



# Algoritmos

Diseño Estructurado

# ¿Qué es un algoritmo?

- La forma de expresar la resolución del problema en la fase previa a la codificación
- Proceso:
  1. Especificación del problema
  2. Escribir el algoritmo
  3. Codificación del programa

# Tipos de algoritmos

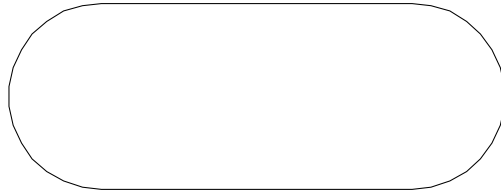


- Diagramas de flujo
- Pseudocódigo

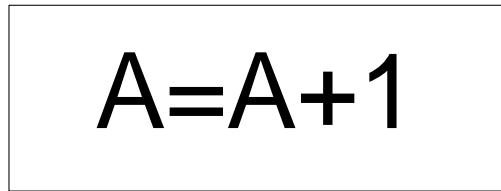
# Diagramas de flujo

- Expresa el diseño del programa de forma gráfica
- Se puede expresar el problema a diferentes niveles de refinamiento o detalle, en función de la concreción que se desee aplicar :
  - ▣ Bloques
  - ▣ Instrucciones

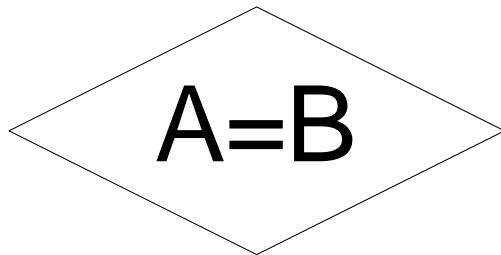
# Símbolos



Inicio / Fin



Instrucción



Condicional



Entrada por  
teclado

# Símbolos



Importe

Salida en  
pantalla



Impresión

Salida por  
impresora



Datos

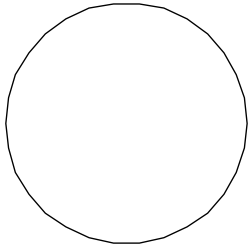
Entrada / Salida  
genérica



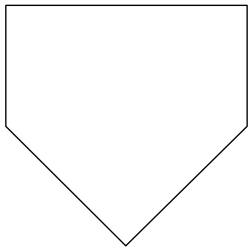
Procedi  
miento

Ejecución de  
procedimiento

# Símbolos

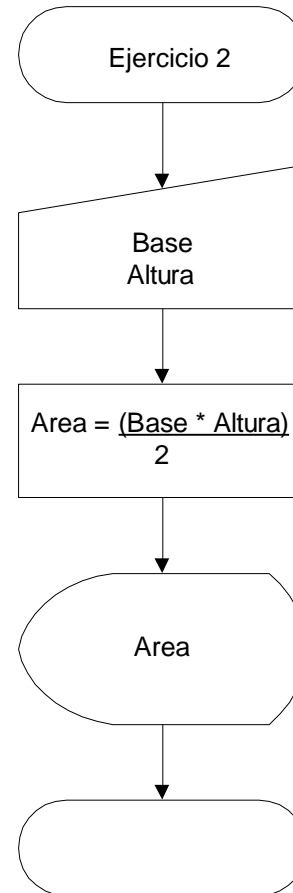
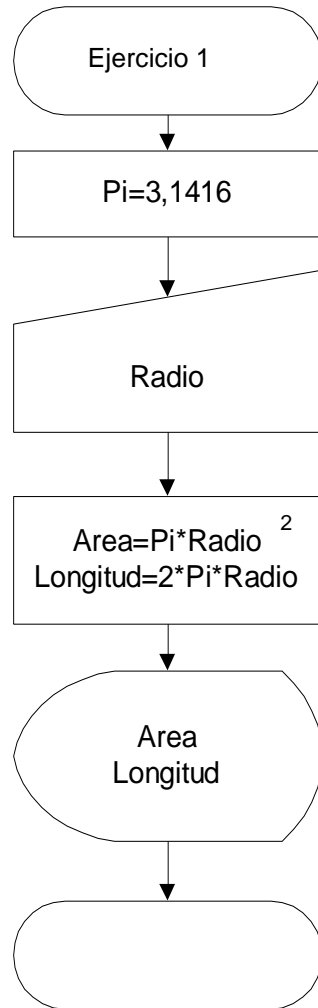


Conector dentro  
de página



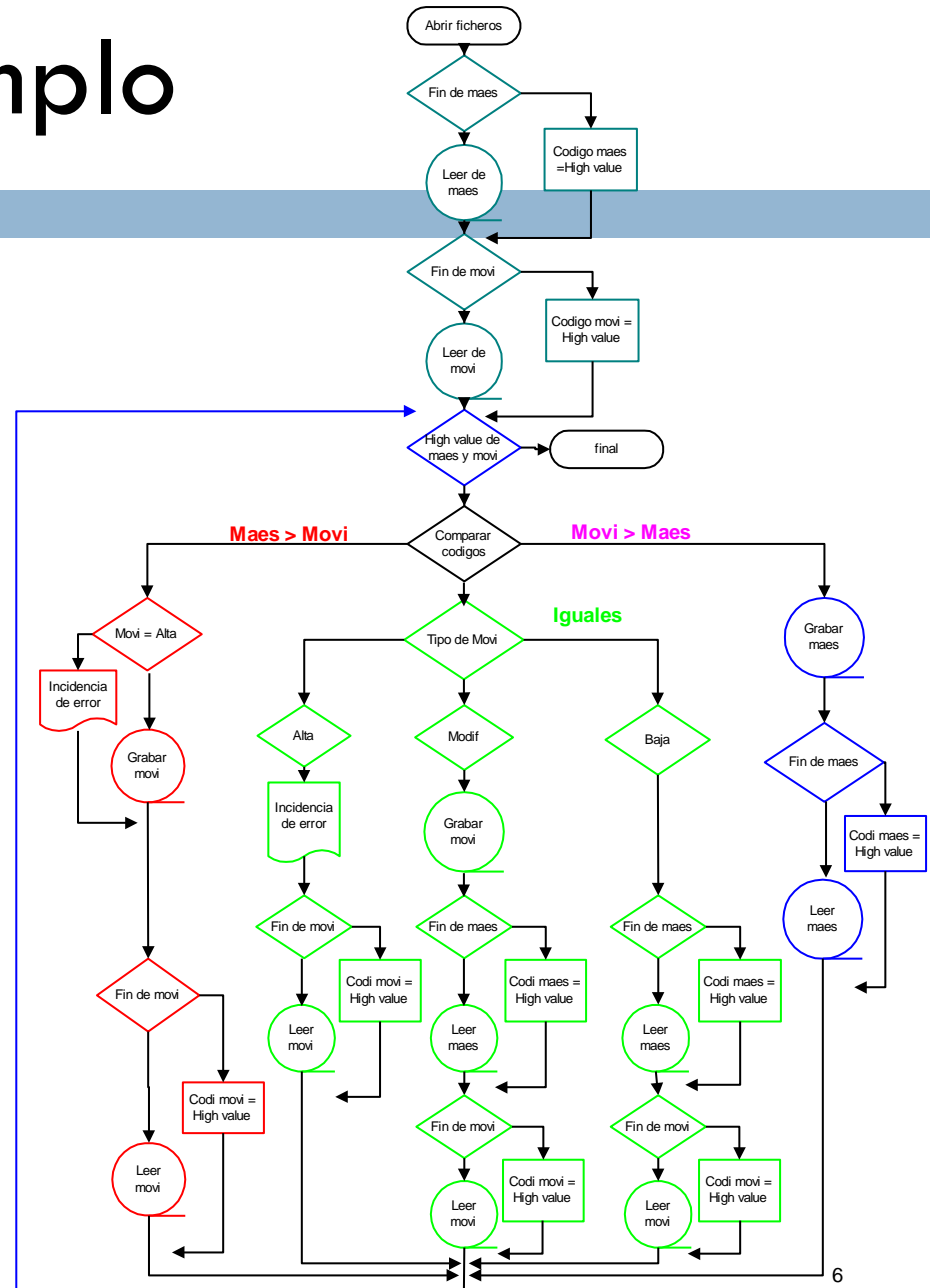
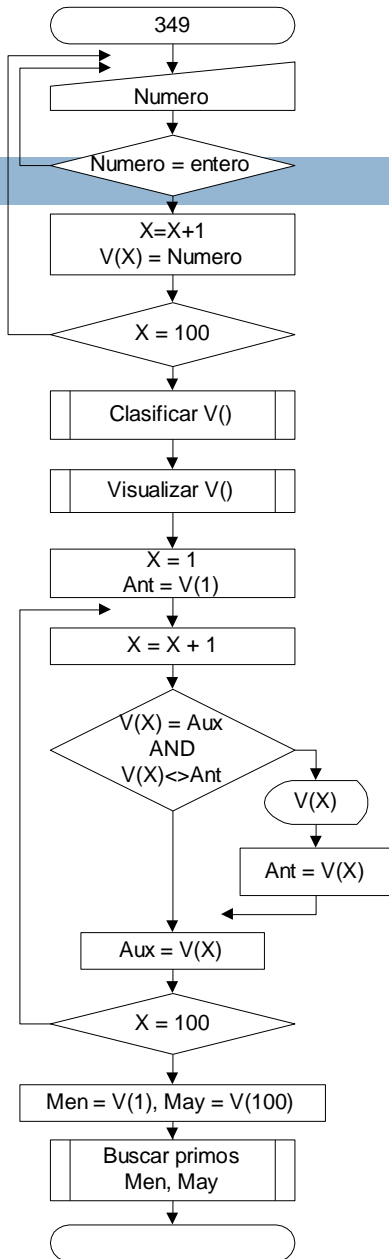
Conector fuera  
de página

# Ejemplo





# Ejemplo



# Diagrama de flujo

Algunas reglas:

- Siempre se desarrollan de arriba abajo y de izquierda a derecha
- Si no se especifica, las salidas en los símbolos de decisión son por la derecha cuando cumple y por la izquierda cuando no se cumple
- Nunca debe quedar una línea de ejecución sin una salida o solución

# Pseudocódigo

---

- Es la expresión escrita en lenguaje “humano” de un algoritmo

# Pseudocódigo

- La forma de escribir el Pseudocódigo seguirá las mismas reglas que las de un programa

Inicio

Variables

real base, altura

real area

FinVariables

Leer base

Leer altura

area=base\*altura/2

Escribir area

Fin

Inicio

Variables

entera x

Fin Variables

x=0

Mientras x<=10

Escribe x

x=x+1

FinMientras

Fin



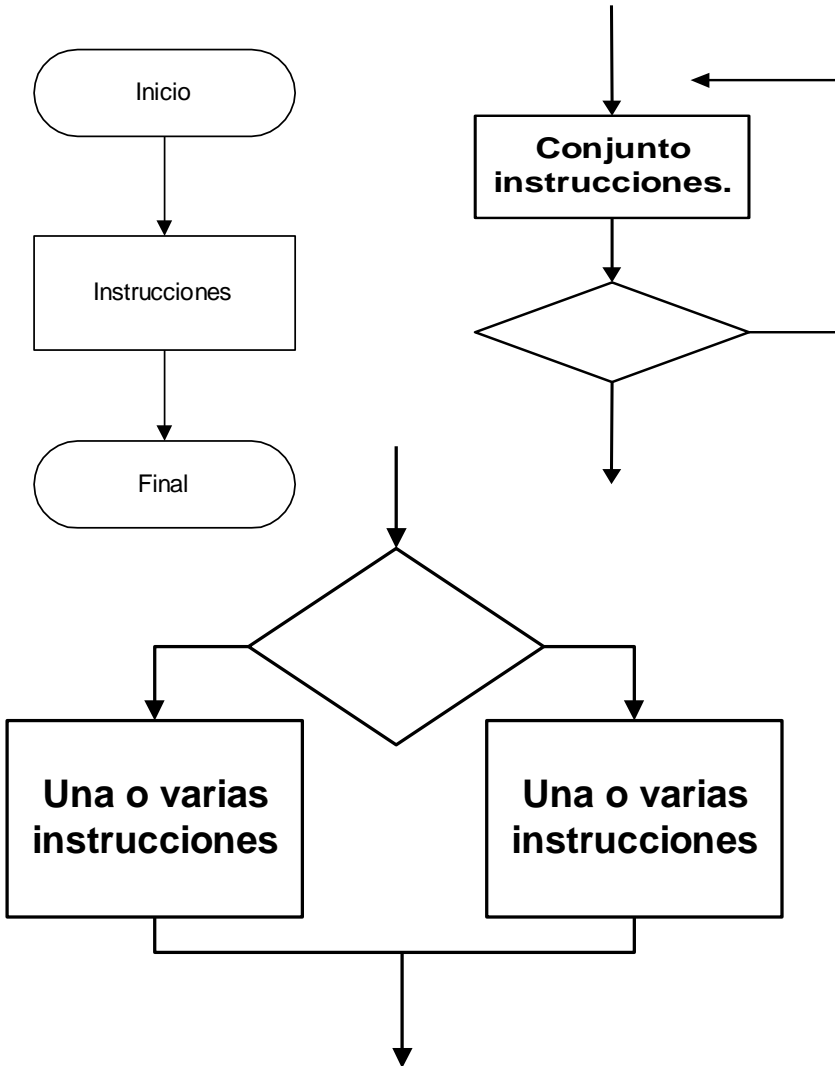
# Estructuras Básicas

Secuencial

Alternativa

Repetitiva

# Estructuras básicas

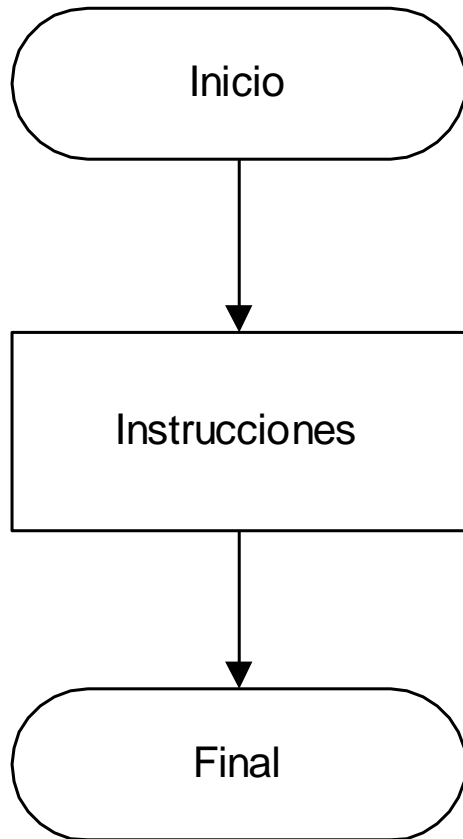


- Cualquier problema se puede resolver utilizando una o varias de las estructuras básicas que hay en programación
- Estas estructuras son tres
  - Secuenciales
  - Alternativas
  - Repetitivas



Secuencial

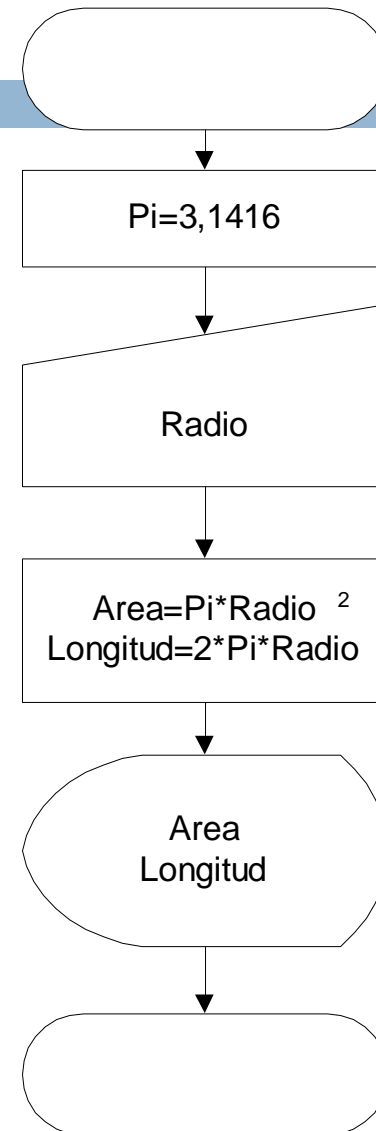
# Secuencial



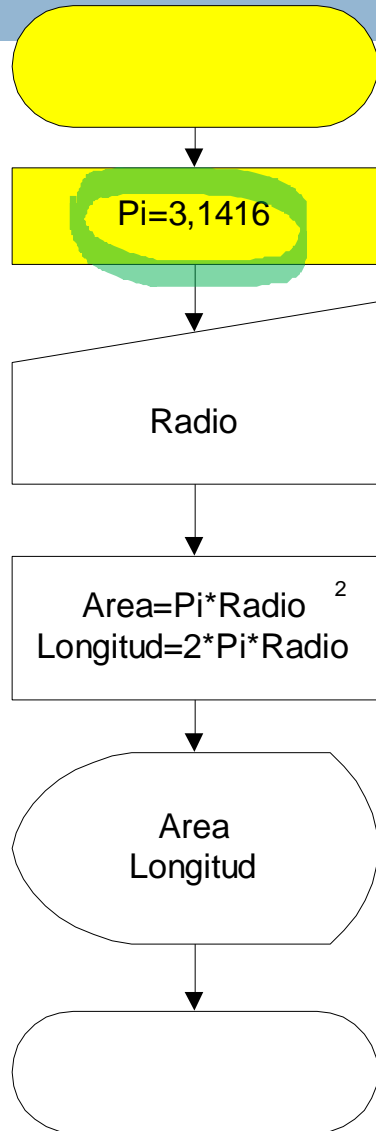
- La ejecución tiene un comienzo y un final
- Las instrucciones se ejecutan en el mismo orden en el que están escritas
- Puede haber una o muchas instrucciones desde que se inicia la ejecución hasta su final



# Secuencial. Un ejemplo

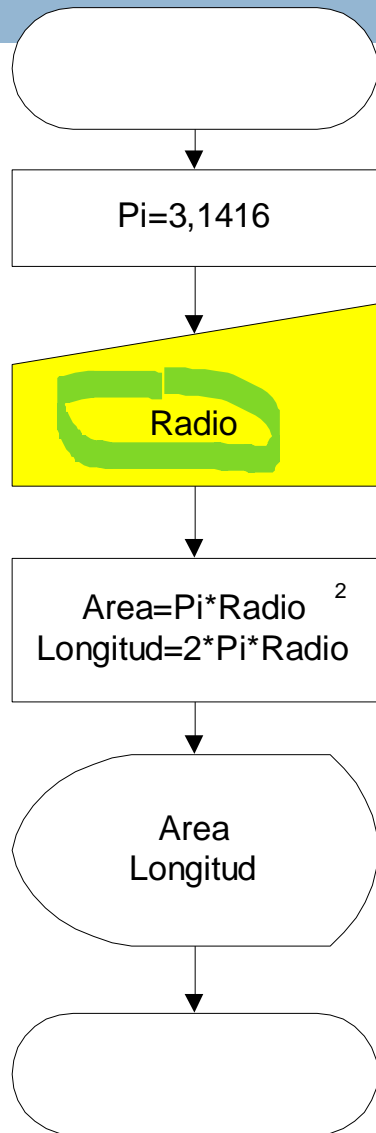


# Secuencial. Un ejemplo



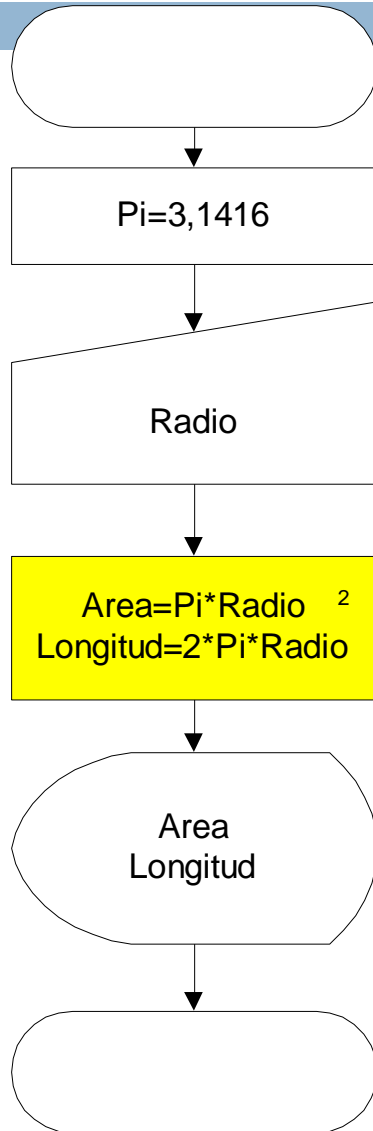
- Iniciamos el proceso y declaramos la constante Pi

# Secuencial. Un ejemplo



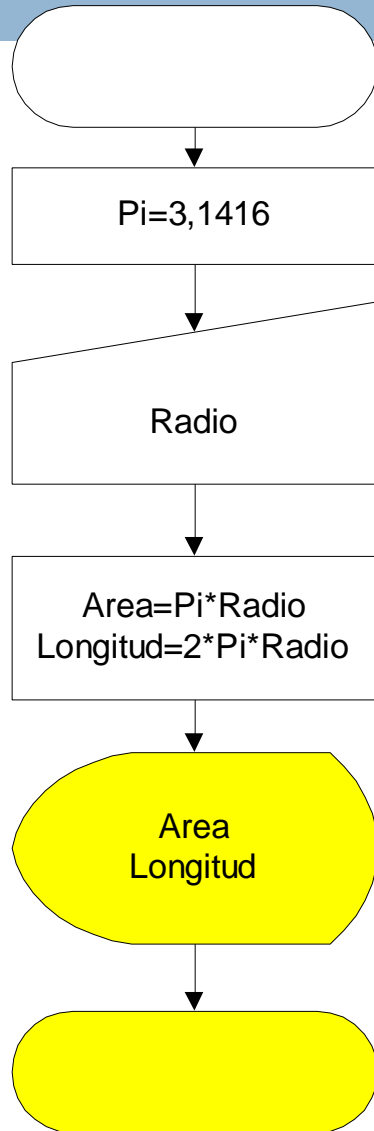
- Se solicita el radio de una circunferencia

# Secuencial. Un ejemplo



- Calculamos el área y la longitud de la circunferencia

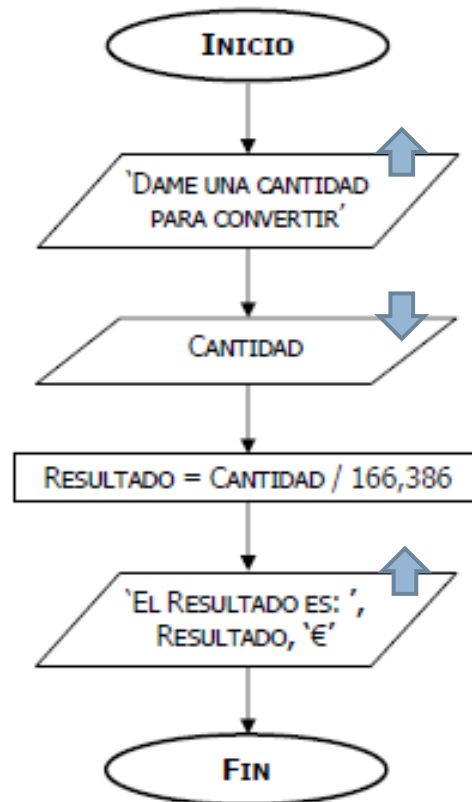
# Secuencial. Un ejemplo



- Visualizamos el resultado y finalizamos la ejecución del programa

# Estructura Secuencial. Ejemplo

Un ejemplo de Ordinograma sería el siguiente: Un algoritmo que calcule los euros correspondientes a una cantidad en pesetas dada se podría representar de la siguiente forma:

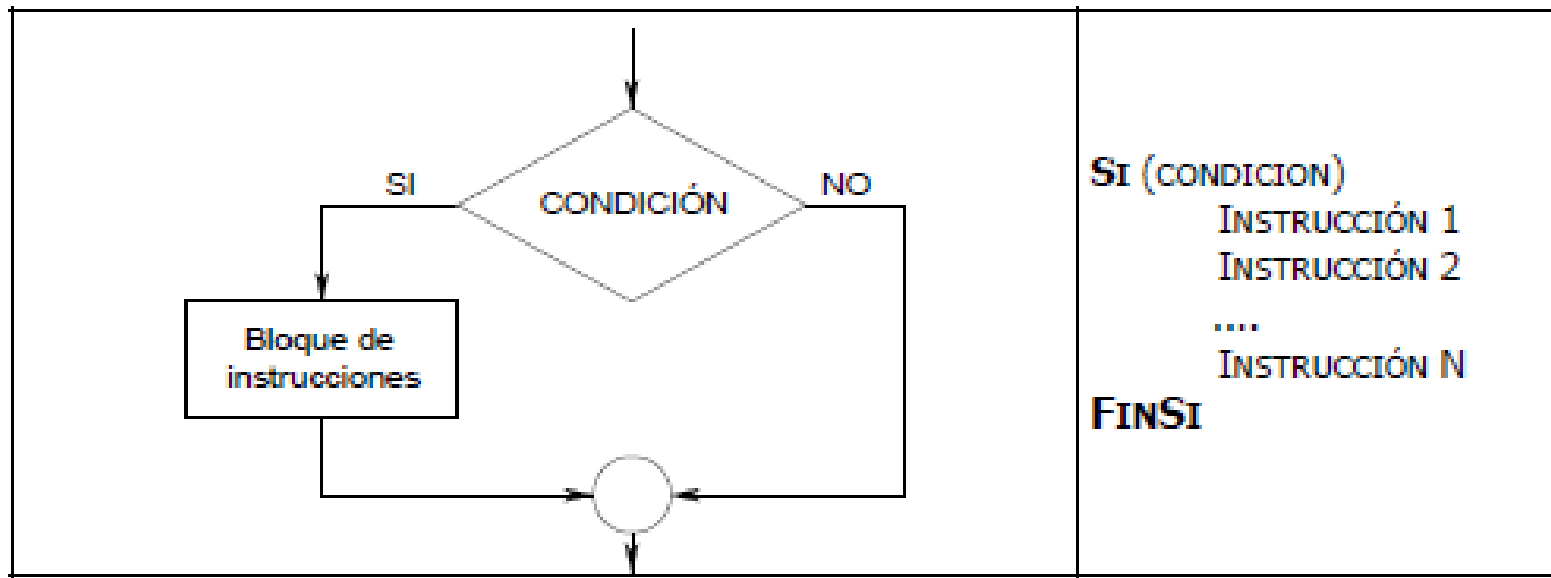




# Alternativa

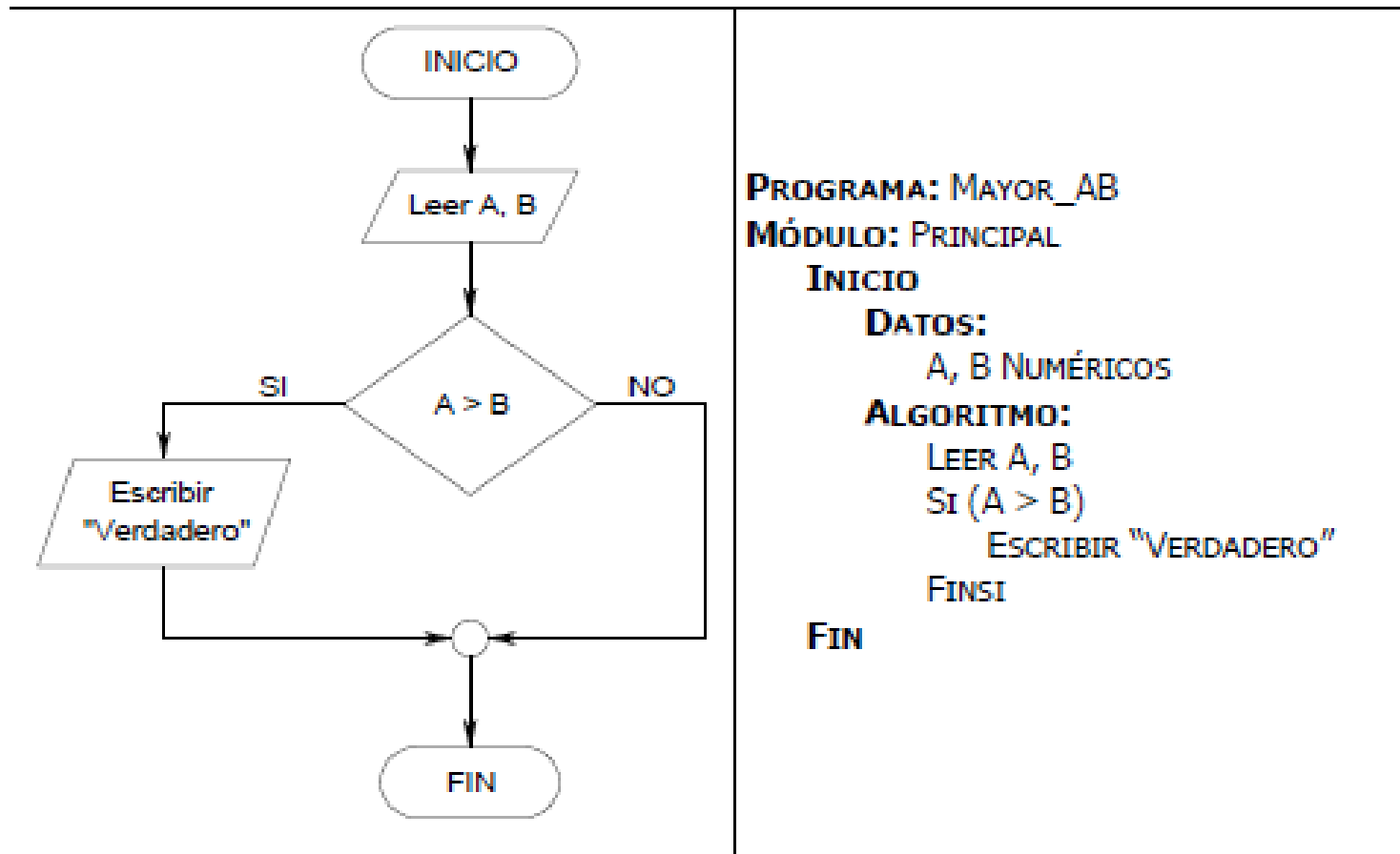
○ Condicional

# Alternativa simple

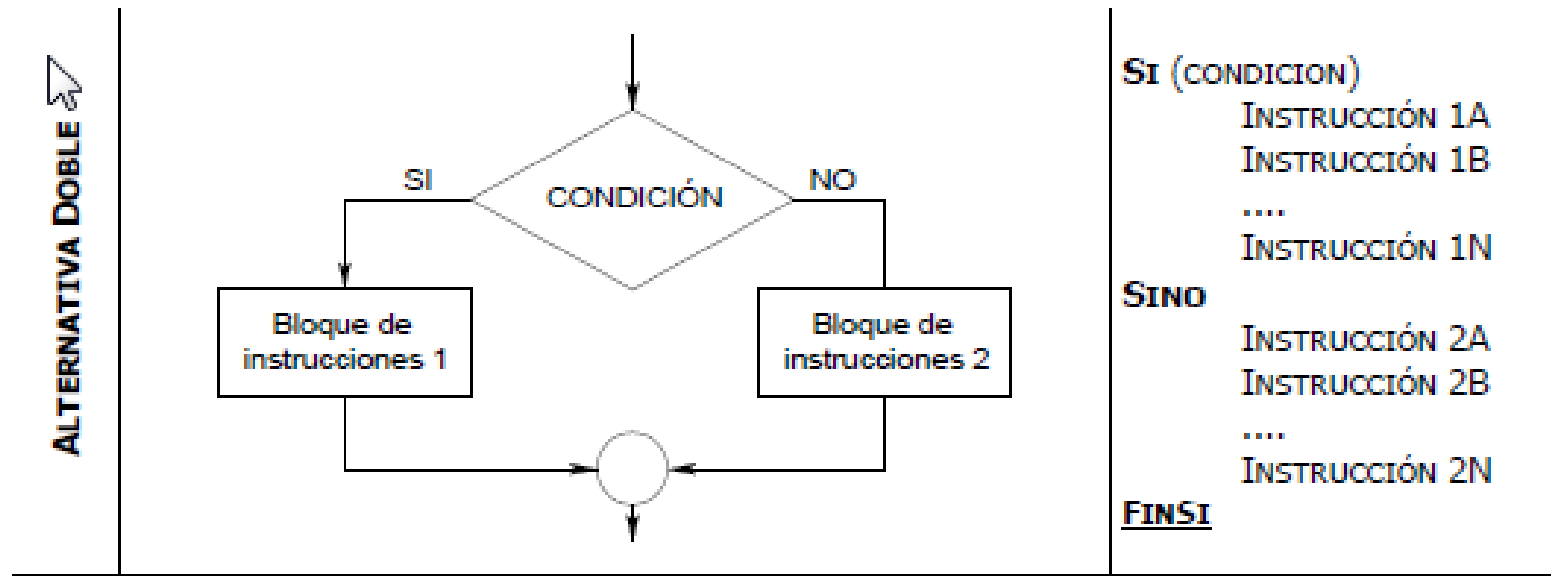




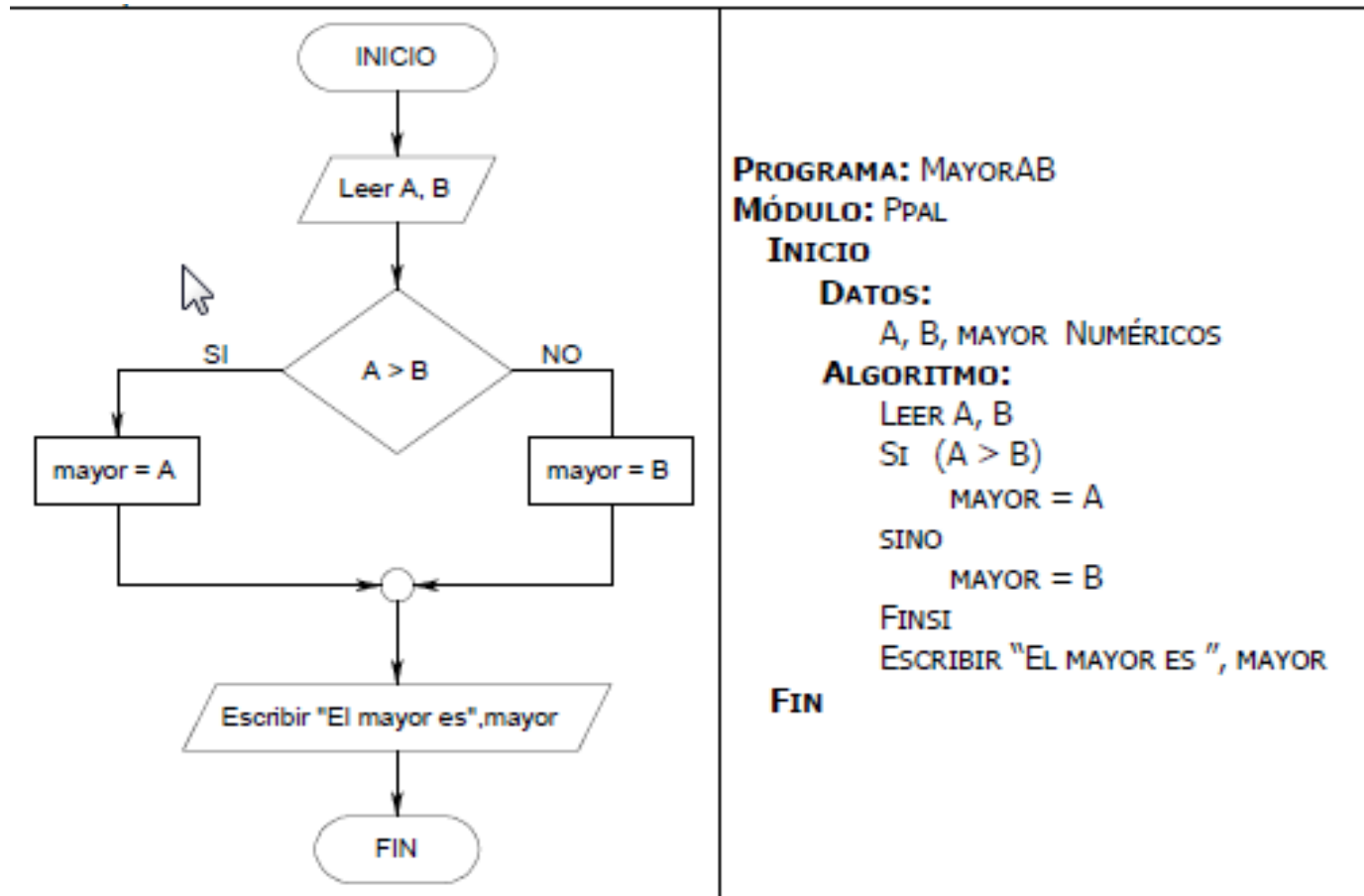
# Alternativa simple. Ejemplo



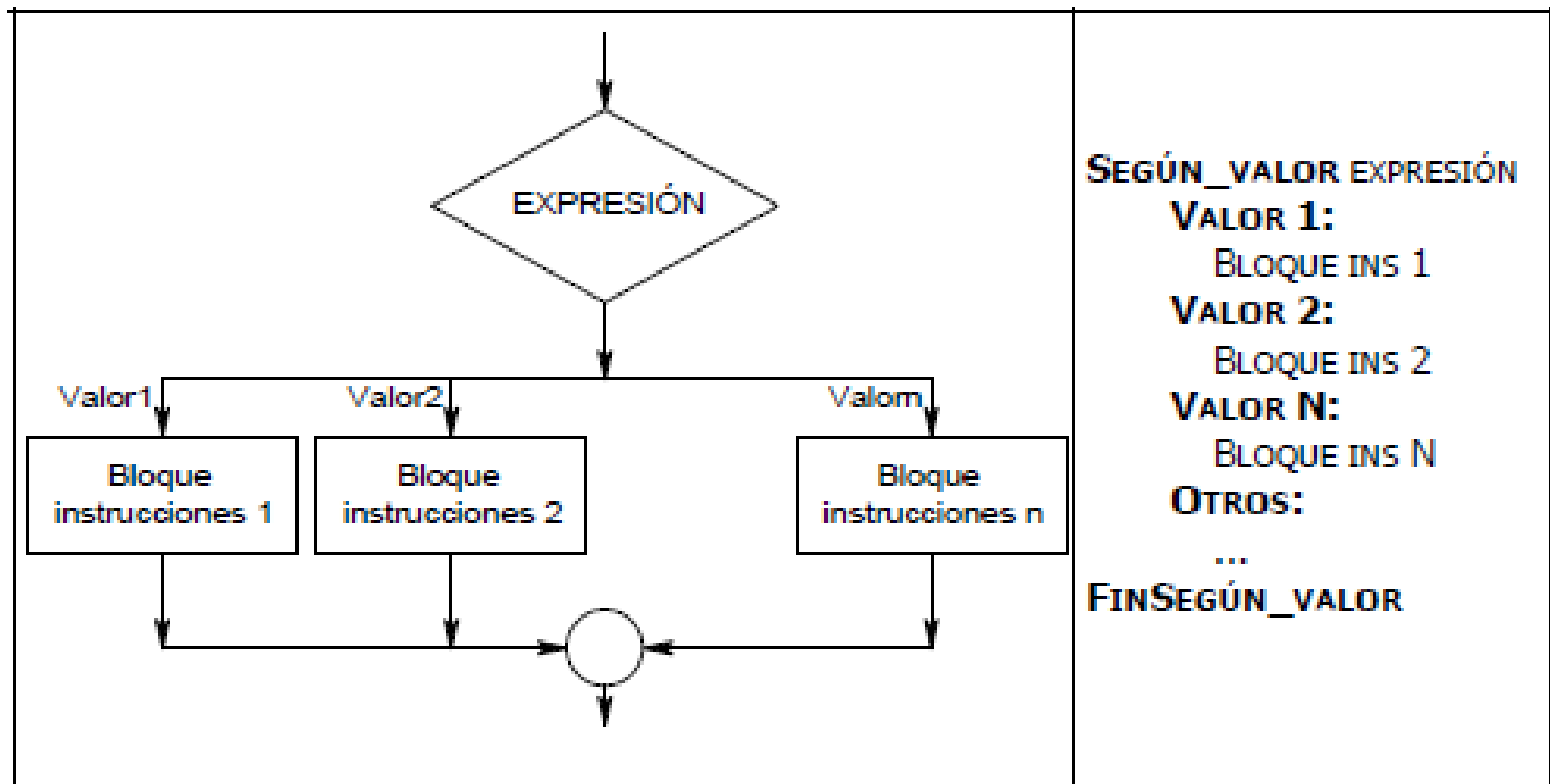
# Alternativa doble



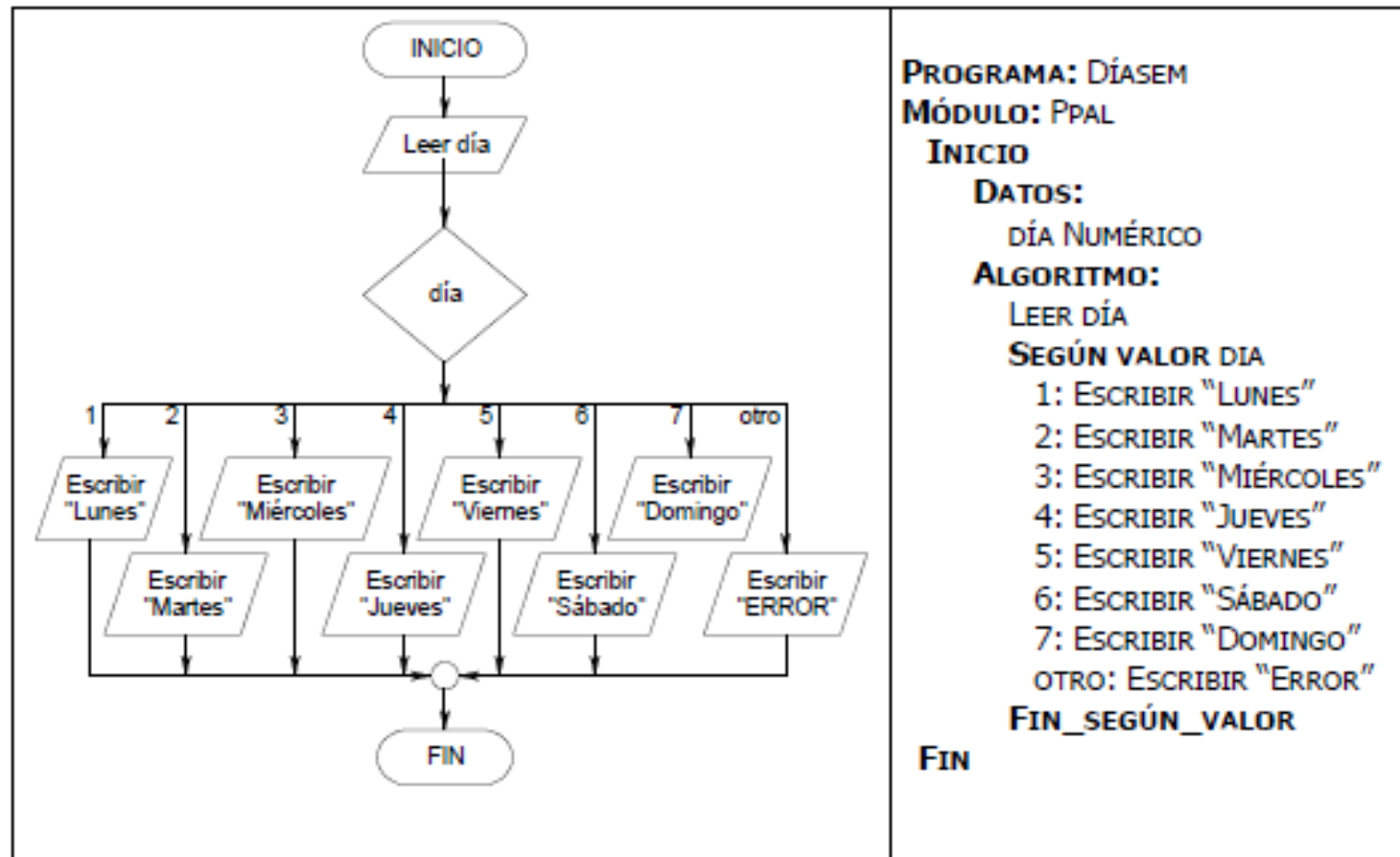
# Alternativa doble. Ejemplo



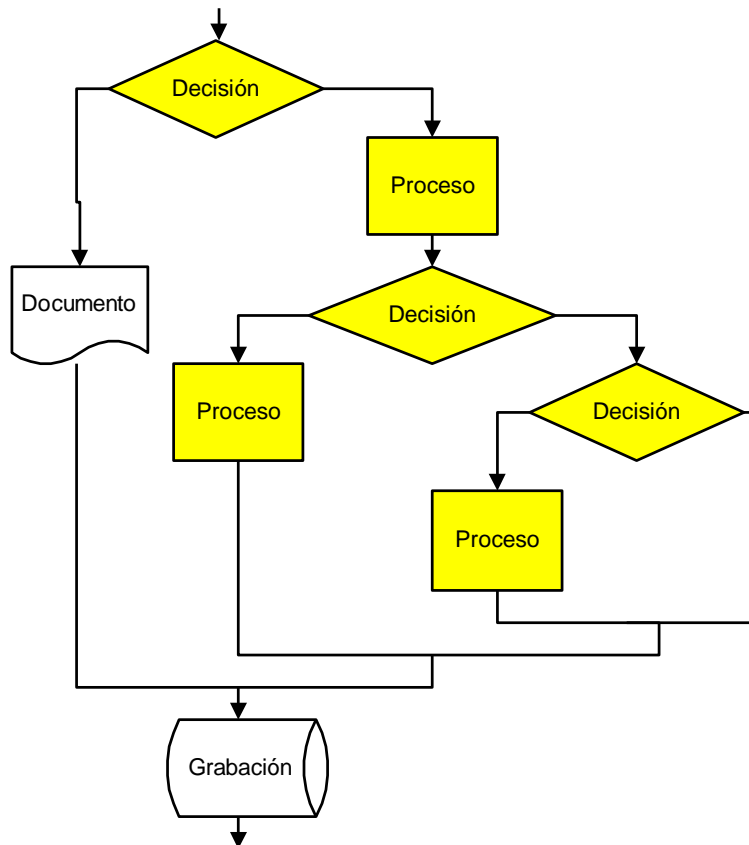
# Alternativa múltiple



# Alternativa múltiple. Ejemplo

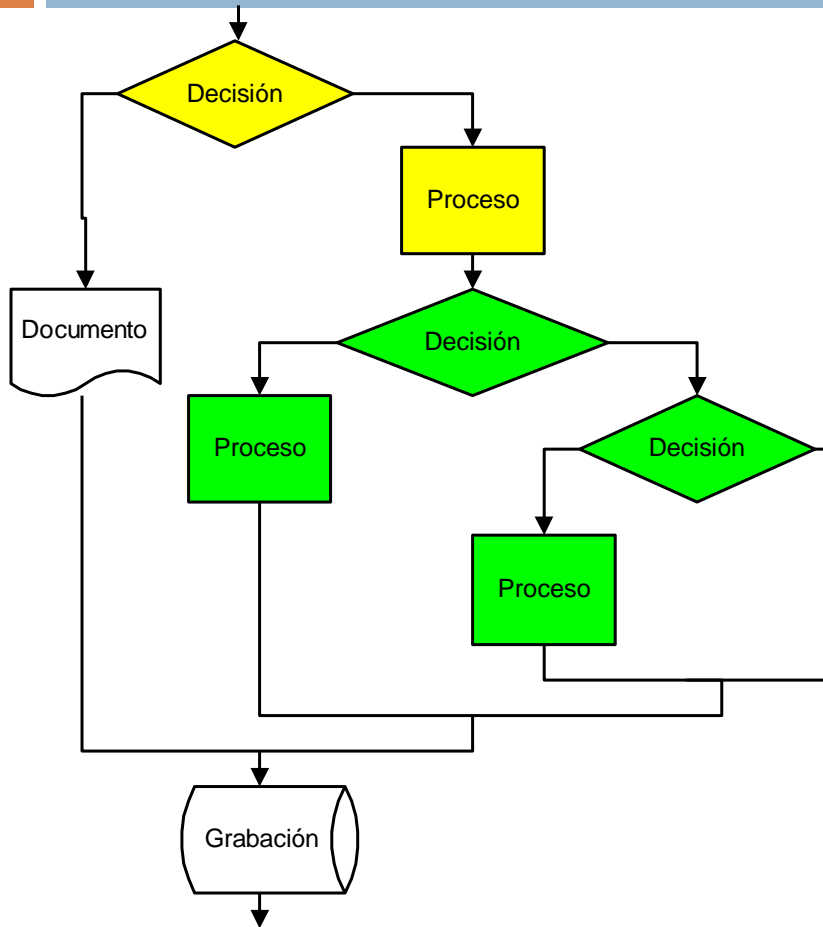


# Alternativa anidadas



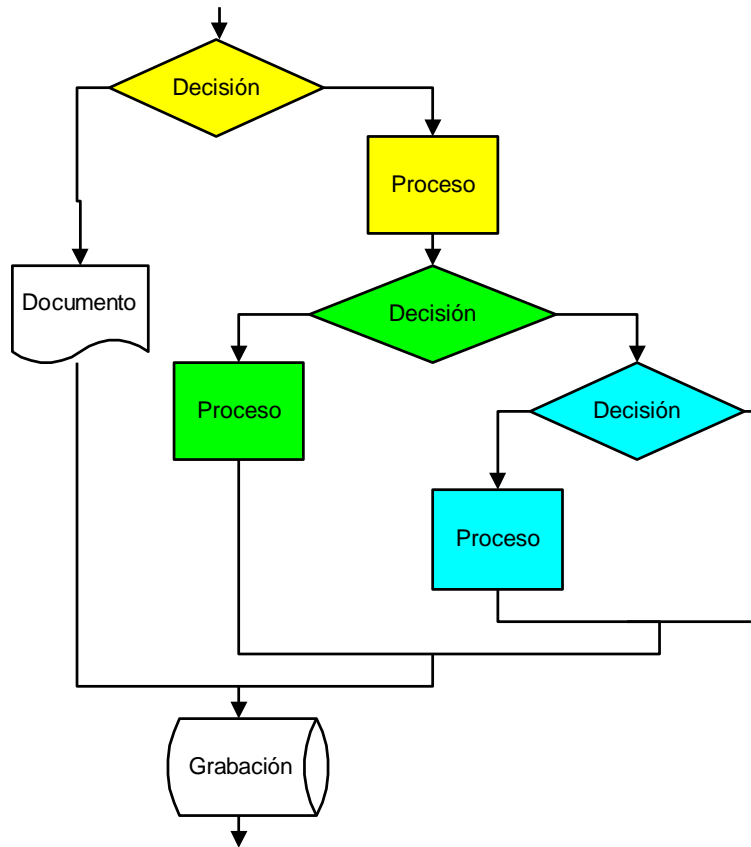
- Las instrucciones alternativas pueden estar anidadas unas dentro de otras
- Vemos como la zona amarilla esta toda condicionada al si de la primera pregunta

# Alternativa anidadas



- Si se cumple la primera condición se ejecutará la parte del sí
- La segunda condición encabeza toda la zona en verde

# Alternativa anidadas



- Si la condición verde es cierta se ejecutará la condición azul
- La ejecución del proceso azul dependerá de que la condición sea cierta o falsa
- Solo se ejecutará cuando la condición sea falsa

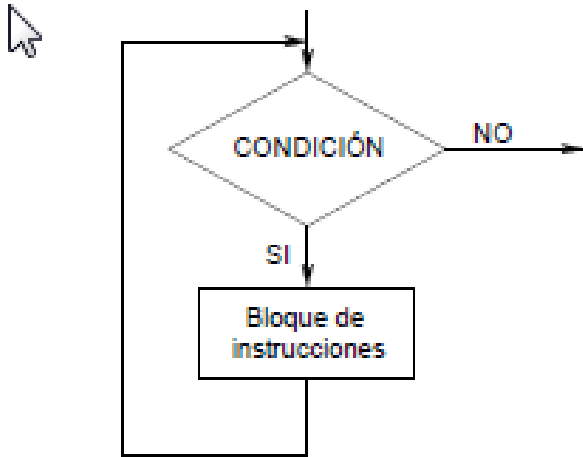




# Repetitiva

○ Iterativa

# Repetitiva. Condición inicial

FLUJOGRAMA	PSEUDOCÓDIGO
	<pre><b>MIENTRAS</b> CONDICIÓN <b>HACER</b>     INSTRUCCIÓN 1     INSTRUCCIÓN 2     ...     INSTRUCCIÓN N <b>FINMIENTRAS</b></pre>

# Repetitiva. Condición inicial. Ejemplo



**PROGRAMA:** NUMEROS\_50

**MÓDULO:** PPAL

**INICIO**

**DATOS:**

NUM NUMÉRICO

**ALGORITMO:**

ESCRIBIR "LOS Nº ENTRE EL 1 Y EL 50 SON"

NUM=1

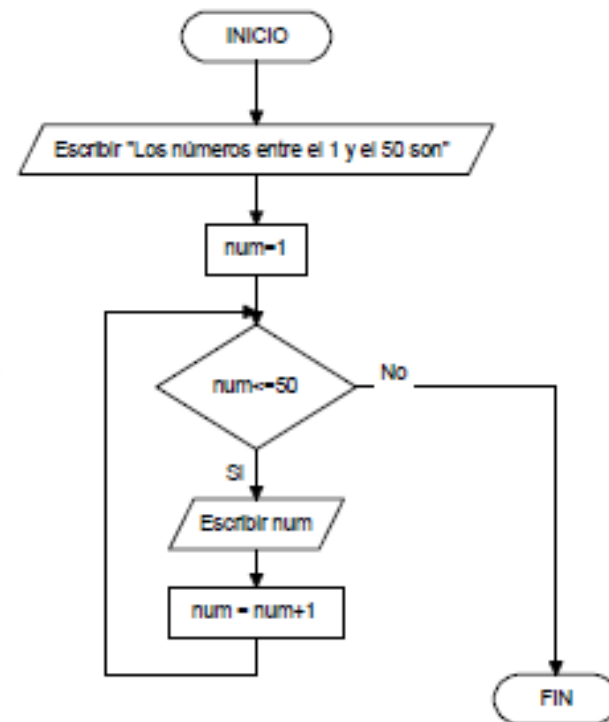
MIENTRAS (NUM <= 50) HACER

    ESCRIBIR NUM

    NUM=NUM+1

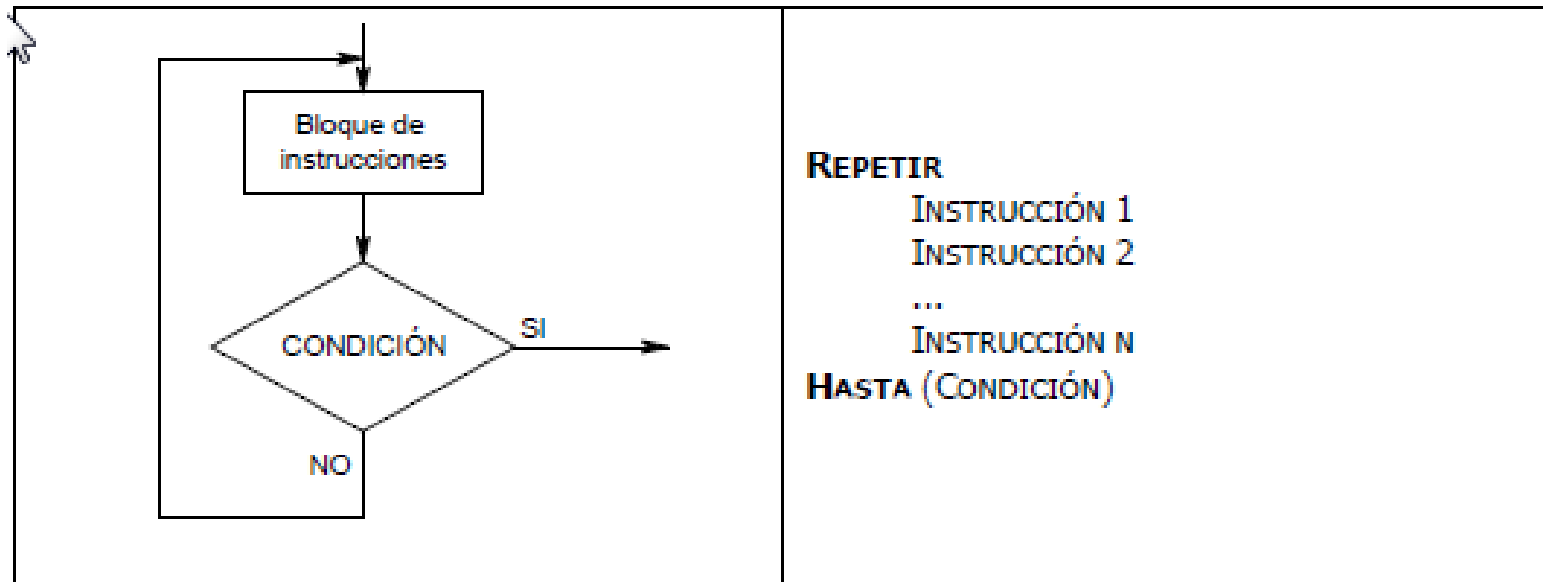
FINMIENTRAS

**FIN**

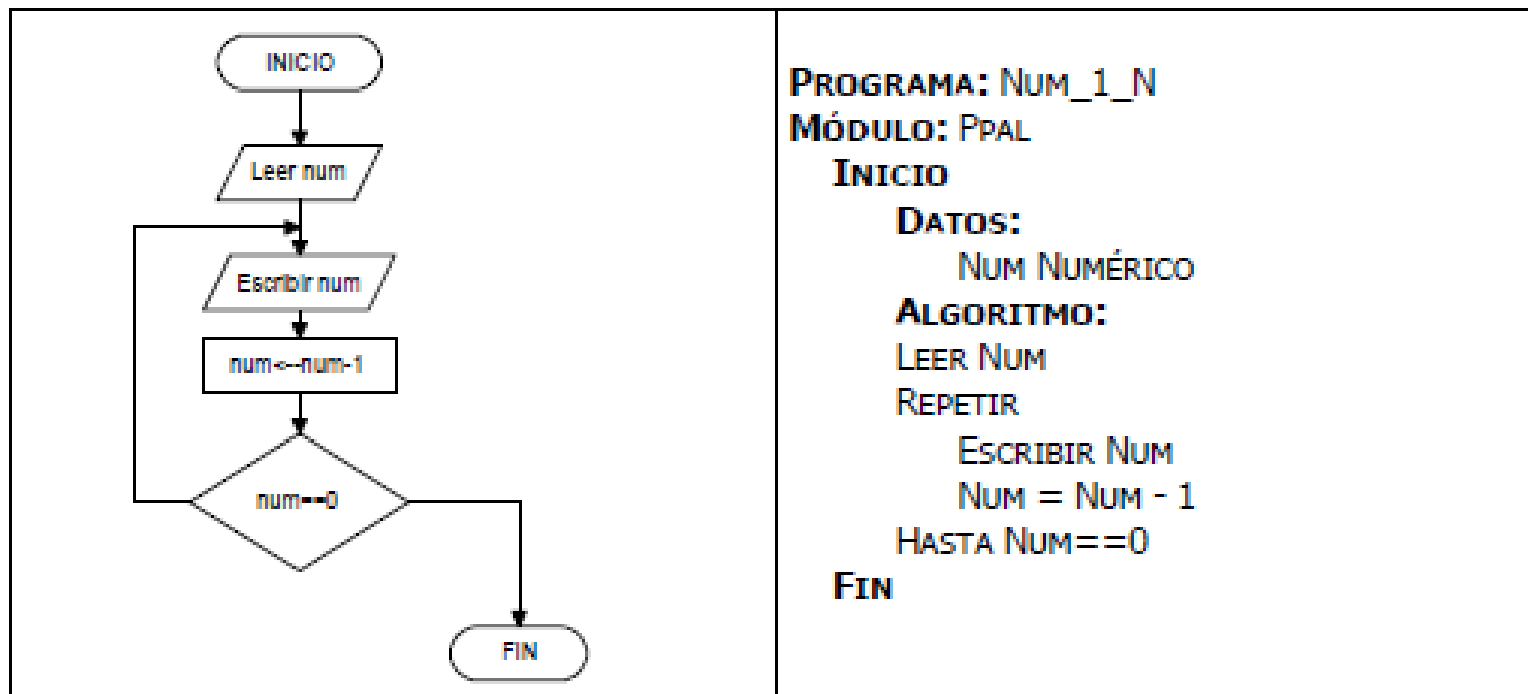


# Repetitiva. Condición final (Hasta)

Repetir... Hasta  
**Java no utiliza esta estructura**

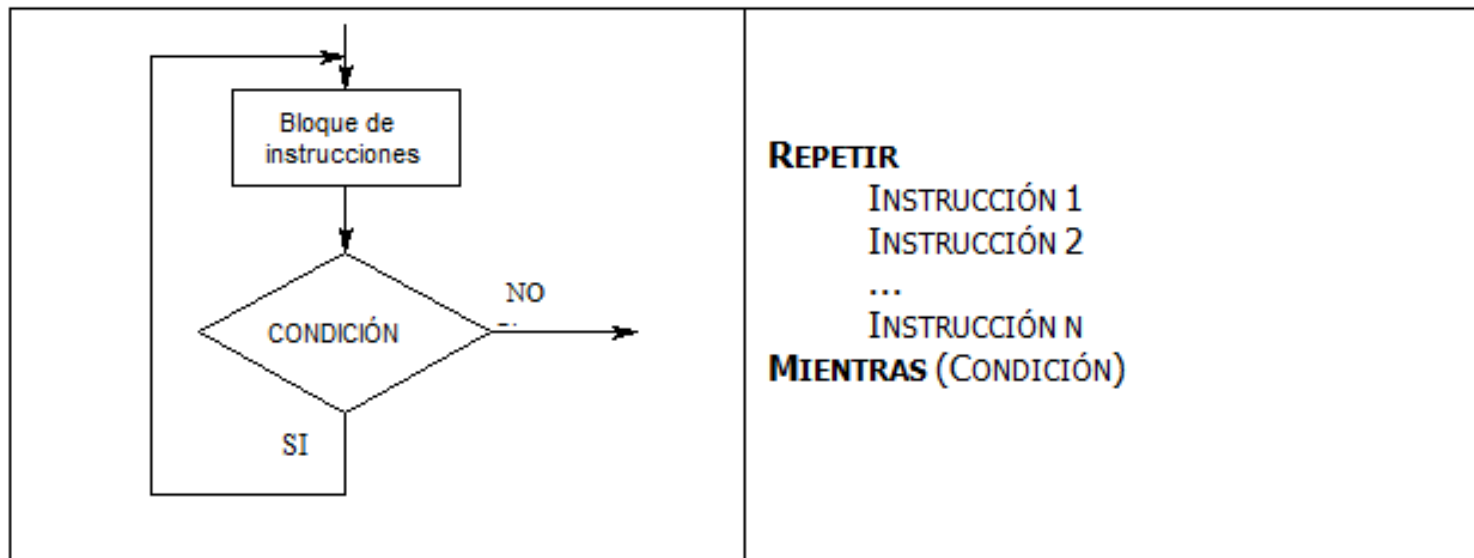


# Repetitiva. Condición final. Ejemplo

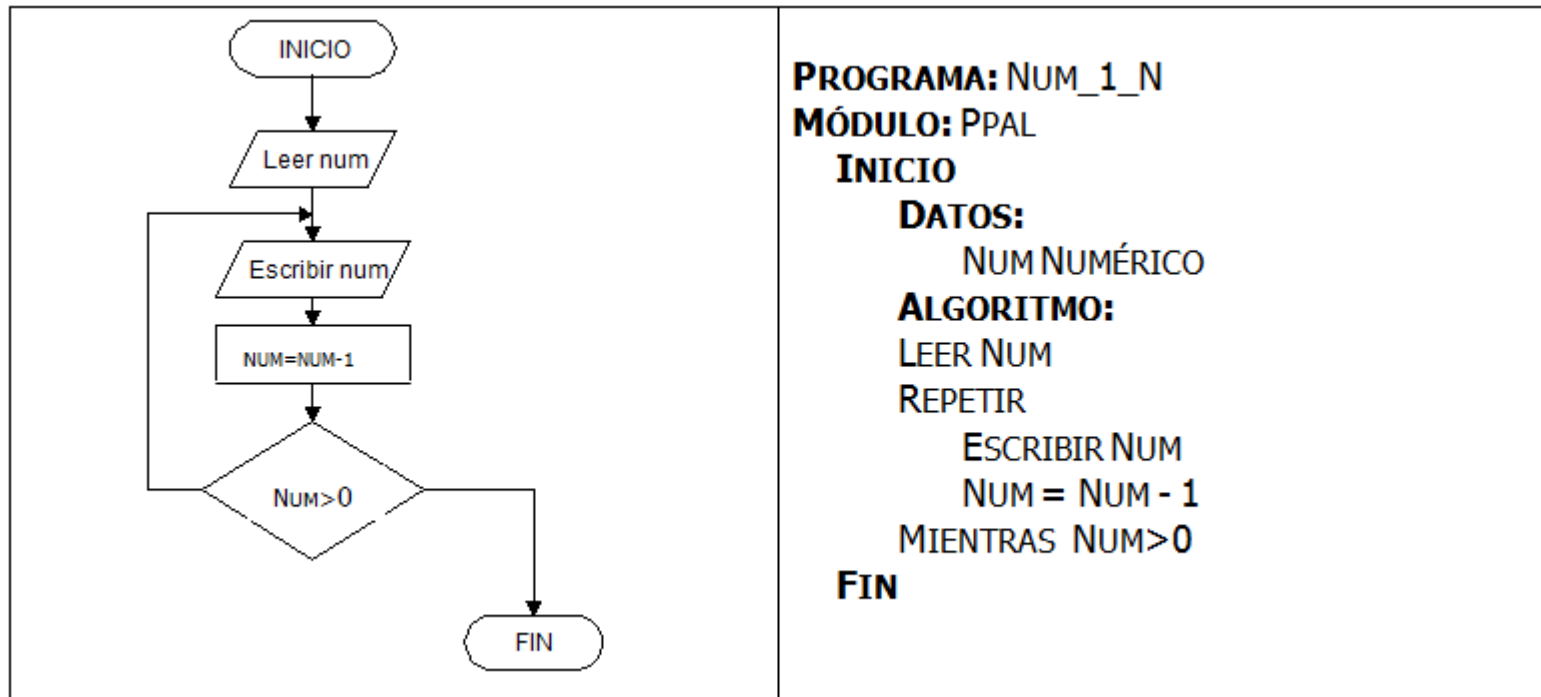


# Repetitiva. Condición final (Mientras)

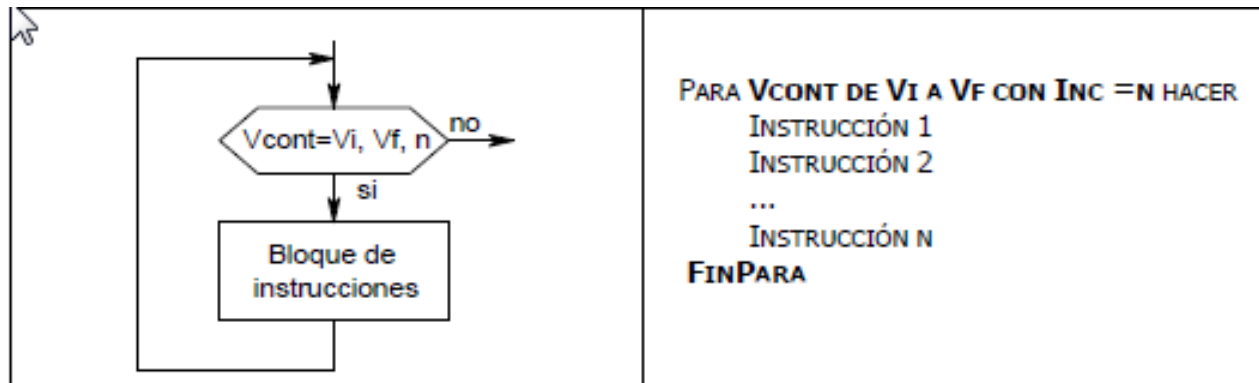
Repetir... Mientras  
**Java utiliza esta estructura**



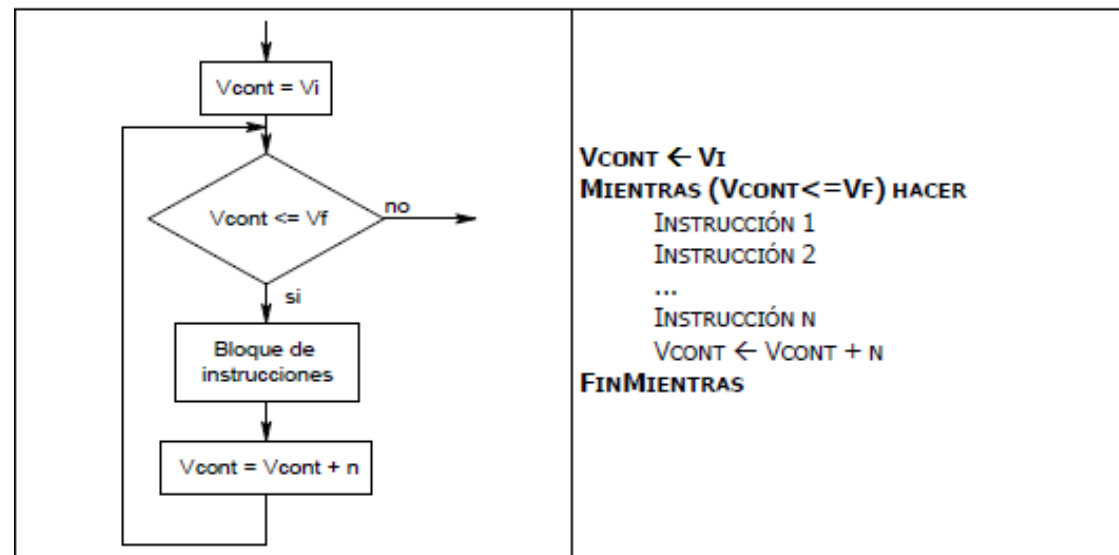
# Repetitiva. Condición final. Ejemplo



# Estructura Para...

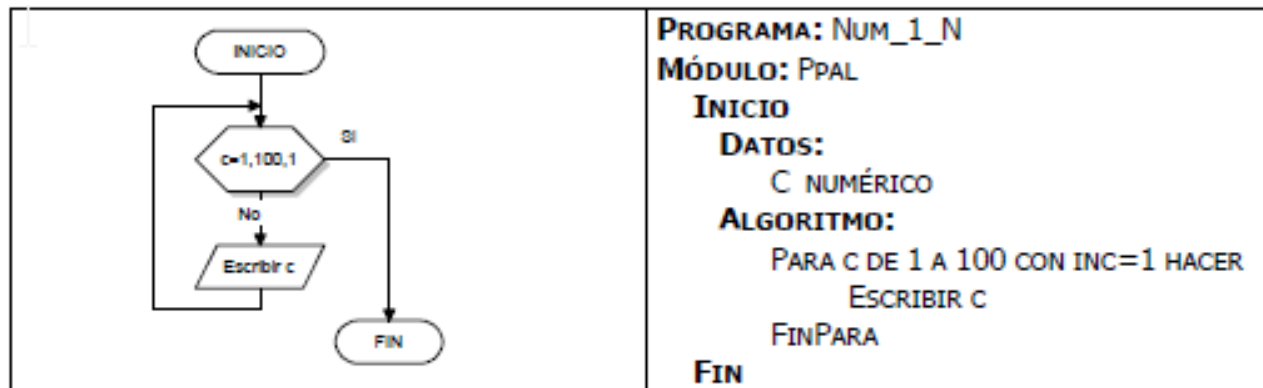


La instrucción Para  
es una forma  
compacta de  
representar un  
bucle Mientras  
Específico:





# Estructura Para... Ejemplo



Si resolviéramos el mismo problema utilizando una estructura **MIENTRAS** quedaría:

