



# Unidad I: Herramientas de Programación para la Solución de Problemas con Computadoras

Un recorrido por las principales herramientas y técnicas que permiten diseñar algoritmos y traducirlos a lenguajes de programación.

# ¿Qué es un Algoritmo?

Para implementar la solución de un problema mediante una computadora, es necesario establecer una serie de pasos denominada **algoritmo**: una serie de operaciones detalladas y no ambiguas que conducen a la resolución de un problema.

## Preciso

Indica el orden exacto en que deben realizarse los pasos hacia la solución.

## Definido

Bajo las mismas condiciones, el resultado siempre debe ser el mismo.

## Finito

No debe caer en repeticiones innecesarias; debe terminar en algún momento.

# Partes de un Algoritmo

Todo algoritmo que se transformará en un programa de computadora debe considerar tres componentes esenciales:

## Descripción de datos

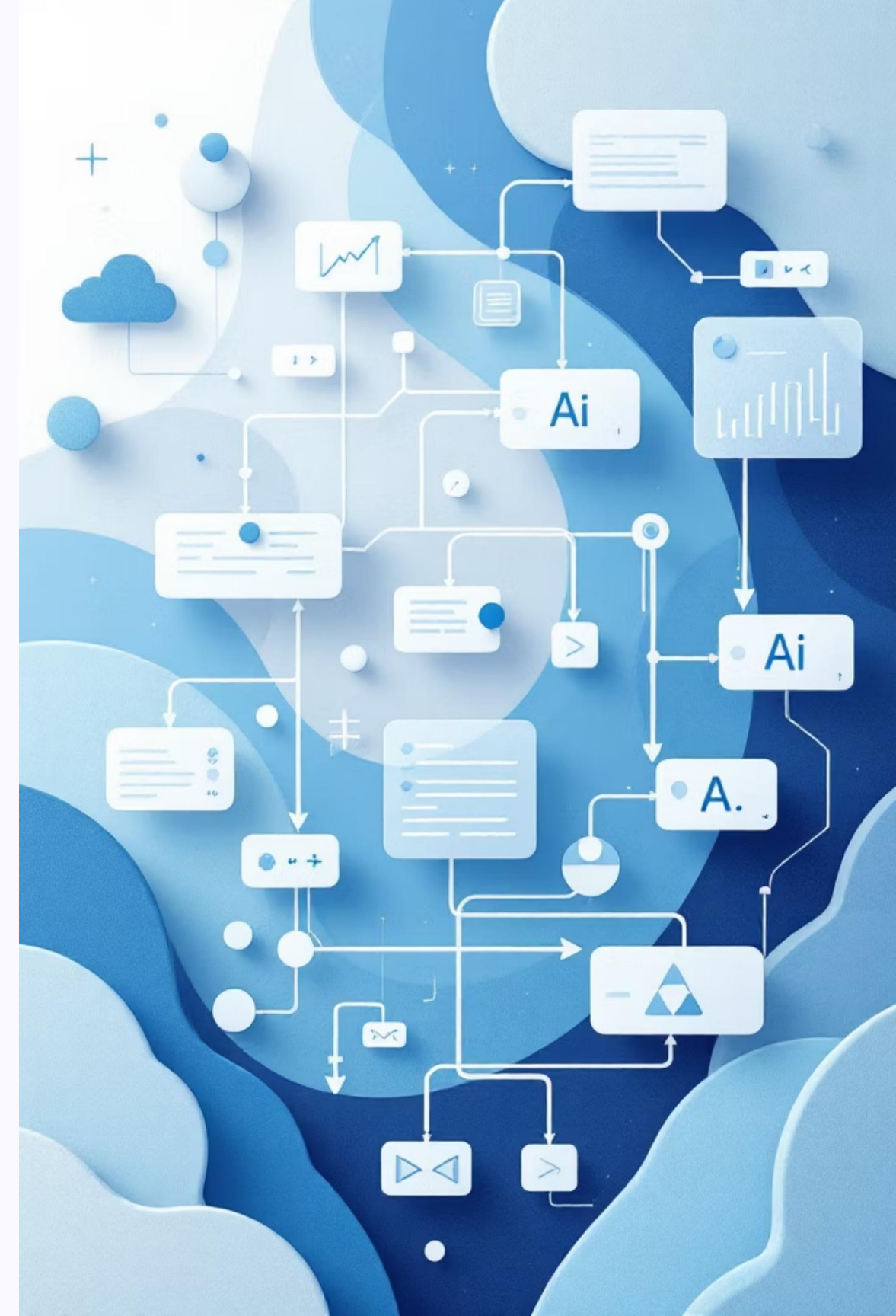
Los datos que serán manipulados durante el proceso.

## Descripción de acciones

Las acciones que deben ejecutarse para manipular los datos.

## Resultados

Los resultados que se obtendrán por la manipulación de los datos.



# Herramientas de Programación

Las herramientas de programación son métodos que permiten la elaboración de algoritmos escritos en un lenguaje entendible. Las más utilizadas a lo largo del estudio son:

01

---

## Pseudocódigo

Serie de pasos detallados y claros que permiten pasar casi directamente la solución a un lenguaje de programación.

02

---

## Diagramas de flujo

Representación gráfica mediante símbolos especiales de los pasos secuenciales para resolver un problema.

03

---

## Diagramas Nassi-Shneiderman (N/S)

Herramienta alternativa para la representación estructurada de algoritmos.

# Identificadores: Variables y Constantes

Los **identificadores** son los nombres asignados a los objetos (variables o constantes) que intervienen en los procesos de solución de un problema.

## Constante

Su valor **no cambia** durante la ejecución.

Ejemplo:

$$PI = 3.1416$$

## Variable

Su valor **cambia** durante la ejecución.  
Ejemplos: sueldo, descuento, largo (L) y ancho (A) de un rectángulo para calcular su área.

# Tipos de Variables

Las variables se clasifican según lo que representan en el algoritmo:

1

## Numéricos

10

8

### Entero

Sin valores decimales.  
Ejemplos: número de personas, días trabajados, edad.

### Real - decimales

Con valores decimales.  
Ejemplos: sueldo de una persona, porcentaje de una cantidad.

### String / Cadena

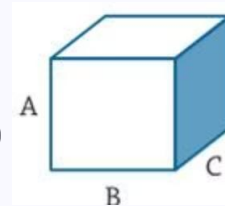
Valores de caracteres.  
Ejemplos: nombre, sexo, tipo de sangre, verdadero/falso.

# Pseudocódigo

El pseudocódigo es una de las herramientas más conocidas para el diseño de soluciones. Permite pasar casi de manera directa la solución a un lenguaje de programación específico.

Ejemplo:

Volumen de una caja ( $A \times B \times C$ )



- 1 Inicio.
- 2 Leer (A, B y C)
- 3 Calcular  $V \leftarrow A * B * C$ .
- 4 Escribir (V).
- 5 Fin.

- **Instrucción Inicio / Fin**

**Inicio** y **Fin** son Palabras Reservadas, es decir, palabras que pueden ser usadas únicamente para cumplir una función específica, por tanto, éstas no pueden ser usadas como identificadores.

- **Instrucción de Asignación**

Esta instrucción permite representar las operaciones aritméticas. Siempre que se desee realizar un procedimiento, operación o cálculo – con operadores aritméticos, relacionales y/o lógicos –, se debe utilizar una instrucción de Asignación.

- **Instrucción de Lectura**

Entonces Leer es una palabra reservada.

- **Instrucción de Escritura**

Siempre que se desee mostrar, visualizar o imprimir los resultados, en el mundo del pseudocódigo se usa la instrucción Escribir.

# Pseudocódigo

En pseudocódigo, los algoritmos pueden escribirse en tres secciones:

- **Cabecera**

En la cabecera de un algoritmo puede indicarse el nombre (identificador) asignado al mismo.

```
Algoritmo <nombre_del_algoritmo>
```

- **Declaraciones**

En esta sección se pueden declarar las constantes y las variables que se usan en el algoritmo.

```
NUMERO_PI <- 3.141592  
  
Definir area, radio Como Real
```

- **Cuerpo**

En el cuerpo se escriben todas las instrucciones del algoritmo.

```
Escribir "Introduzca radio: "  
  
Leer radio  
  
area <- NUMERO_PI * radio ^ 2  
  
Escribir "El área es: ", area
```

# Diagramas de Flujo

Los diagramas de flujo representan visualmente las operaciones y su secuencia para solucionar un problema, mediante **símbolos especiales**. Facilitan la comunicación entre programadores y usuarios, y permiten detectar errores de lógica antes de codificar.



## Terminal / Inicio

Marca el inicio o fin del proceso.



## Entrada de datos

Representa la lectura de datos de entrada.



## Proceso

Indica una operación o cálculo a realizar.



## Decisión





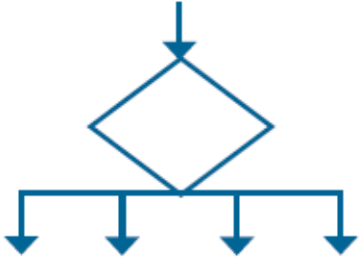



Bifurcación condicional con dos o más salidas.

# Operadores en Diagramas de Flujo

Dentro de los diagramas de flujo se utilizan los siguientes símbolos para indicar las operaciones que se realizan sobre los datos:

Símbolo	Operación
+	Suma
-	Resta
*	Multiplicación
/	División
^	Exponenciación

Símbolo	Operación
>	Mayor que
<	Menor que
>=	Mayor o igual que
<=	Menor o igual que
<>	Diferente que

Símbolo	Significado
	Terminal /Inicio.
	Entrada de datos.
	Proceso.
	Decisión.
	Decisión múltiple.
	Imprimir resultados.
	Flujo de datos.
	Conectores.

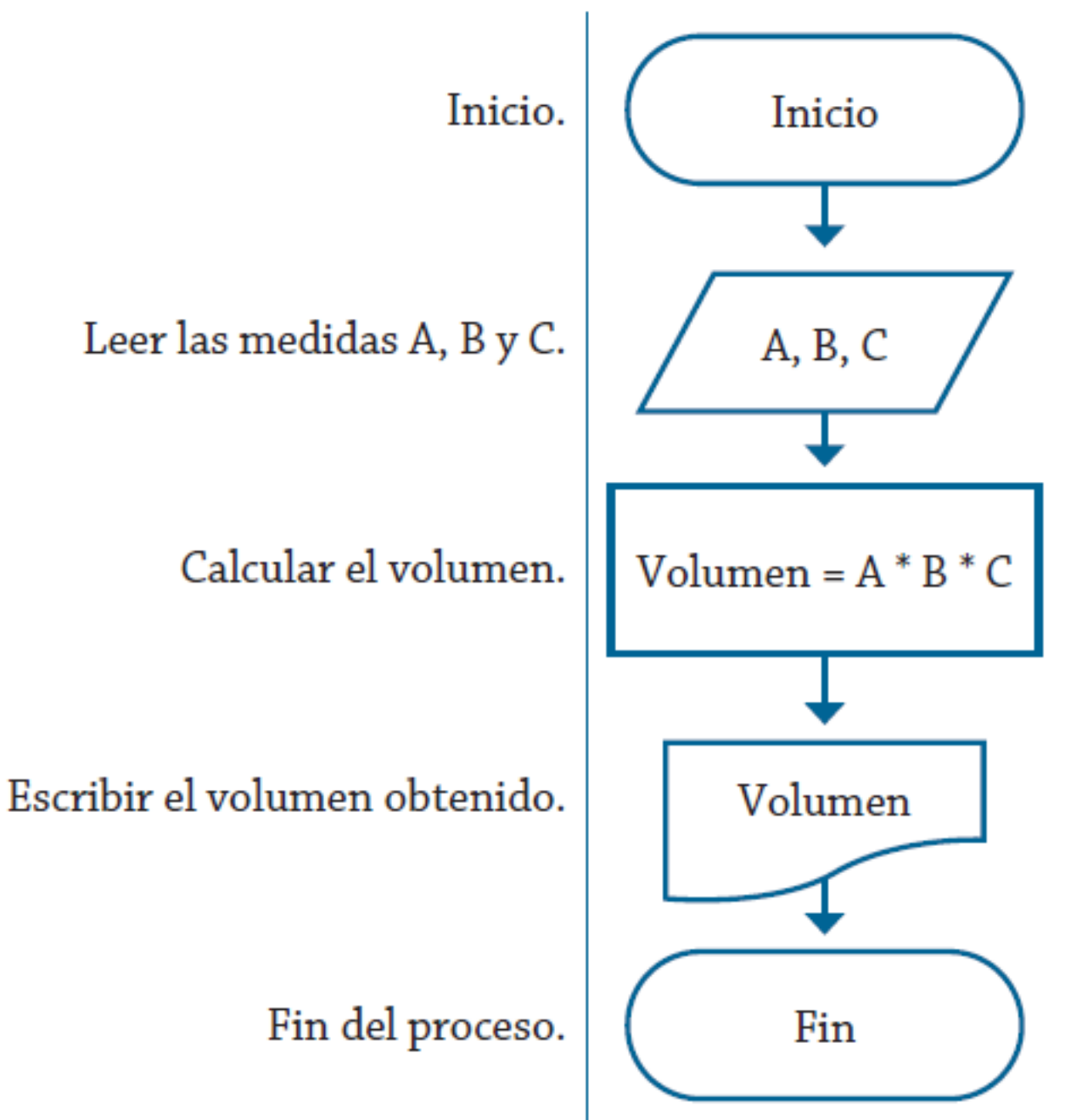
# Diagrama de Flujo

En los diagramas de flujo se pueden utilizar los símbolos con los cuales se indican las operaciones que se efectuarán a los datos con el fin de producir un resultado.

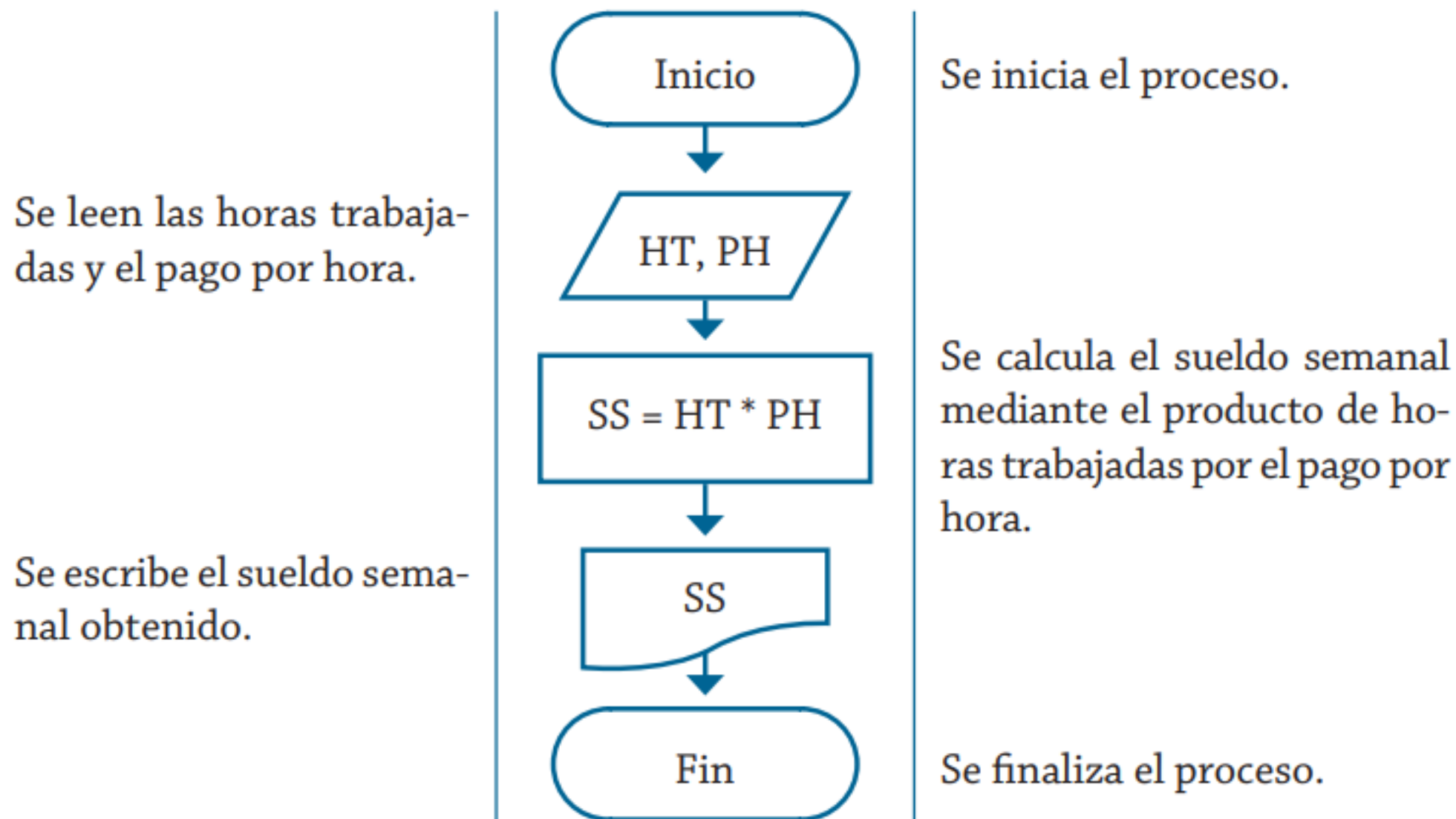
# Ejemplo: Diagrama de Flujo para el Volumen de una Caja

El siguiente diagrama de flujo ilustra los mismos pasos del pseudocódigo anterior, ahora representados gráficamente. Se establece de forma precisa la secuencia para obtener el volumen  $V = A \times B \times C$ .

- ❑ Tanto el pseudocódigo como el diagrama de flujo son representaciones equivalentes del mismo algoritmo. La elección entre uno u otro depende de la preferencia del programador y el nivel de experiencia.



# Ejemplo: Diagrama de Flujo



# Ejemplo: Diagrama de Flujo

