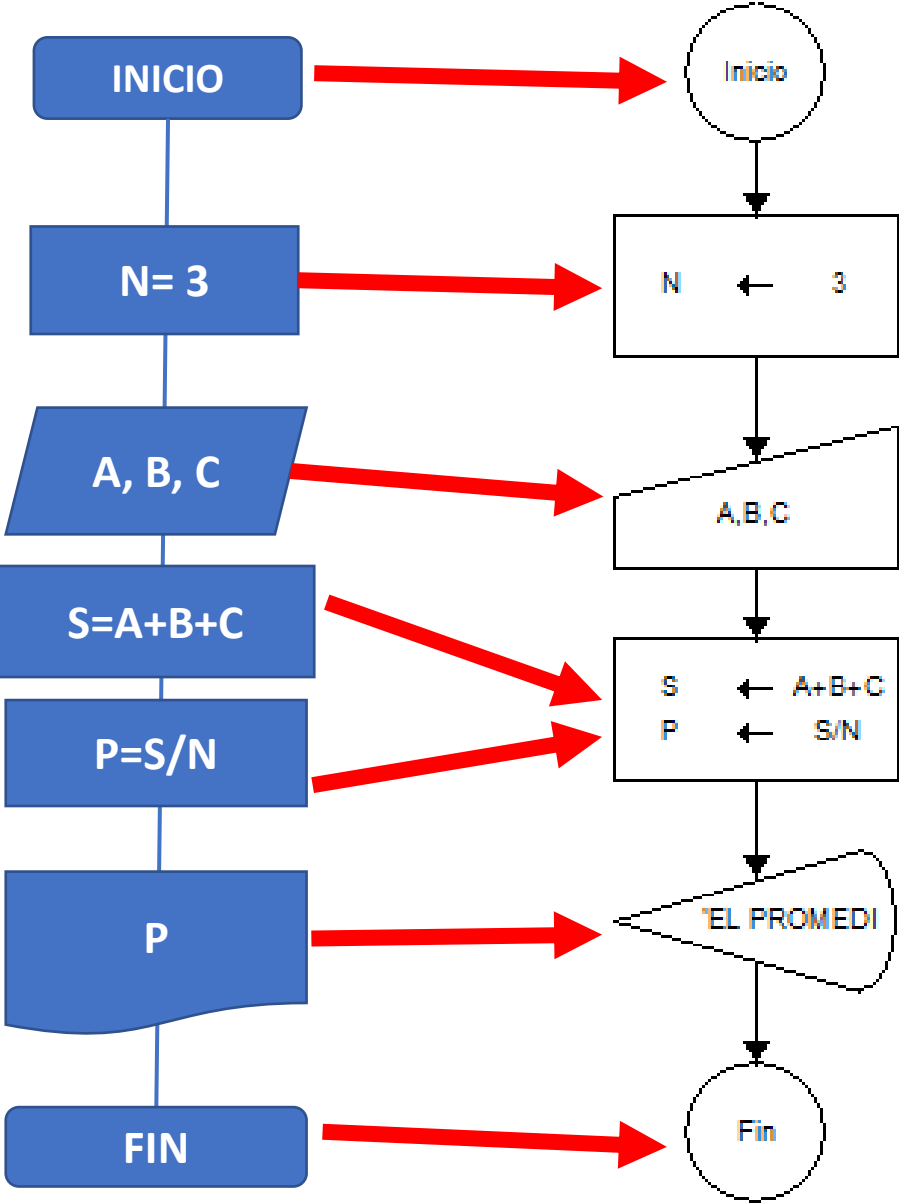
*Ingresar tres valores reales, calcule y muestre el promedio.*

CONDICIONES VINCULANTES

Número	Descripción
1	N=3
2	Ingresar A, B, C
3	$S = A + B + C$
4	$P = S / N$
5	Mostrar P

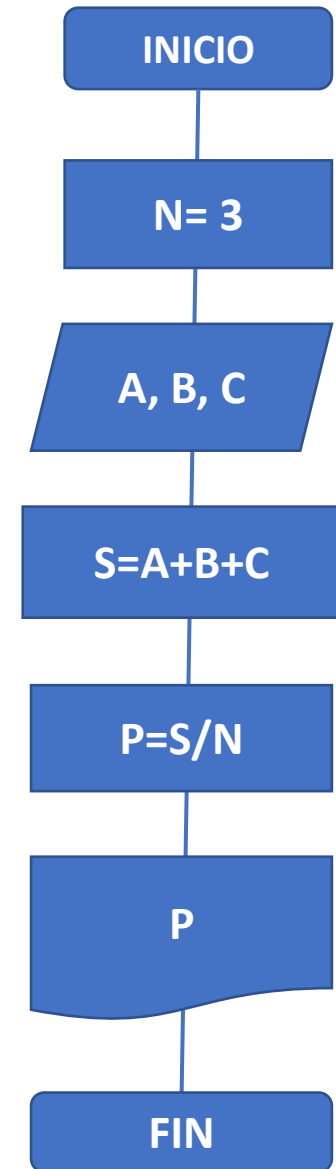
ALGORITMO

- 1. Inicio
- 2. N=3
- 3. Ingresar o leer A, B, C
- 4.  $S = A + B + C$
- 5.  $P = S / N$
- 6. Mostrar P
- 7. Fin



## VENTAJAS DE LA DIAGRAMACIÓN

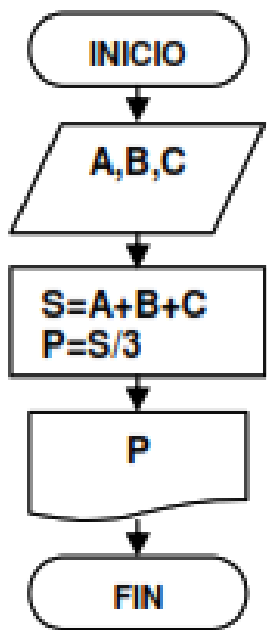
- Favorecen la comprensión del proceso mostrándolo como un dibujo. El cerebro humano reconoce fácilmente los dibujos.
- **De uso:** Facilita su empleo.
- **De destino:** Permite la correcta identificación de actividades.
- **De comprensión e interpretación:** Simplifica su comprensión.
- **De simbología:** Disminuye la complejidad y accesibilidad.
- **De diagramación:** Se elabora con rapidez y no requiere de recursos sofisticados.



# PRUEBA DE ESCRITORIO

La prueba de escritorio es una ejecución ‘a mano’ del algoritmo, por lo tanto se debe llevar registro de los valores que va tomando cada una de las variables involucradas en el mismo.

La prueba de escritorio es una tabla en la cual en la primera fila se colocan todos los identificadores de las variables y constantes y a continuación se completa con los valores que toma cada una de ellas.



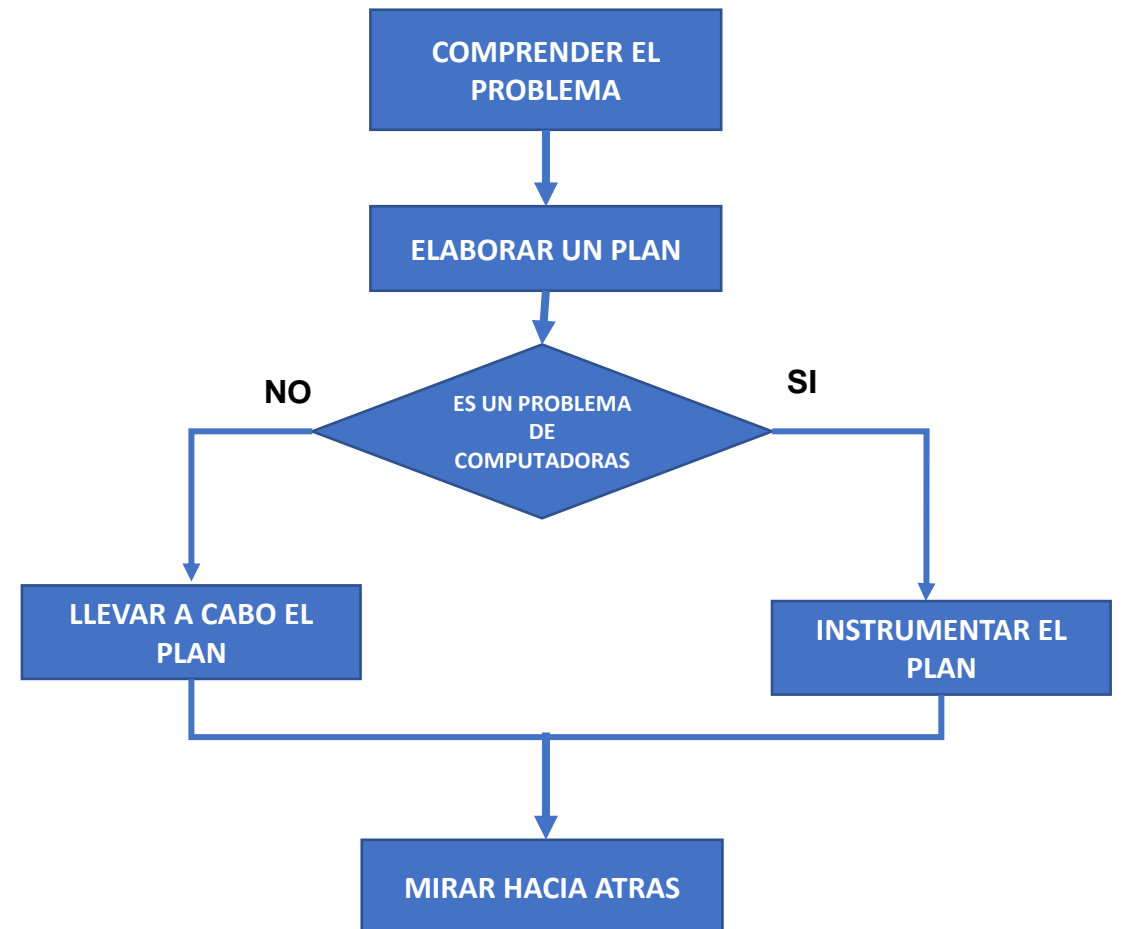
<u>PRUEBA DE ESCRITORIO</u>				
A	B	C	S	P
3	4	5	12	4

# PAUTAS BÁSICAS PARA EL DISEÑO GENERAL DE UN ALGORITMO

Según el matemático George Polya, brindan una serie de pautas para conducir nuestro pensamiento durante el proceso de solución de un problema, disponiendo así de un método estratégico para la resolución de problemas.

## ENUNCIADO GENERAL DE LOS PRINCIPIOS DE GEORGE POLYA

1. Comprender el problema.
2. Elaborar un plan.
3. ¿Es un problema de computadora?
  - Afirmativo: Instrumentar el plan.
  - Negativo: Llevar a cabo el plan.
4. Mirar hacia atrás.





## 1. Comprender el problema.

1. ¿Cuáles son los input o entradas?
2. ¿Cuáles son los output o salidas deseados?
3. ¿Cuál o cuáles serían las condiciones vinculantes?
4. Dibujar una figura representativa esquematizando el enunciado del problema.
5. Introducir una notación adecuada, a usar durante el desarrollo resolutivo del problema.
6. Separar clara y adecuadamente las distintas partes del problema.

## 2. Elaborar un Plan

1. Encontrar conexiones entre las entradas y salidas
2. ¿Puede derivarse algo del uso de determinados input?
3. Tal vez sea útil introducir algún problema auxiliar.
4. Tratar de hacer una buena suposición.
5. ¿Vio antes este problema?
6. ¿Conoce Usted algún problema similar?
7. Mirar lo desconocido.
8. Tal vez convenga simplificar el problema tratando de acercarnos a la solución.

## 3. ¿Es un problema de computadora?

Revisar si cumple con las condiciones para que el algoritmo sea considerado informático: **Secuencial – Definido – Finito – General.**

**Afirmativo:** Instrumentar el plan, elaborando el algoritmo correspondiente.

**Negativo:** Llevar a cabo el plan.

## 4. Mirar hacia atrás

1. Revisar lo hecho y preguntarnos si podemos mejorar el plan.
2. ¿Podríamos usar esta solución en otras soluciones?
3. ¿Puede usted controlar los resultados?
4. En base a los estudios anteriores se podrían derivar los resultados en forma diferente?

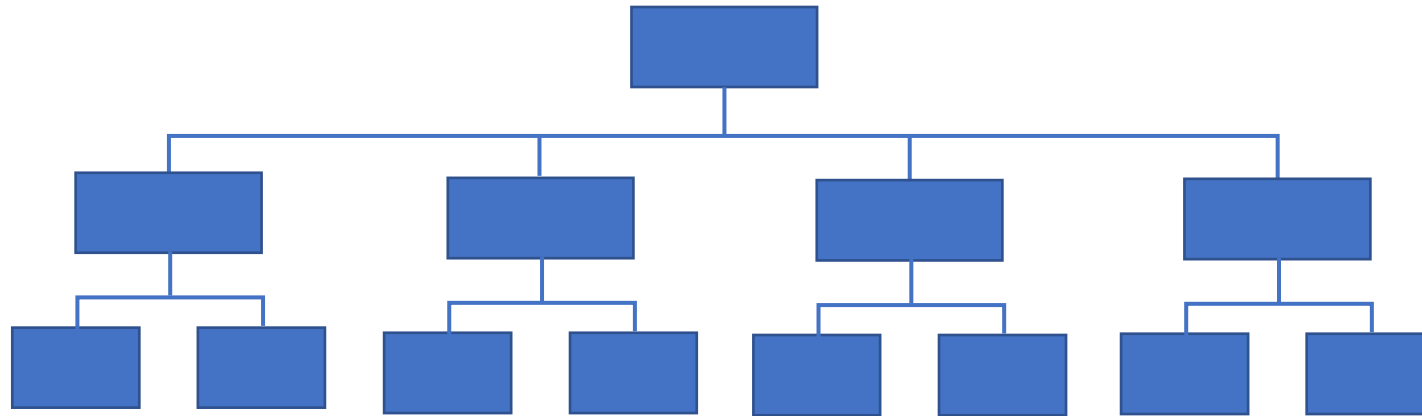
## EL DISEÑO DESCENDENTE

También denominado diseño **Top-Down**, o de **arriba hacia abajo**. Esta forma de razonar la solución de un problema forma parte de las Técnicas de Programación Modular.

El diseño descendente permite dividir el sistema en subsistemas y sus requerimientos.

El diseño descendente es compatible con la manera de pensar sobre los sistemas en general.

Cuando el analista de sistemas emplea un enfoque descendente, esta pensando acerca de las interdependencias de los subsistemas, tal como es en la organización existente.



## **Ventajas**

- Evitar el caos originado al tratar de diseñar el sistema "en un solo paso".
- Posibilidad de contar con grupos trabajando por separado pero simultáneamente en subsistemas independientes, pero necesarios.
- Ahorrar una gran cantidad de tiempo.
- Asegurar la calidad total del sistema.

## **Inconvenientes**

- Riesgo de que el sistema se divida en subsistemas "incorrectos".
- Se debe prestar atención a la necesidad de la superposición y la distribución de los recursos, es importante que cada subsistema se integre de manera correcta al sistema.
- Una vez que se realizan las divisiones en subsistemas, sus interfaces pueden descuidarse o simplemente ignorarse. La responsabilidad para lograr la adecuada interrelación debe quedar bien detallada.

El enfoque descendente proporciona al grupo de sistemas una clara división establecida de los usuarios en fuerzas de tareas para los subsistemas.

Los grupos asignados a las tareas establecidos por esta manera pueden tener una función doble, tal y como los círculos de control de calidad de los sistemas informáticos para la administración.

# EL TEOREMA FUNDAMENTAL DE LA PROGRAMACIÓN ESTRUCTURADA

**La técnica de la programación modular posee dos ventajas fundamentales:**

- Parcializar el estudio de la solución del problema, resolviendo el tema que plantea un determinado módulo.
- Trabajar en equipo.

Pero el correr de los tiempos exige que se avance rápidamente; es así que en la década de los 90 se habla de la programación orientada a objetos-

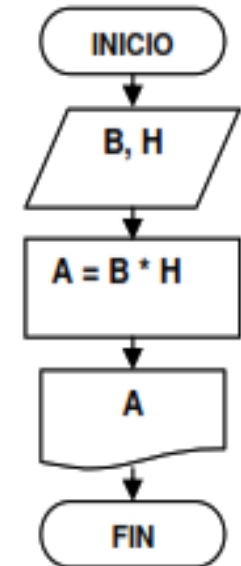
Definiremos los siguientes conceptos, dentro de las Técnicas de Programación Estructurada:

## PROGRAMA

Es un conjunto ordenado de instrucciones que permiten transformar los datos de entrada en datos de salida.

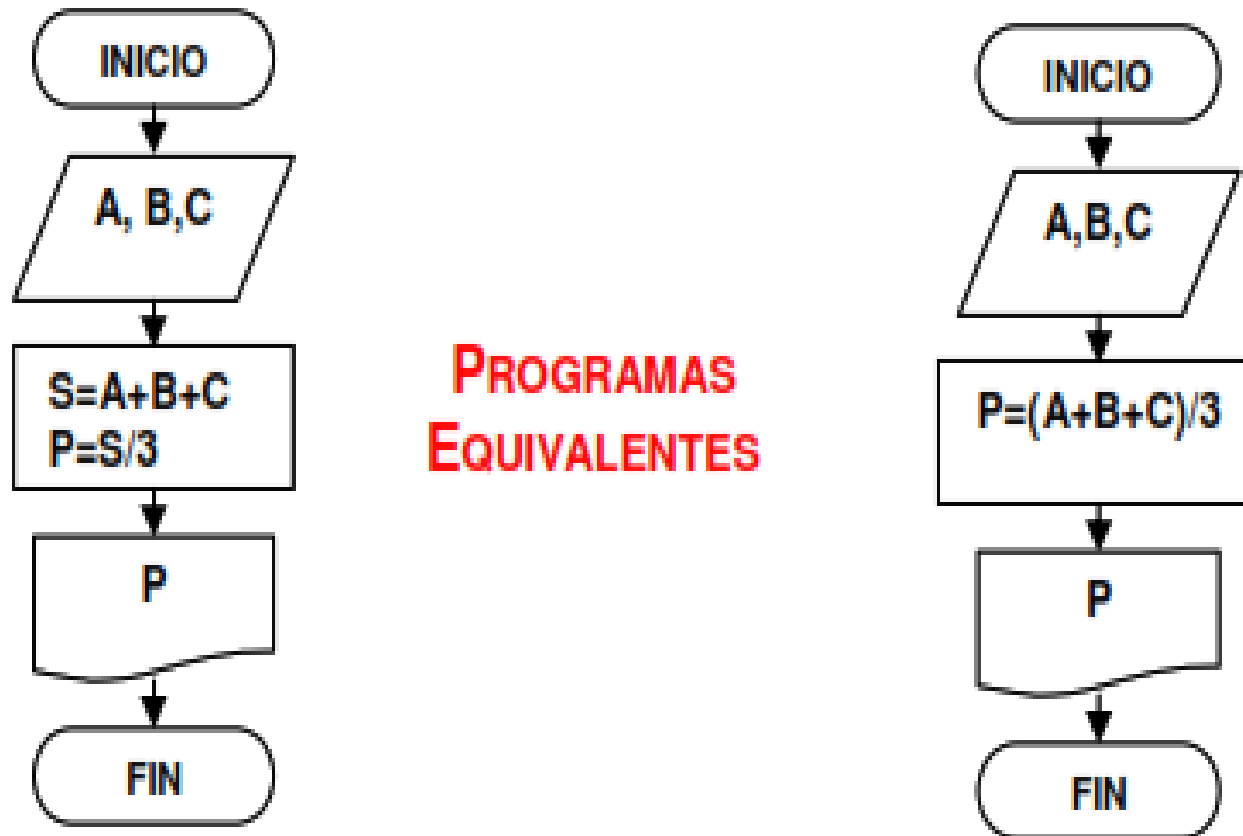
## PROGRAMA PROPIO

Es aquel que posee un solo punto de entrada y uno de salida para controlar el programa.

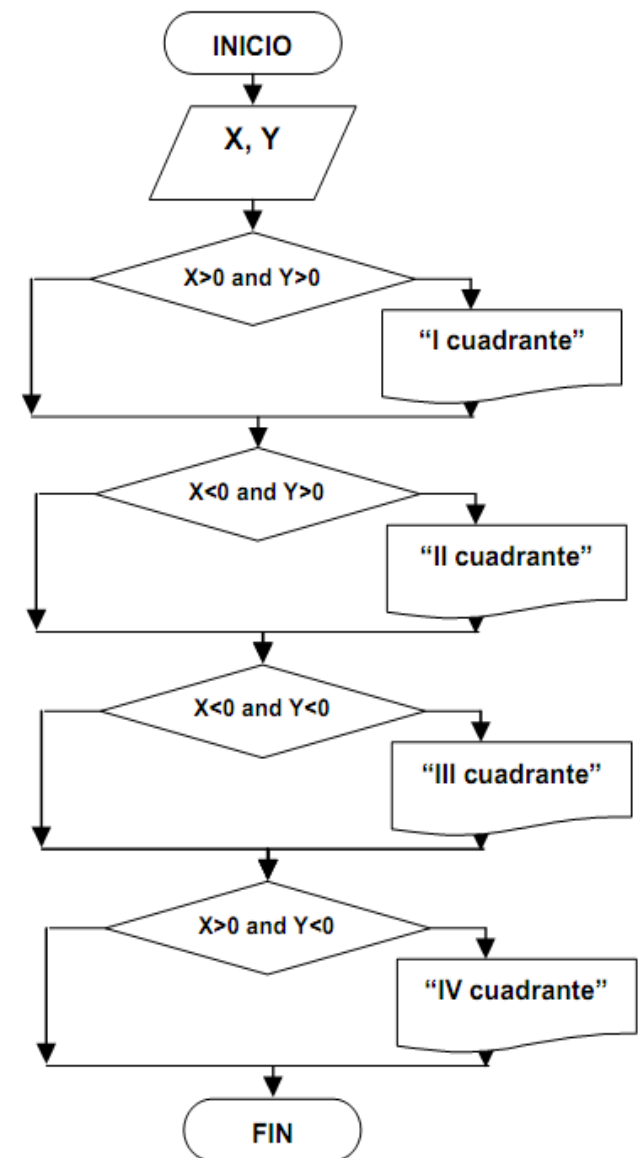
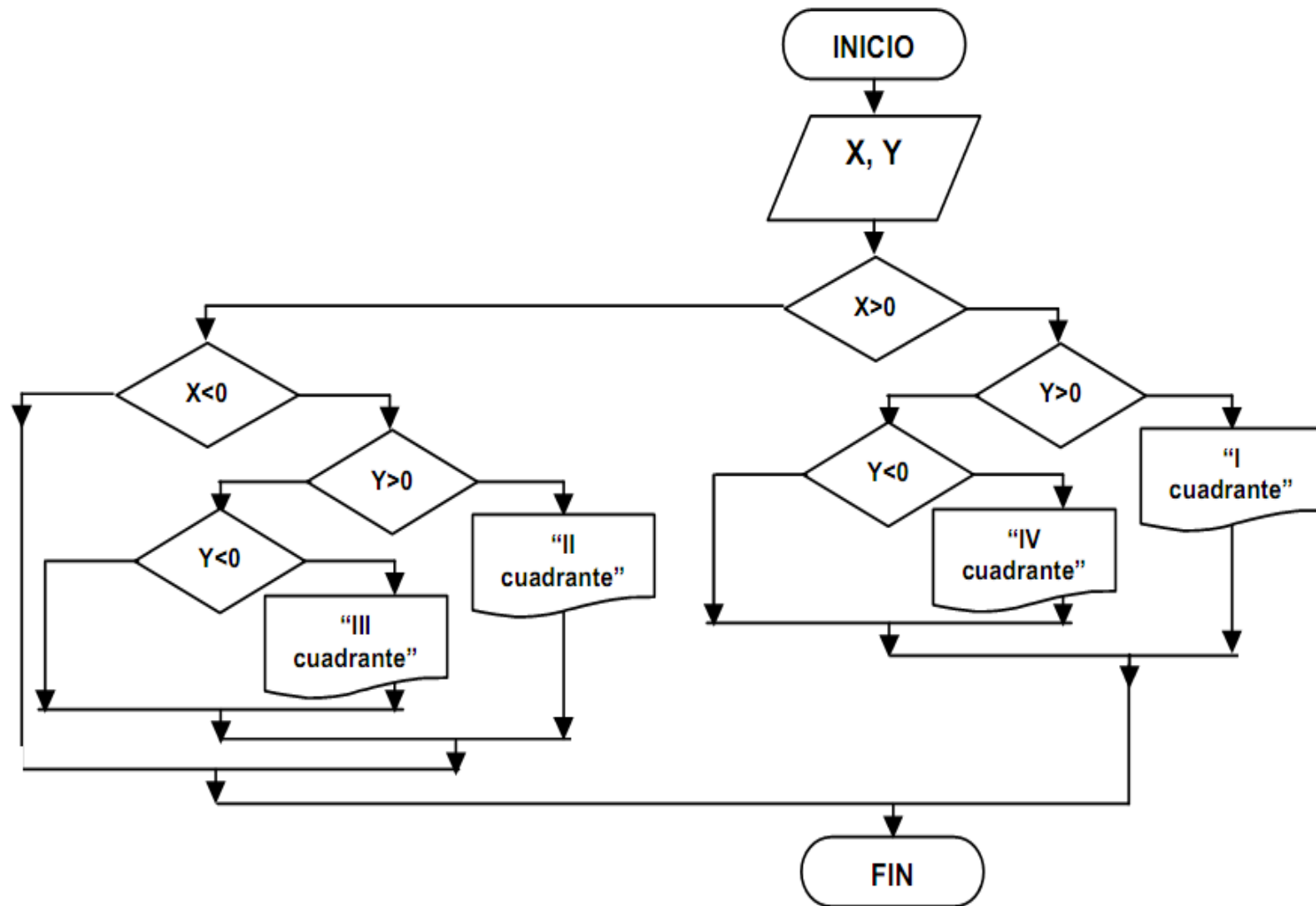


## PROGRAMA EQUIVALENTE

Son aquellos programas que producen iguales transformaciones de datos, es decir que para las mismas entradas producen idénticas salidas.



## PROGRAMA EQUIVALENTE



## TEOREMA FUNDAMENTAL DE LA PROGRAMACIÓN ESTRUCTURADA

*“Todo programa propio puede ser sustituido por un programa equivalente que tenga únicamente como estructuras lógicas las siguientes estructuras básicas:*

- 1. Secuencial,*
- 2. Condicional, de Decisión o Selección,*
- 3. Repetición o Iteración”*

*La Filosofía de la Programación Estructurada se basa en 3 pautas fundamentales:*

1. Estructuras Básicas.
2. Recursos abstractos o razonamiento.
3. Razonamiento Top-Down o Diseño Descendente.

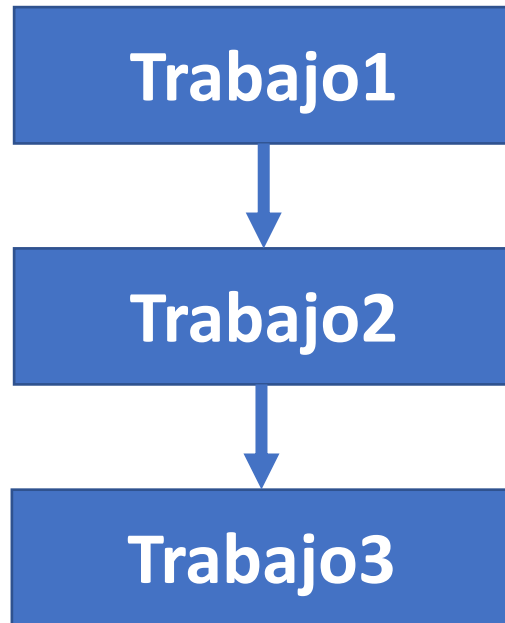
# TIPOS DE ESTRUCTURAS

- SECUENCIAL
- DE SELECCIÓN Y
- REPETICIÓN



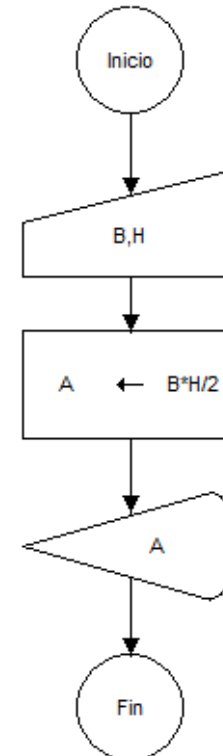
## ESTRUCTURA SECUENCIAL

Representa una secuencia simple de **problemas de evaluación**, se representa esquemáticamente usando bloques convencionales de procesos de datos.



### Ejemplo 1:

Introduzca la altura y la base de un triángulo, encuentre y muestre el área del mismo.

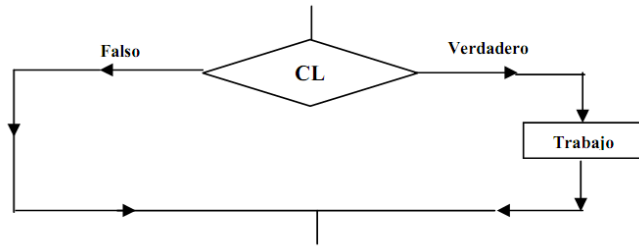


# ESTRUCTURA DE SELECCIÓN

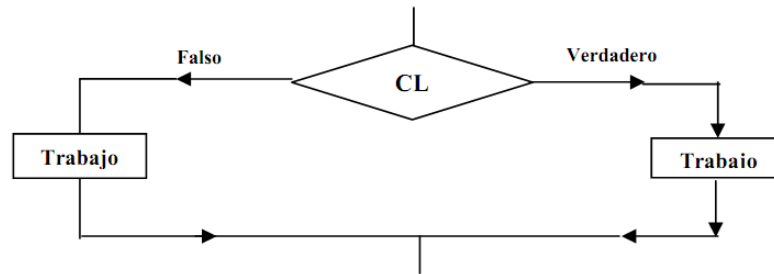
Su uso es evidente en los casos en los cuales se necesita bifurcar la solución por dos caminos alternativos de acuerdo a alguna **condición lógica**.

Las estructuras de selección pueden ser:

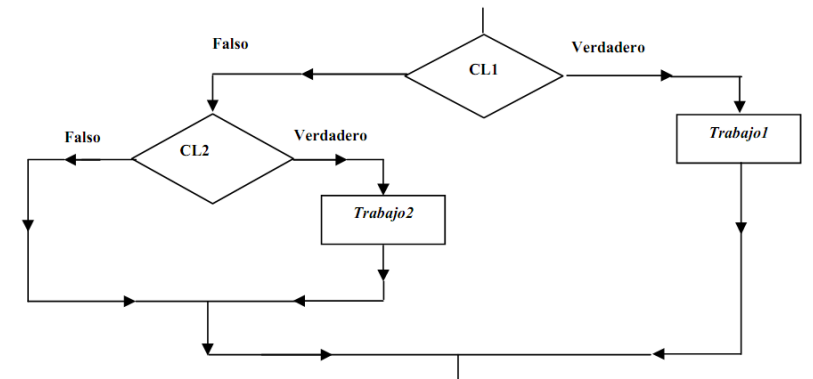
## 1. Selección Simple



## 2. Selección Doble

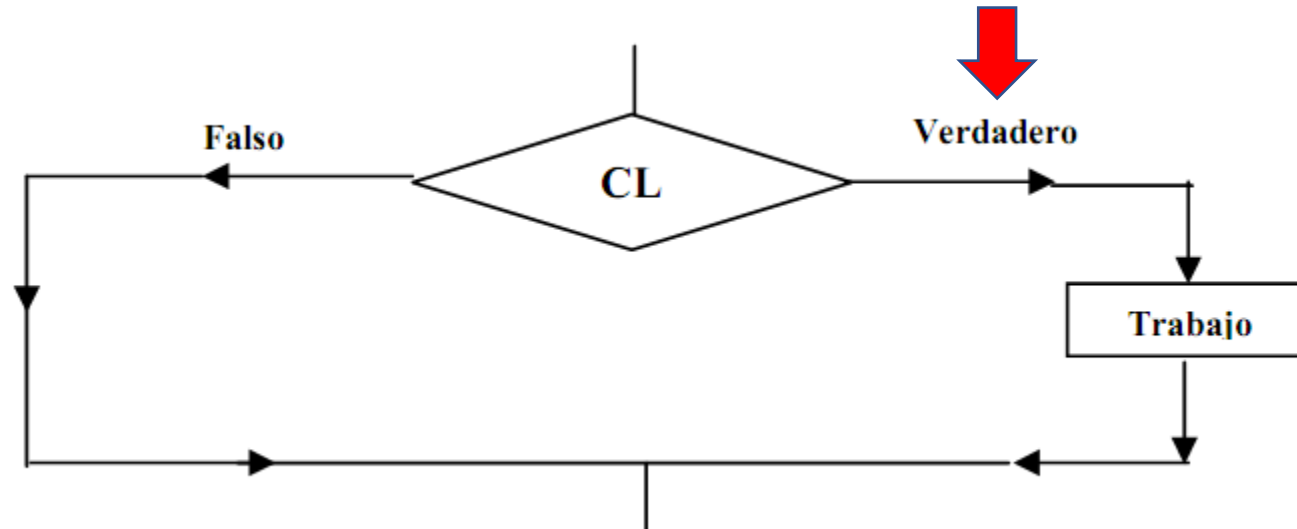


## 3. Selección Múltiple



## Estructura de Selección Simple

Esta estructura presenta trabajo por una sola rama, generalmente por la rama cuyo resultado de la condición lógica es **verdadero**.



## Ejemplo 2:

Introduzca un número entero e indique si el mismo es positivo o no.

### Resultado

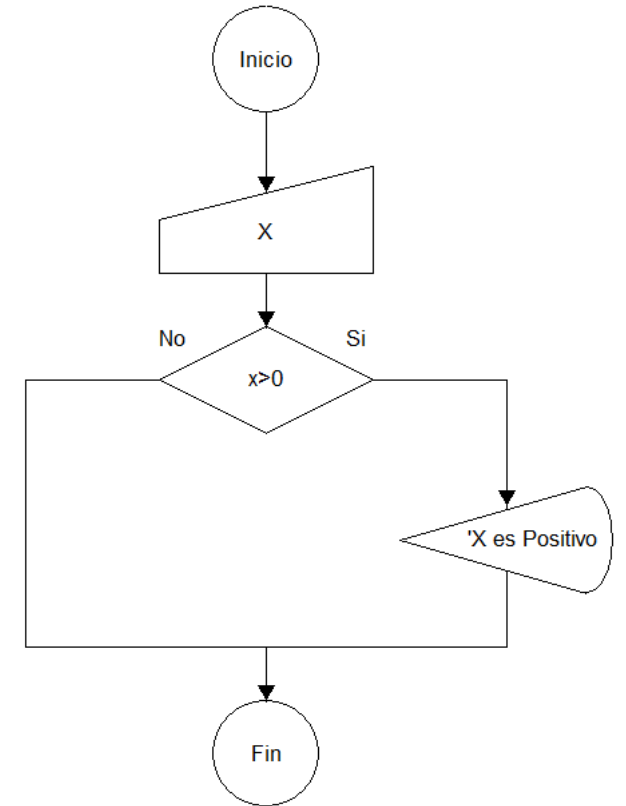
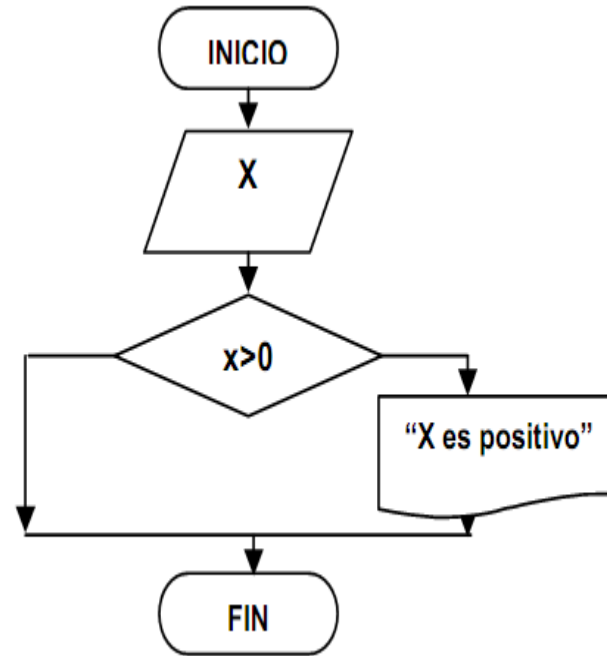
Mensaje: "X es positivo"

### Datos

X: identificador que indica el valor del número entero ingresado.

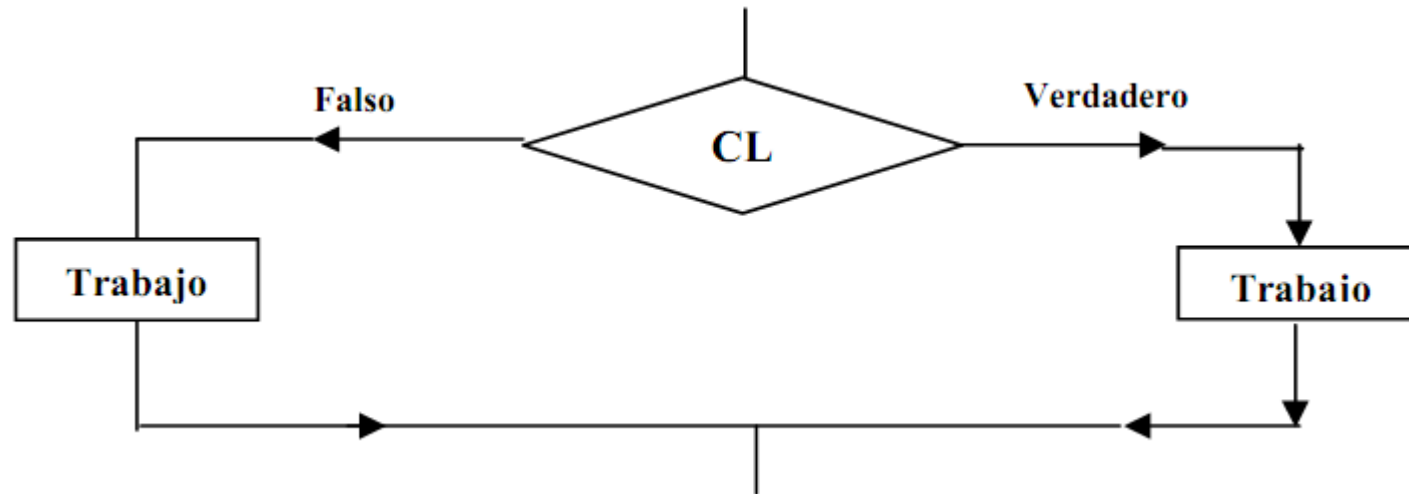
### Condiciones Vinculantes

Si  $(X > 0)$  entonces "**X es positivo**"



## ESTRUCTURA DE SELECCIÓN DOBLE

Esta estructura presenta trabajo por ***ambas ramas***. Según sea la condición se hará el proceso correspondiente.



### Ejemplo 3:

Introduzca un número entero e indique si el mismo es positivo o no lo es.

#### Resultado

Mensajes:

“X es Positivo”

“X NO es Positivo”

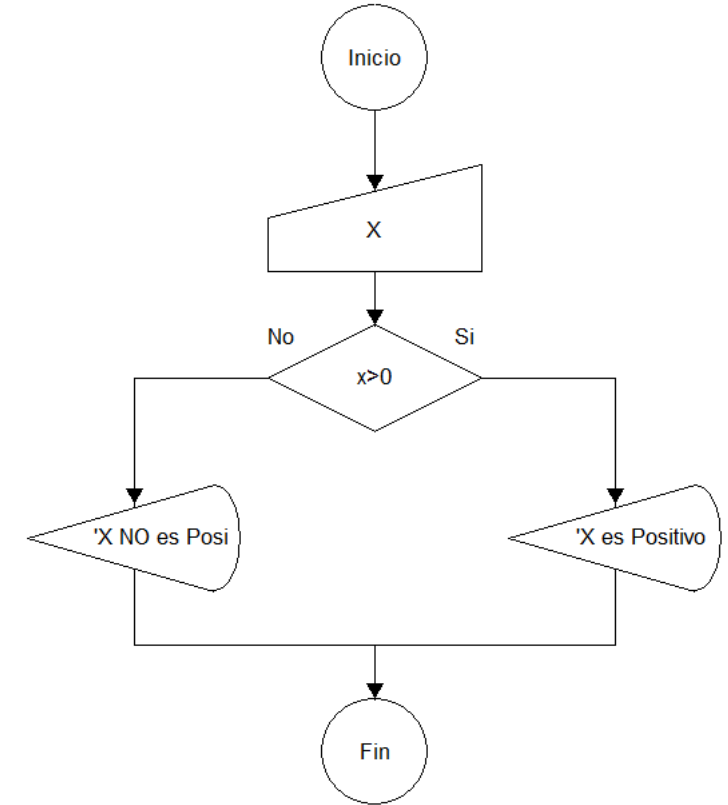
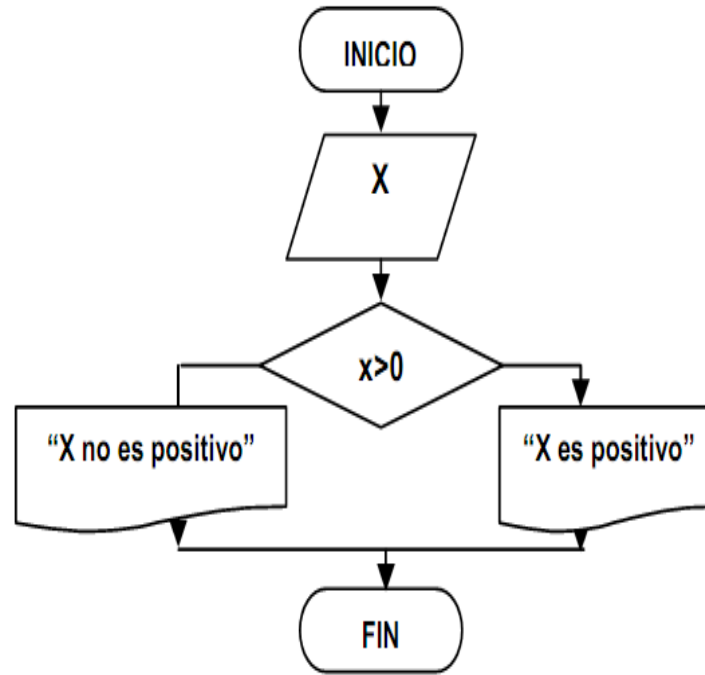
#### Datos

X: identificador que indica el valor del número entero ingresado.

#### Condiciones Vinculantes

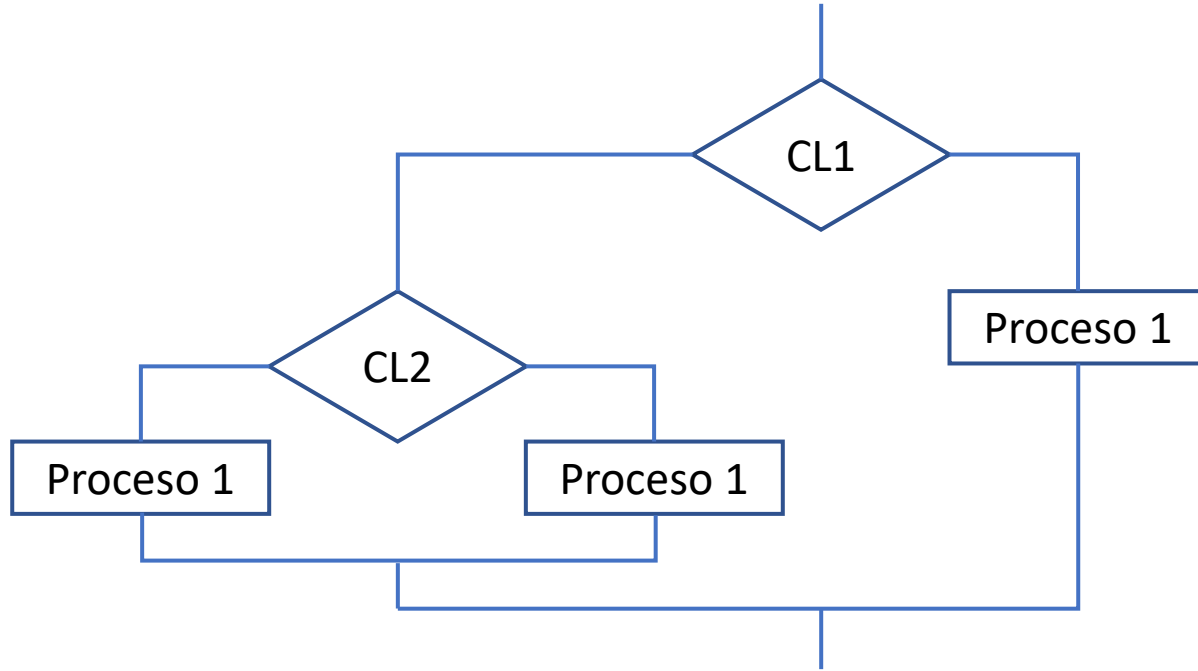
Si  $(X > 0)$  entonces “X es positivo”

caso contrario “X NO es Positivo”



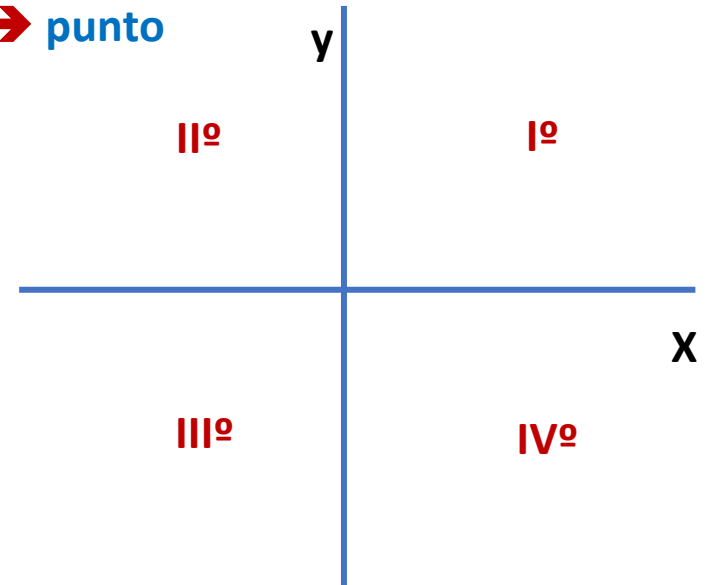
# ESTRUCTURA DE SELECCIÓN MÚLTIPLE

La estructura de selección múltiple presenta **condiciones lógicas anidadas**, es decir que por una o por varias ramas existen también otras condiciones lógicas. Este es un ejemplo de una estructura de selección múltiple.



**Ejemplo4:** Introduzca las coordenadas de un punto en el plano, e indique a qué cuadrante pertenece dicho punto.

X,Y → punto



## Condiciones Vinculantes

Si  $X > 0 \Rightarrow$  si  $Y > 0 \Rightarrow$  "I cuadrante"

sino  $Y < 0 \Rightarrow$  "IV cuadrante"

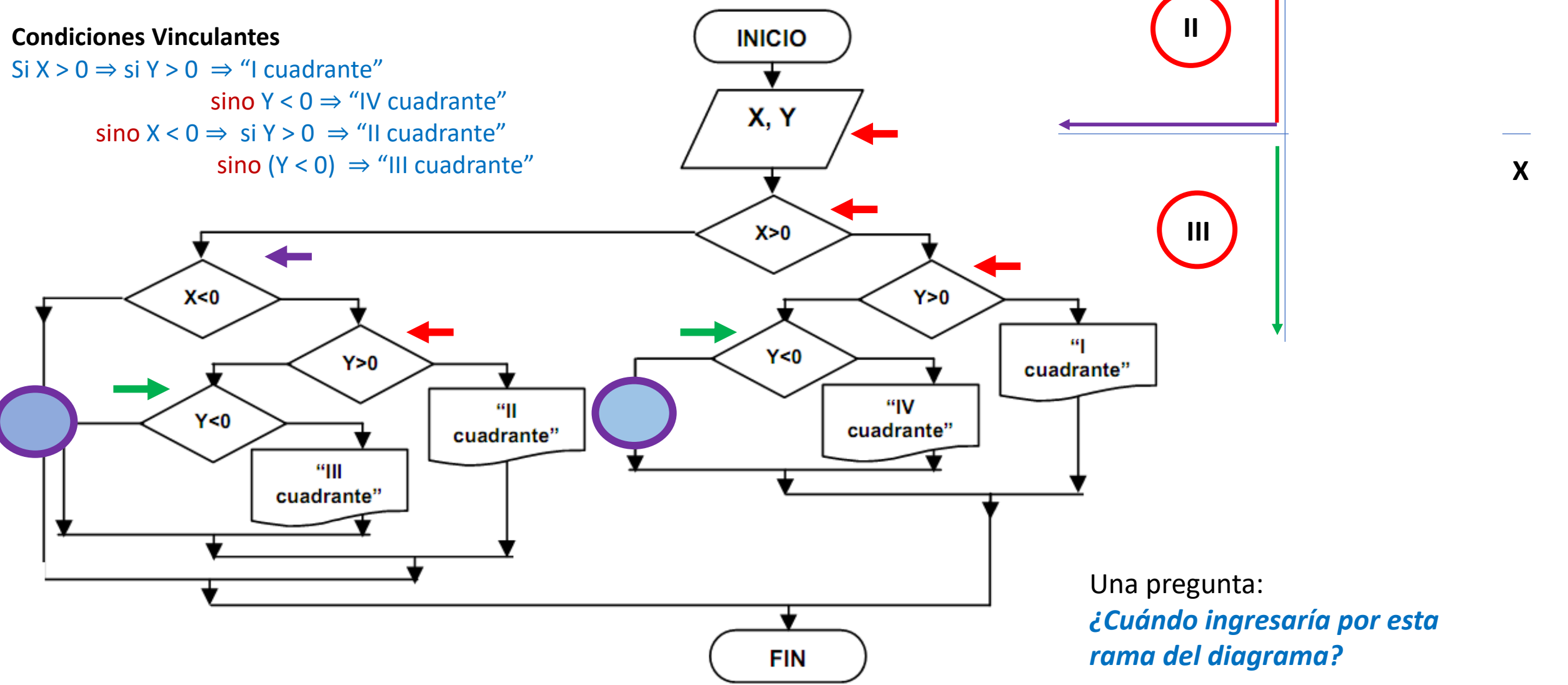
sino  $X < 0 \Rightarrow$  si  $Y > 0 \Rightarrow$  "II cuadrante"

sino  $(Y < 0) \Rightarrow$  "III cuadrante"

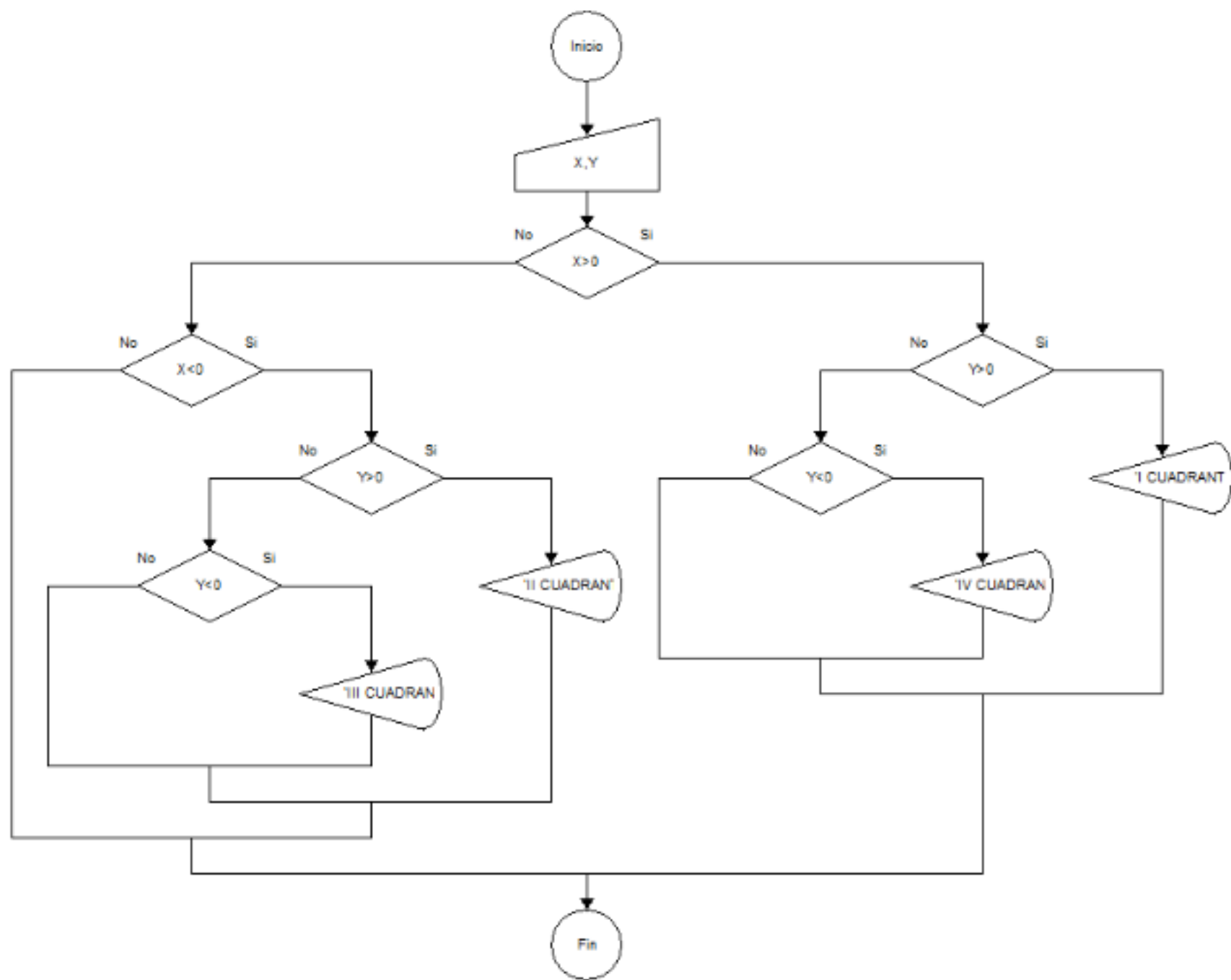
**Ejemplo4:** Introduzca las coordenadas de un punto en el plano, e indique a qué cuadrante pertenece dicho punto.

**Condiciones Vinculantes**

Si  $X > 0 \Rightarrow$  si  $Y > 0 \Rightarrow$  "I cuadrante"  
sino  $Y < 0 \Rightarrow$  "IV cuadrante"  
sino  $X < 0 \Rightarrow$  si  $Y > 0 \Rightarrow$  "II cuadrante"  
sino  $(Y < 0) \Rightarrow$  "III cuadrante"







Toda estructura de selección múltiple que se abre a lo ancho puede ser representada por una estructura más simple pero cuyas condiciones lógicas serán **expresiones compuestas**.

### **Resultados**

Mensajes:

“I cuadrante”

“II cuadrante”

“III cuadrante”

“IV cuadrante”

### **Datos**

**X:** identificador que indica el valor de la abscisas del número ingresado.

**Y:** identificador que indica el valor de las ordenadas del número ingresado.

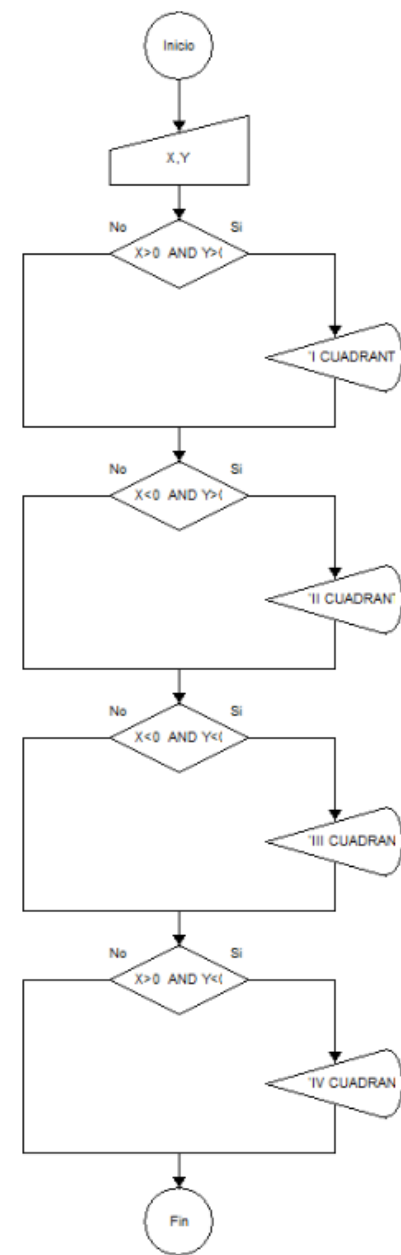
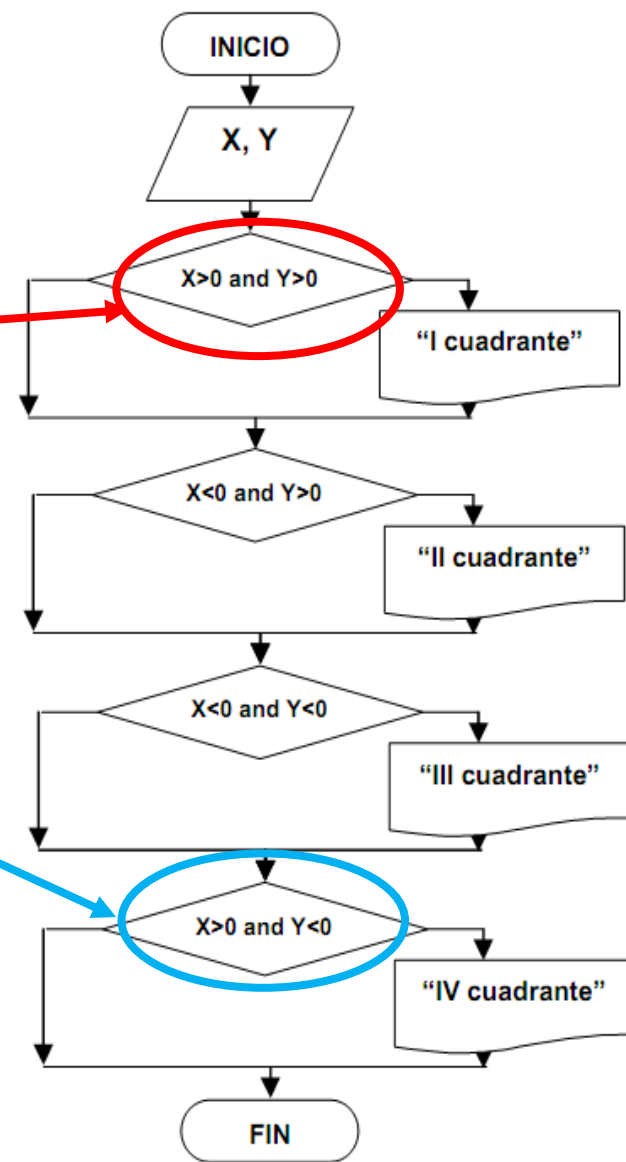
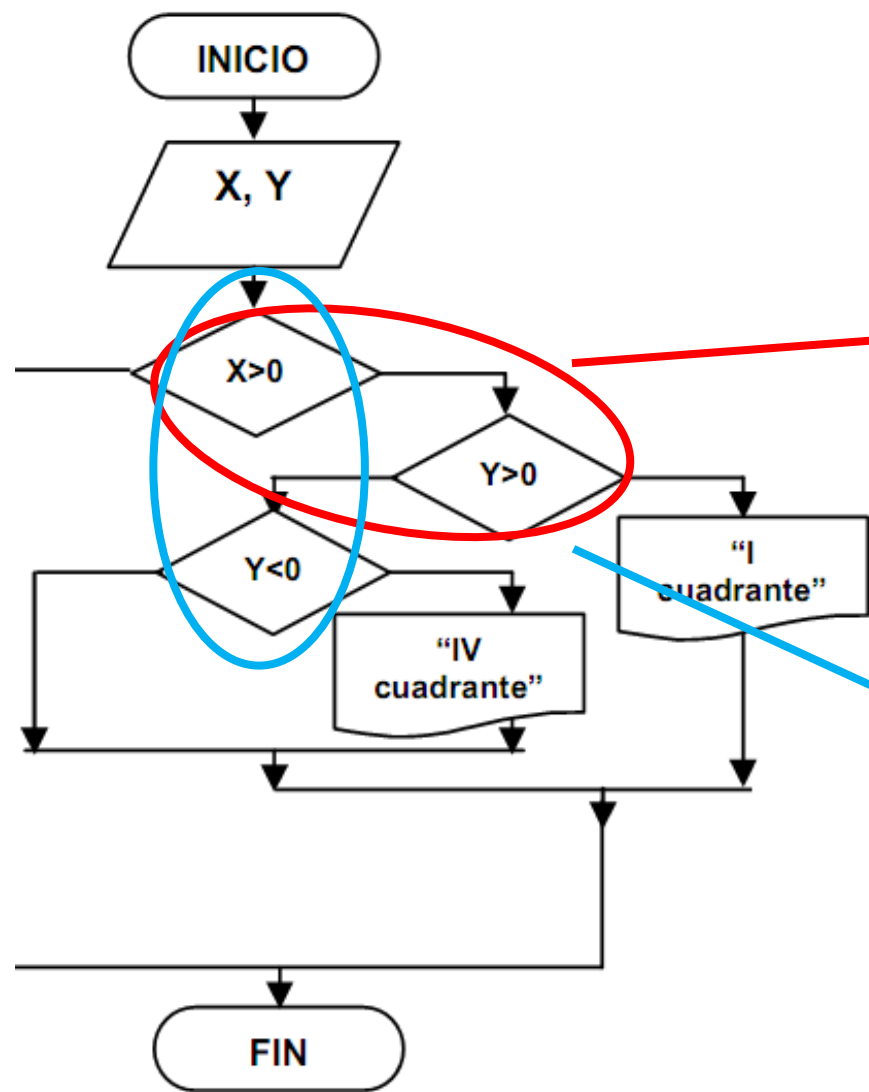
### **Condiciones Vinculantes**

$(X > 0) \text{ and } (Y > 0) \Rightarrow \text{“I cuadrante”}$

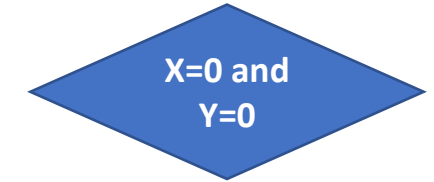
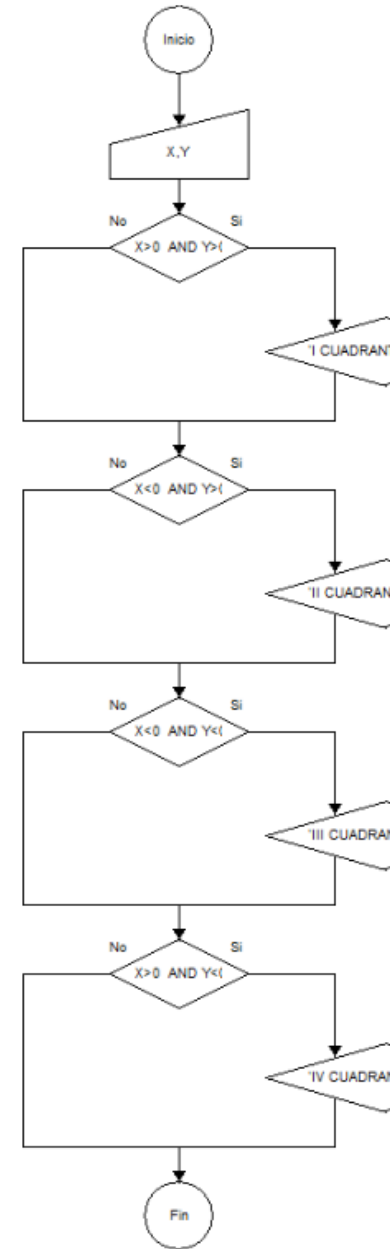
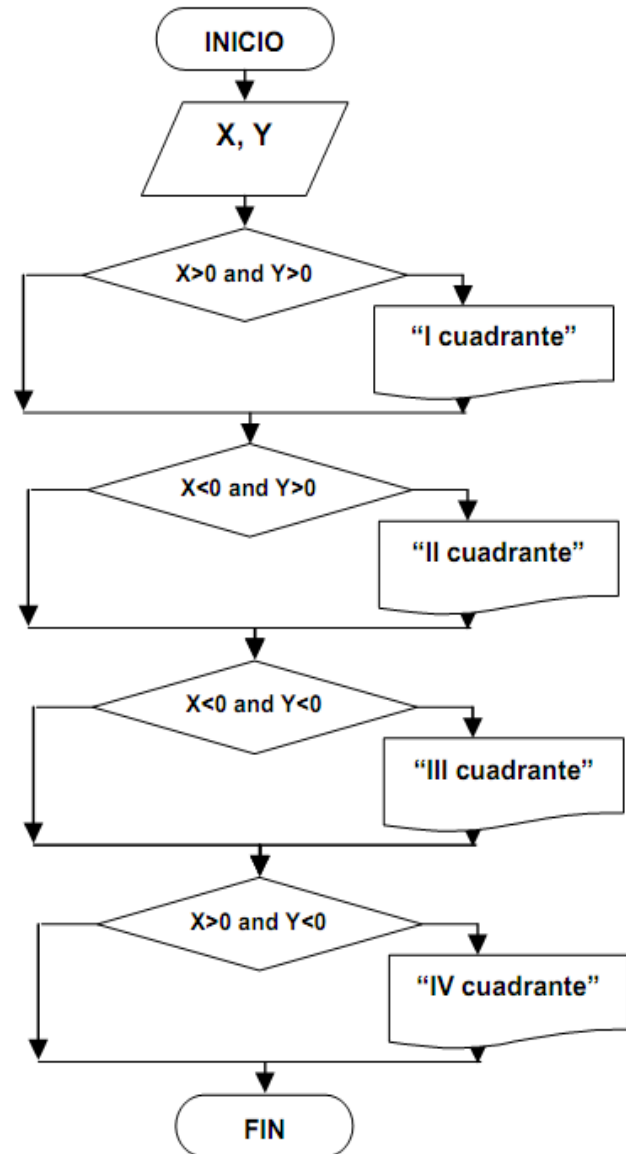
$(X < 0) \text{ and } (Y > 0) \Rightarrow \text{“II cuadrante”}$

$(X < 0) \text{ and } (Y < 0) \Rightarrow \text{“III cuadrante”}$

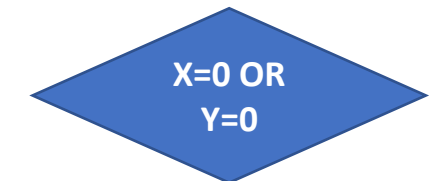
$(X > 0) \text{ and } (Y < 0) \Rightarrow \text{“IV cuadrante”}$



Y en que parte pondrían la condición para saber si el punto está sobre algunos de los ejes cartesianos?



Esta en  
el centro



Esta sobre  
algún eje

### Ejercicio N° 1:

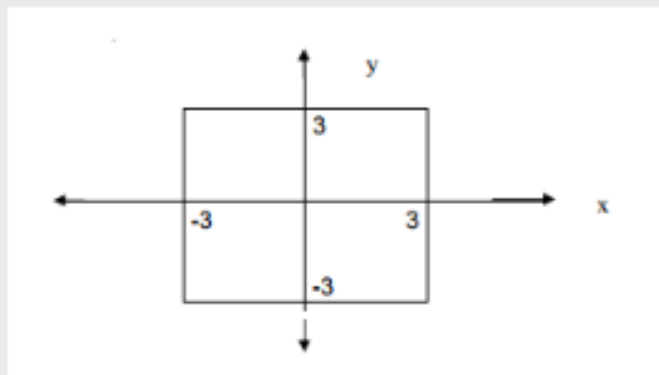
Ingresar el valor del Kw y el consumo eléctrico en una casa, se pide:

- Mostrar un mensaje "No tiene consumo" si el consumo es igual a cero.
- Mostrar lo que debe pagar el usuario, teniendo en cuenta que se le realiza un descuento del 5% para consumos menores a 350 Kw.

### Ejercicio N° 2

Ingresar las coordenadas de un punto en el plano, dos variables X, Y.

Indicar si dicho punto si está dentro o fuera del cuadrado.



## EJERCICIO PARA PRACTICAR

Encontrar los valores de la raíces de la ecuación de segundo grado, ingresando los coeficientes **a**, **b**, **c** como datos. Considere todos los casos posibles, y mostrar también un mensaje indicando si son reales iguales, reales distintas, o complejas.

**SE PIDE:**

(a) Analizar las partes del problema,

(b) Realizar el diagrama de flujo.

$$ax^2 + bx + c = 0$$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

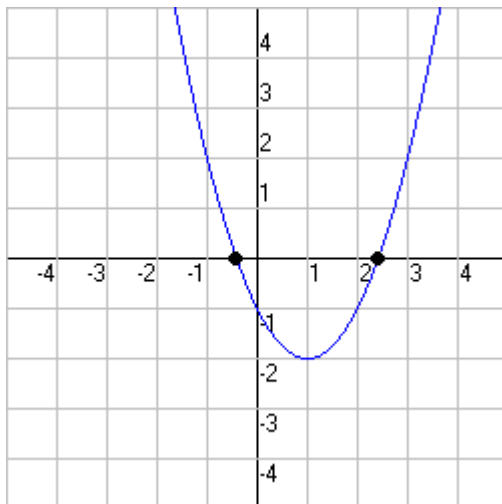
$$x1 = -b + \sqrt{b^2 - 4ac} / 2a$$

$$x2 = -b - \sqrt{b^2 - 4ac} / 2a$$

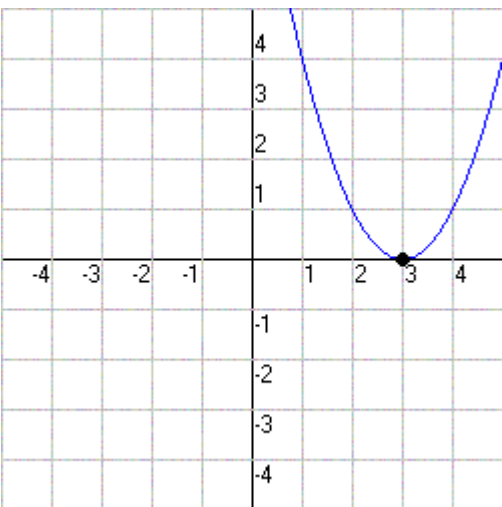
Si el discriminante  **$b^2 - 4ac$**  es negativo, entonces no hay soluciones reales de la ecuación.

Si el discriminante es cero, hay únicamente una solución.

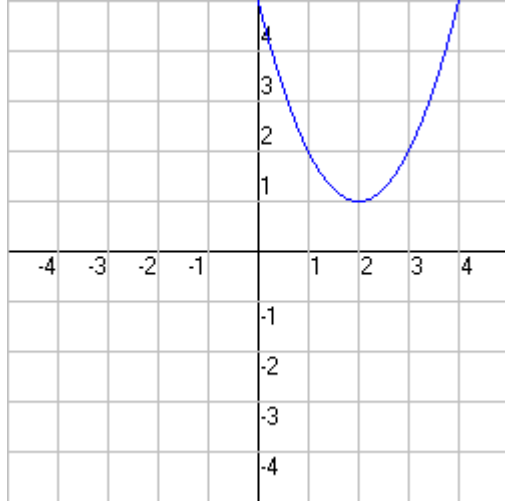
Si el discriminante es positivo, entonces el símbolo  $\pm$  significa que obtiene dos respuestas.



**Parábola con dos intersecciones en  $x$   
(discriminante positivo)**



**Parábola con una intersección en  $x$   
(discriminante cero)**



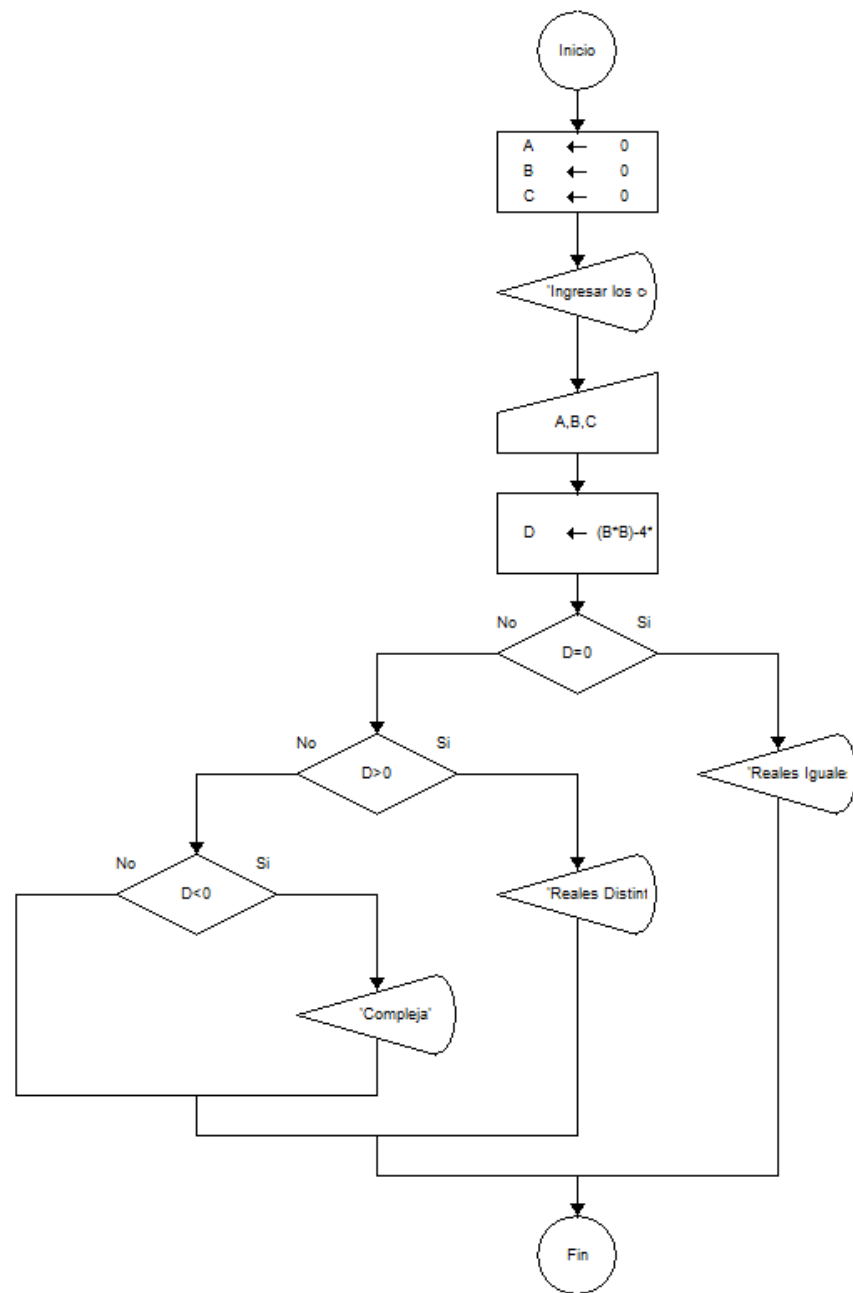
**Parábola sin intersección en  $x$   
(discriminante negativo)**

**EJEMPLO 1:**  $x^2 - x - 12 = 0$

**EJEMPLO 2:**  $3x^2 + 2x + 1 = 0$

**EJEMPLO 3:**  $x^2 - 6x + 9 = 0$





**FIN DE LA CLASE**

**SIGUIENTE TEMA...**

**ESTRUCTURA DE  
REPETICIÓN O ITERACIÓN**