



# **DIAGRAMAS DE FLUJO Y PSEUDOCÓDIGO**



# PSEUDOCÓDIGO



# TECNOLOGÍAS DE LA INFORMACIÓN II

## PSEUDOCÓDIGO Y DIAGRAMAS DE FLUJO

### Pseudocódigo.

- **El pseudocódigo** es una manera de escribir algoritmos de forma poco estricta (con una sintaxis relajada) o estructuras de datos poco detalladas, pero intentando acercar las ideas del algoritmos a estructuras y sintaxis parecidas a las de los lenguajes de alto nivel en los que vamos a programar el algoritmo.
- Es para ser leído por personas, por tanto no se preocupa en detalles sintácticos.
- Es un lenguaje de especificación de algoritmos, pero muy parecido a cualquier lenguaje de programación, por lo que luego su traducción al lenguaje de programación es muy sencillo, pero con la ventaja de que no se rige por las normas de un lenguaje en particular. Nos centramos más en la lógica del problema.
- El pseudocódigo también va a utilizar una serie de palabras claves o palabras especiales que va indicando lo que significa el algoritmo.



# TECNOLOGÍAS DE LA INFORMACIÓN II

## PSEUDOCÓDIGO Y DIAGRAMAS DE FLUJO

### Pseudocódigo - Sintaxis Utilizada.

1. **INICIO y FIN:** Por donde empieza y acaba el algoritmo.
2. **DATOS:** Aquí se declaran e inicializan las variables que utilizará el algoritmo.
3. **ALGORITMO:** En esta sección se escribe el algoritmo.

#### Pseudocódigo de un algoritmo genérico:

INICIO.

DATOS:

entero a ;

real b = 0 ;

**\*\* esto es un comentario \*\***

**\*\* declaración de una variable entera \*\***

**\*\* declaración e inicialización de una variable \*\***

ALGORITMO:

leer a ;

b = a + 5 ;

escribir b ;

FIN.



## DIAGRAMAS DE FLUJO



## Introducción.

- Es una notación gráfica para implementar algoritmos.
- Se basa en la utilización de unos símbolos gráficos denominados bloques, en los que escribimos las acciones que tiene que realizar el algoritmo.
- Estos bloques están conectados entre sí por líneas y eso nos indica el orden en el que tenemos que ejecutar las acciones.
- En todo algoritmo **siempre** habrá un bloque de inicio y otro de fin, para el principio y final del algoritmo.



# TECNOLOGÍAS DE LA INFORMACIÓN II

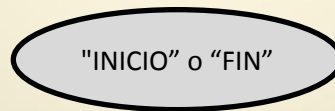
## PSEUDOCÓDIGO Y DIAGRAMAS DE FLUJO

### Símbolos Utilizados.

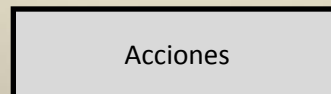
**Líneas de flujo:** Es una línea con una flecha (y solo una flecha) que permite conectar los bloques del diagrama. La flecha indica la secuencia en la que se van a ejecutar las acciones.



**Principio y Fin:** Todo algoritmo y por lo tanto, todo diagrama de flujo tiene un principio y un fin (y son únicos). Dentro de este bloque se coloca la palabra "INICIO" o "FIN" según corresponda.



**Proceso:** Aquí dentro se escribe la acción que debe realizar el programa. Si son varias, se escriben una debajo de la otra, sin olvidarse que se ejecutan una a una en forma secuencial según en qué orden fueron escritas.



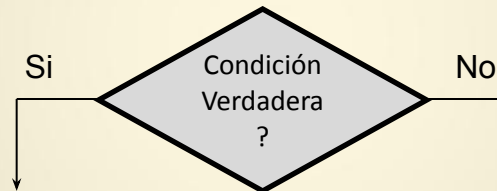


# TECNOLOGÍAS DE LA INFORMACIÓN II

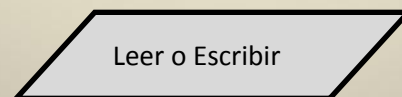
## PSEUDOCÓDIGO Y DIAGRAMAS DE FLUJO

### Símbolos Utilizados.

**Condición:** Dentro de este bloque se escribe una condición. Si ésta es verdadera, entonces el algoritmo tomará una de sus salidas, de lo contrario, tomará la siguiente. Permite representar estructuras del tipo selectivas y repetitivas.



**Entrada y Salida:** Representa acciones de entrada salida desde un teclado o hacia una pantalla respectivamente. Es decir, si debemos ejecutar una acción que consiste en leer un dato que se ingresa mediante el teclado de una PC y almacenarlo en la variable de nombre “a”, entonces dicha acción se describe dentro de este bloque como “leer a”.







# TECNOLOGÍAS DE LA INFORMACIÓN II

## PSEUDOCÓDIGO Y DIAGRAMAS DE FLUJO

### Símbolos Utilizados.

**Conectores:** Permiten “unir” diagramas de flujo cuando éstos no caben en una misma columna de la hoja por completo. Es decir, cuando debemos, por cuestiones de espacio en la hoja, fragmentar el programa entonces utilizamos estos bloques para indicar los puntos de unión. Cada par de puntos que se deben unir llevarán dentro de este bloque el mismo número.

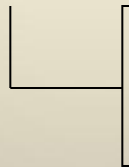
En la misma hoja:



En otra hoja:



**Comentarios:** Es una aclaración para comprender mejor el código del programa, pero no forma parte del código, es decir, no se ejecuta.

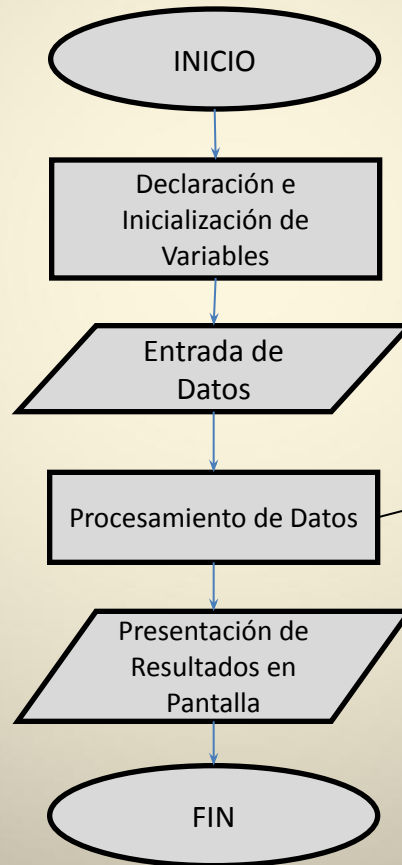




## TECNOLOGÍAS DE LA INFORMACIÓN II

### PSEUDOCÓDIGO Y DIAGRAMAS DE FLUJO

#### Diagrama de Flujo Elemental.



**RECORDAR: CONCEPTO DE ALGORITMO:** Un algoritmo es un conjunto de pasos o secuencia de instrucciones que, ejecutadas en un determinado orden, permiten resolver un problema determinado.

Esto es un comentario.



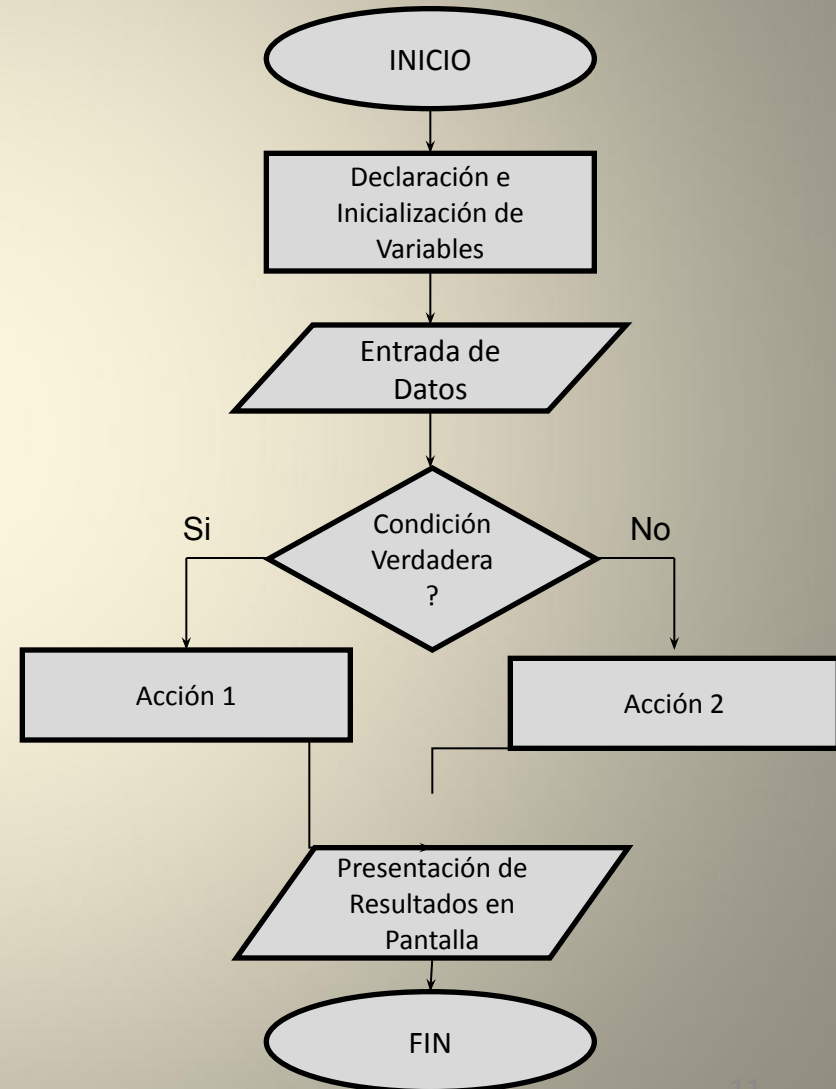
# TECNOLOGÍAS DE LA INFORMACIÓN II

## PSEUDOCÓDIGO Y DIAGRAMAS DE FLUJO

### Diagrama de Flujo Elemental.

Ejemplo de DF de un algoritmo genérico que incluye una bifurcación.

Si la Condición es Verdadera, se ejecuta la Acción 1 y en caso contrario (Falsa) la Acción 2.



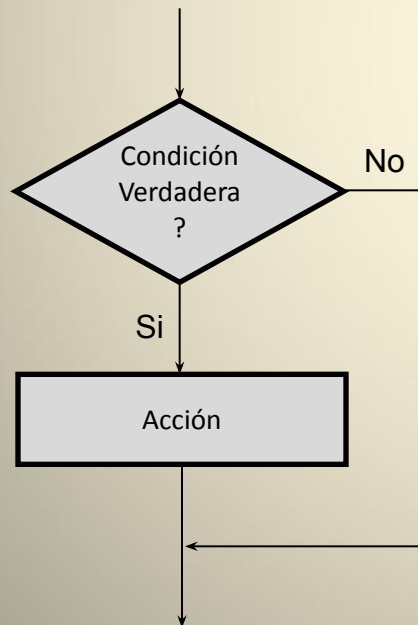


# TECNOLOGÍAS DE LA INFORMACIÓN II

## PSEUDOCÓDIGO Y DIAGRAMAS DE FLUJO

### Estructuras de Control Selectivas.

**Si:** Si la Condición es Verdadera, se ejecuta la Acción, sino el algoritmo continua con su ejecución.

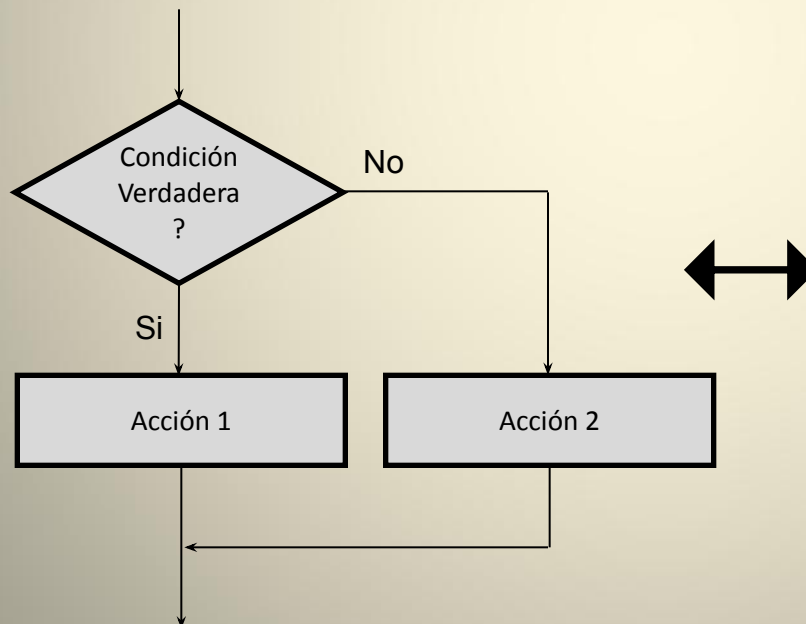


Si < Condición >  
    <Acción>  
Fin\_Si



### Estructuras de Control Selectivas.

**Si – Sino:** Si la Condición es Verdadera, se ejecuta la Acción 1, sino el algoritmo ejecuta la Acción 2. Luego el algoritmo continua con su ejecución.

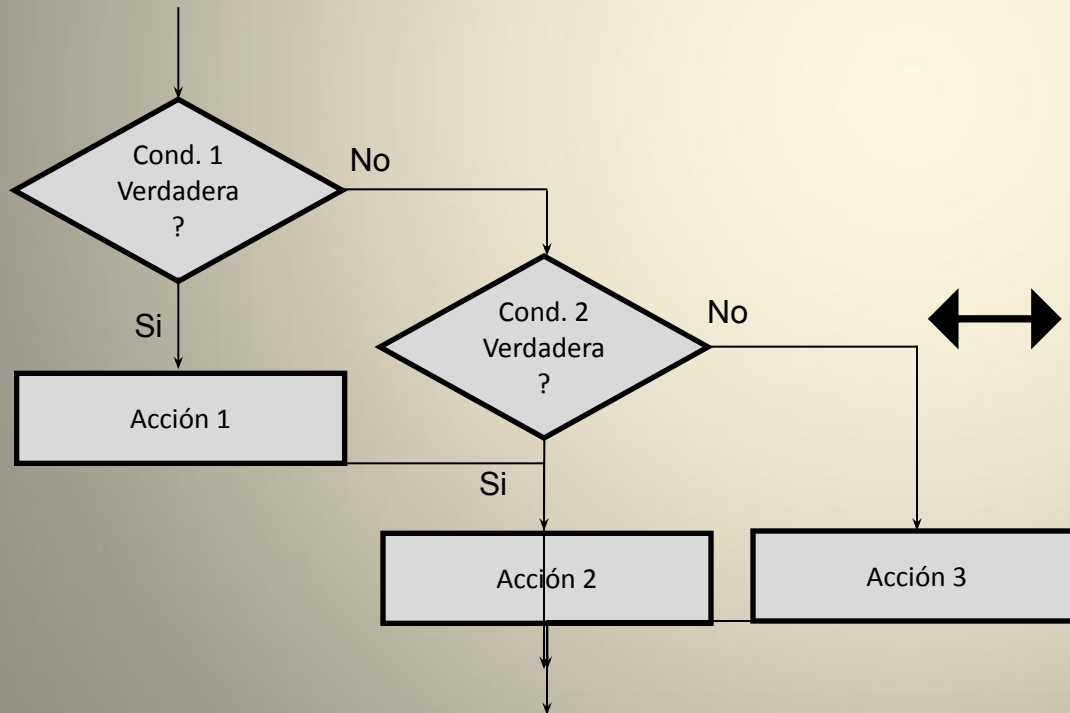


```
Si < Condición >  
    < Acción 1 >  
Sino < Acción 2 >  
  
Fin_Si
```



### Estructuras de Control Selectivas.

**Si – Sino – Si:** Si la Condición 1 es Verdadera, se ejecuta la Acción 1, sino el algoritmo evalúa la Condición 2. Si es Verdadera, ejecuta la Acción 2 y sino la Acción 3.

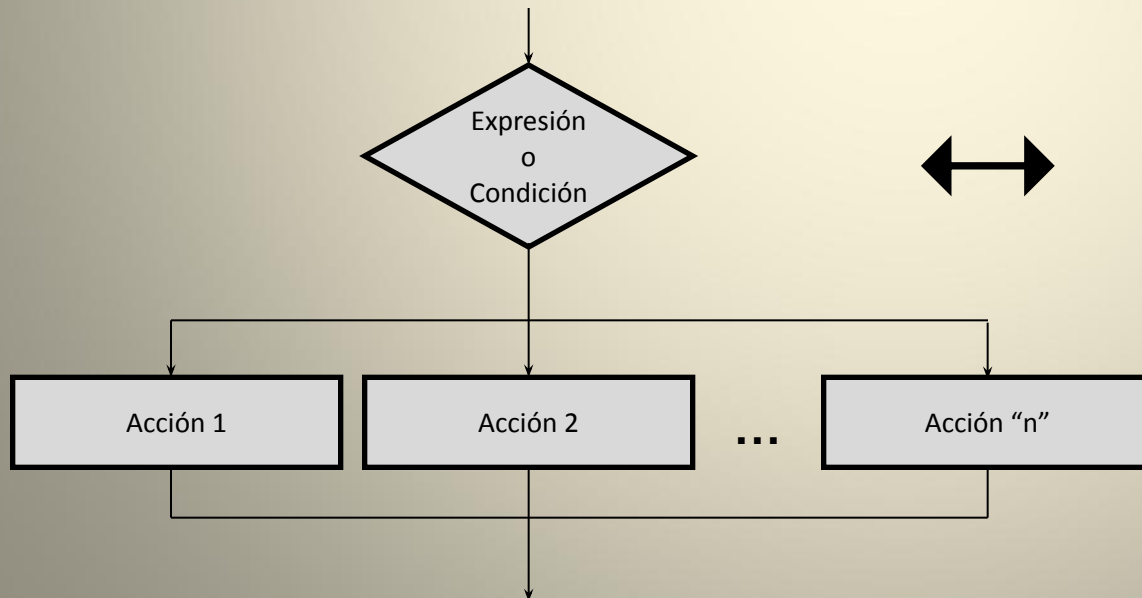


Si            < Condición 1 >  
              < Acción 1 >  
Sino\_Si < Condición 2 >  
              < Acción 2 >  
Sino            < Acción 3 >  
  
Fin\_Si



### Estructuras de Control Selectivas.

**Según (alternativa múltiple):** Se evalúa una condición o expresión que puede tomar “n” valores. Según el valor que la expresión tenga en cada momento se ejecutan las acciones correspondientes al valor. El valor con el que se compara la expresión, va a depender de los lenguajes, de lo que sea ese valor. En general ese valor puede ser un valor constante, un rango de valores o incluso otra condición.



Según <expresión>

< Valor 1 >: < Acción 1 >

< Valor 2 >: < Acción 2 >

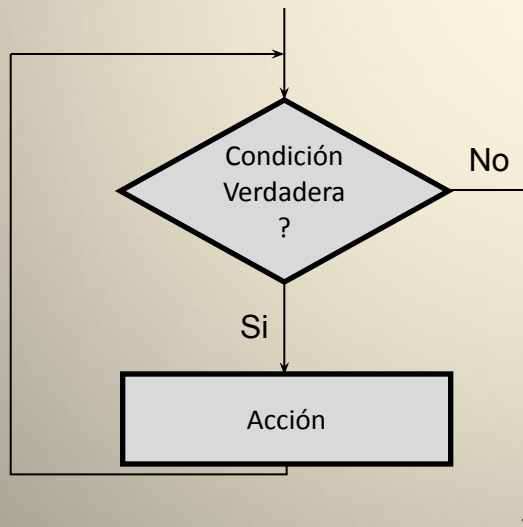
< Otro >: < Acciones >

Fin\_Según



## Estructuras de Control Repetitivas.

**Mientras:** Se evalúa la Condición. Si es Verdadera, se ejecuta la Acción y se evalúa nuevamente la Condición. En el momento en el que la Condición sea Falsa se sale del bucle y se continua con la ejecución del algoritmo. Al evaluarse la Condición, al principio, antes de entrar en el bucle, si la condición es Falsa, nunca se entrará en el bucle. Por lo tanto se utiliza obligatoriamente este tipo de bucle en el caso de que exista la posibilidad de que el bucle pueda ejecutarse 0 veces.



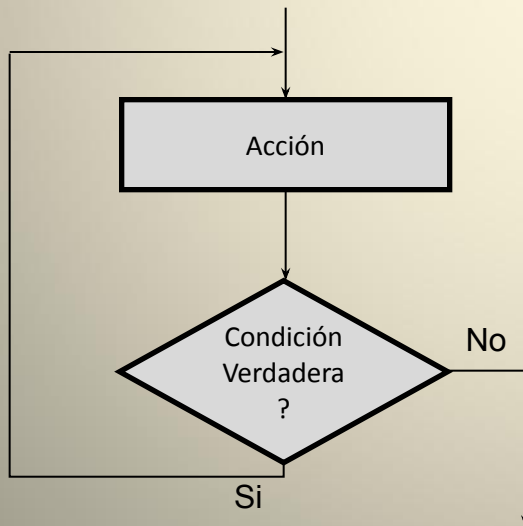
Mientras < Condición >  
    < Acción >  
Fin\_Mientras





## Estructuras de Control Repetitivas.

**Hacer - Mientras:** En primera instancia se ejecuta la Acción. Posteriormente, se evalúa la Condición. Si es Verdadera, se ejecuta nuevamente la Acción. Si es Falsa, se continúa con la ejecución del algoritmo. Se repite el bucle mientras la condición sea Verdadera. La Condición se evalúa siempre al final del bucle, si es Verdadera se ejecuta nuevamente la Acción, si es Falsa se sale del bucle. Como la Condición se evalúa al final, incluso aunque la primera vez ya sea Falsa, el bucle (la Acción) se habrá ejecutado al menos una vez.

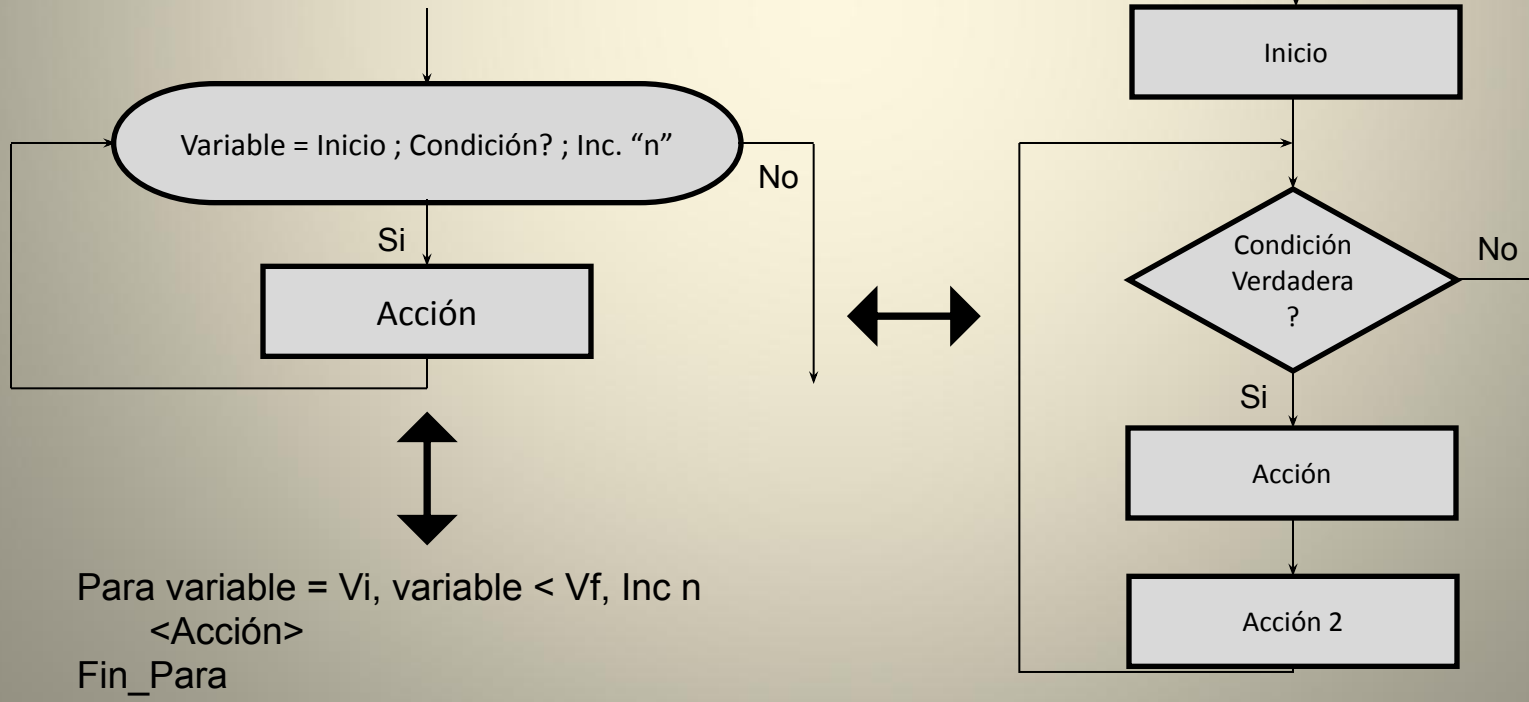


Hacer < Acción >  
Mientras < Condición >



### Estructuras de Control Repetitivas.

**Desde - Hasta:** Se utiliza cuando se sabe el número exacto de veces que hay que iterar. Para ello el bucle llevará asociado una variable “índice”, a la que se le asigna un valor inicial y se determina cual va a ser su valor final y además se va a incrementar (o decrementar) en cada iteración de bucle en un valor constante.

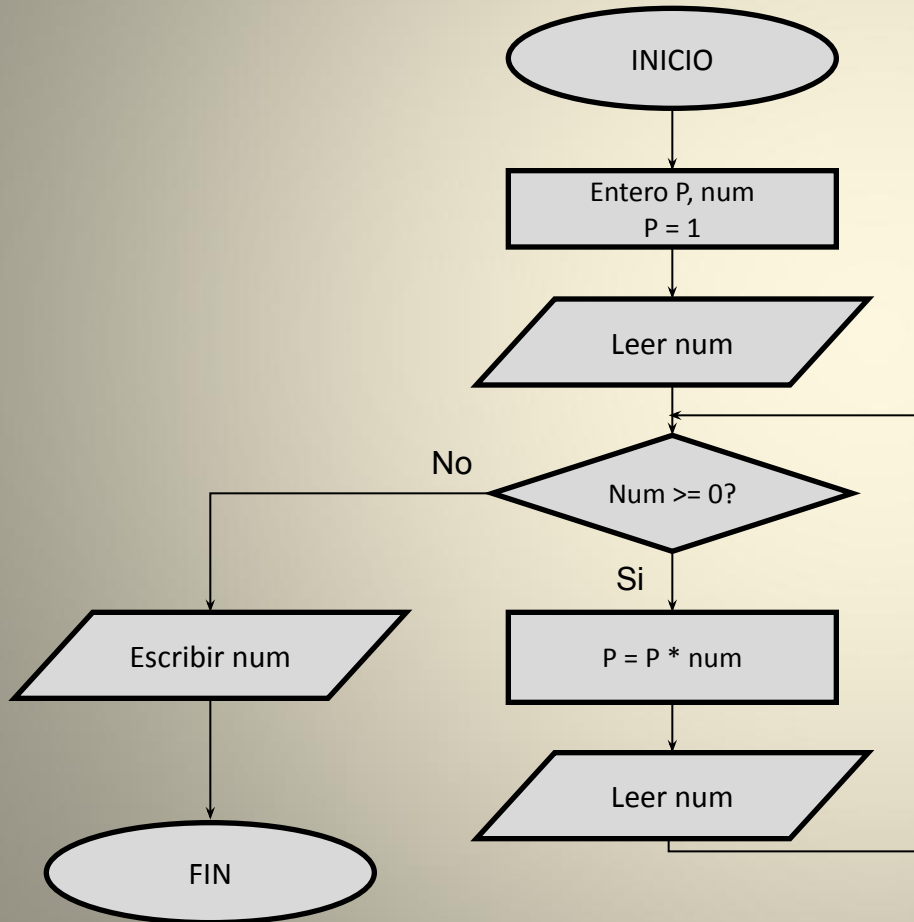




# TECNOLOGÍAS DE LA INFORMACIÓN II

## PSEUDOCÓDIGO Y DIAGRAMAS DE FLUJO

**Ejemplo:** Mostrar el producto de números enteros positivos entrados por teclado hasta el ingreso de un número negativo.



### Pseudocódigo:

Módulo: Principal

INICIO

DATOS:

P, num: entero

ALGORITMO:

$P \leftarrow 1$

Leer num

Mientras num >= 0

$P \leftarrow p * num$

Leer num

Fin\_mientras

Escribir p

FIN