

El Pensamiento Algorítmico - Diagramas de Flujo

Materia: Algoritmos y Programación

Prof. D.Sc. BARSEKH-ONJI Aboud

Facultad de Ingeniería
Universidad Anáhuac México

7 de diciembre de 2025

Agenda

1. Diagramas de flujo: La representación gráfica

1.1 Simbología empleada

Ejemplos de Diagramas de Flujo

Diagramas de Flujo: La Representación Gráfica

Definición

Si el pseudocódigo es la descripción textual de un algoritmo, el **diagrama de flujo** es su representación gráfica. Consiste en un conjunto de símbolos geométricos estándar conectados por flechas que muestran la dirección del flujo de ejecución.

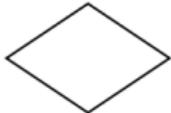
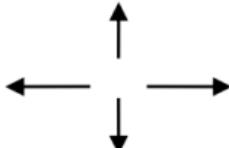
Ventaja Principal: Inmediatez Visual

Permite a cualquier persona, incluso sin formación técnica, seguir la lógica de un proceso de principio a fin. Son herramientas excepcionales para la documentación, la formación y la detección de errores en la fase de diseño.

Simbología para Diagramas de Flujo (Parte 1)

Símbolo	Nombre	Descripción
	Terminal	Representa el punto de Inicio o Fin del algoritmo.
	Entrada/Salida	Se utiliza para leer datos (entrada) o escribir resultados (salida).
	Proceso	Indica una operación, como un cálculo matemático o una asignación de valor.

Simbología para Diagramas de Flujo (Parte 2)

Símbolo	Nombre	Descripción
	Decisión	Representa una bifurcación en el flujo. Contiene una condición que se evalúa como verdadera o falsa.
	Línea de Flujo	Flechas que indican el orden de ejecución, conectando los símbolos.

Simbología para Diagramas de Flujo (Parte 2)

Símbolo	Nombre	Descripción
	Conejero	Permite conectar partes del diagrama para evitar el cruce de líneas.
	Subrgupo	Permite incluir diferentes procesos, un subsistema, dentro de un marco paralelo

Ejemplo 1: Área y Perímetro de un Rectángulo

Descripción

Un algoritmo secuencial clásico. Se leen dos valores, se realizan dos cálculos distintos y se muestran los resultados. El flujo es lineal y directo.

Ejemplo 1: Área y Perímetro de un Rectángulo

Algorithm 1 Área y Perímetro

- 1: **INICIO**
 - 2: **LEER** L, B
 - 3: **ASIGNAR** AREA \leftarrow L * B
 - 4: **ASIGNAR** PERIMETRO \leftarrow 2 * (L + B)
 - 5: **ESCRIBIR** AREA, PERIMETRO
 - 6: **FIN**
-

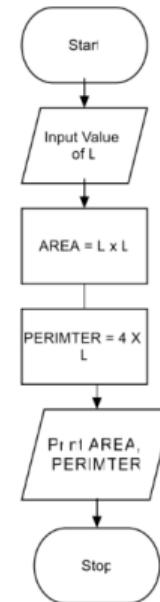


Figura 1: Diagrama de flujo.

Descripción del Algoritmo

Lógica de Decisión Anidada

Este ejemplo es excelente para ilustrar el poder de las decisiones anidadas. Se comparan los números por pares para determinar cuál es el mayor de todos.

Flujo del Proceso

El algoritmo primero compara los dos primeros números. Basado en ese resultado, el ganador de esa comparación se enfrenta al tercer número. El diagrama de flujo visualiza claramente las múltiples rutas que puede tomar la ejecución dependiendo de los valores de entrada.

Ejemplo 2: Pseudocódigo y Diagrama de Flujo

Algorithm 2 Mayor de Tres Números

```
1: INICIO
2: LEER num1, num2, num3
3: if num1 > num2 then
4:   if num1 > num3 then
5:     ESCRIBIR num1
6:   else
7:     ESCRIBIR num3
8:   end if
9: else
10:  if num2 > num3 then
11:    ESCRIBIR num2
12:  else
13:    ESCRIBIR num3
14:  end if
15: end if
16: FIN
```

Ejemplo 2: Pseudocódigo y Diagrama de Flujo

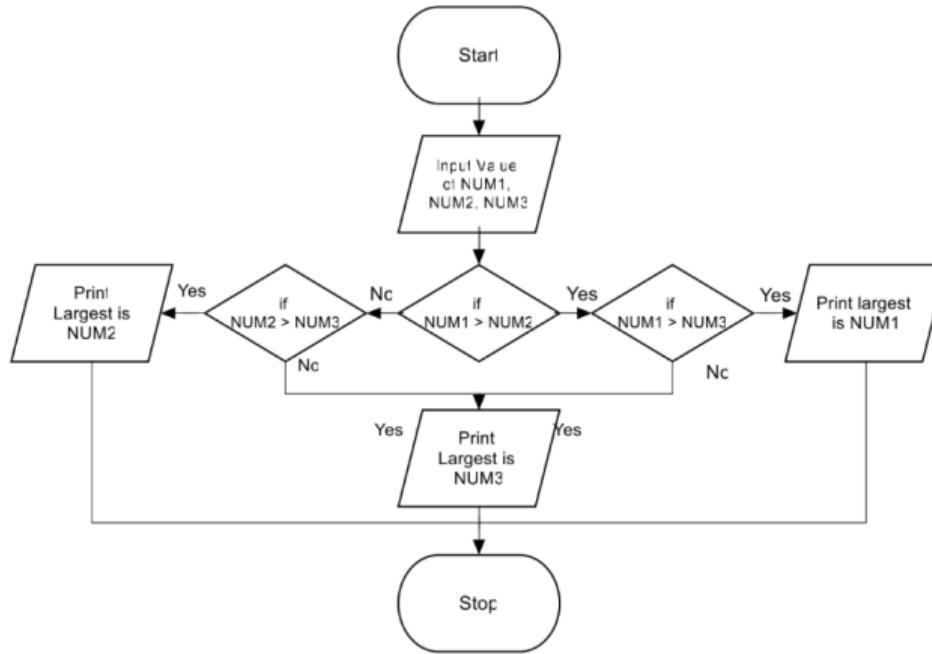


Figura 2: Diagrama de flujo.

Ejemplo 4: Raíces de una Ecuación Cuadrática

Descripción

Este es un problema más avanzado que requiere una estructura de decisión múltiple para manejar los tres casos posibles basados en el valor del discriminante ($b^2 - 4ac$): dos raíces reales, una raíz real doble, o raíces complejas.

Ejemplo 4: Raíces de una Ecuación Cuadrática

Algorithm 3 Raíces de Ecuación Cuadrática

```
1: INICIO
2: LEER A, B, C
3: ASIGNAR DISC ← (B*B) - (4*A*C)
4: if DISC > 0 then
5:   X1 ← (-B + RAIZ(DISC)) / (2*A)
6:   X2 ← (-B - RAIZ(DISC)) / (2*A)
7:   ESCRIBIR 'Raíces Reales y Distintas'
8: else if DISC = 0 then
9:   X1 ← -B / (2*A)
10:  ESCRIBIR 'Raíces Reales e Iguales'
11: else
12:  ESCRIBIR 'Las raíces son imaginarias.'
13: end if
14: FIN
```

Ejemplo 4: Raíces de una Ecuación Cuadrática

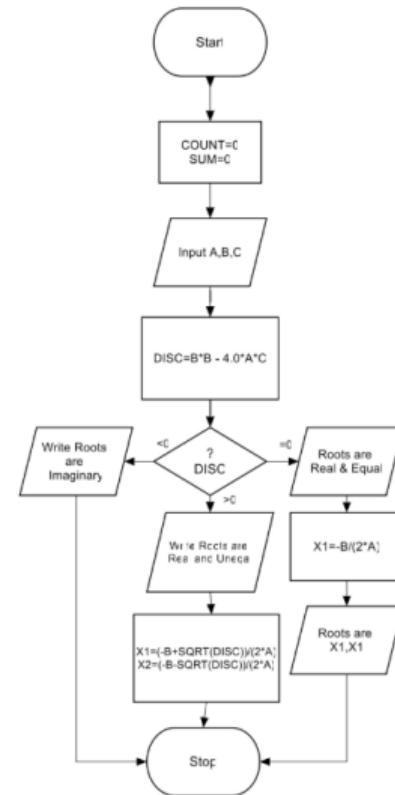


Figura 3. Diagrama de flujo

Ejemplo 4: Máximo Común Divisor (MCD) y Mínimo Común Múltiplo (mcm).

Descripción

Este algoritmo aborda un problema clásico de la teoría de números. Primero, calcula el Máximo Común Divisor (MCD) de dos números utilizando el eficiente **algoritmo de Euclides**, que se basa en la propiedad de que el MCD de dos números no cambia si el número más grande es reemplazado por su diferencia con el número más pequeño. La implementación moderna usa el residuo de la división. Una vez obtenido el MCD, el Mínimo Común Múltiplo (mcm) se calcula fácilmente con la fórmula:
 $mcm(A, B) = (|A \cdot B|)/MCD(A, B)$.

Ejemplo 4: Máximo Común Divisor (MCD) y Mínimo Común Múltiplo (mcm).

Algorithm 4 MCD y mcm de dos números

```
1: INICIO
2: ESCRIBIR 'Ingrese el primer número (A):'
3: LEER A
4: ESCRIBIR 'Ingrese el segundo número (B):'
5: LEER B
6: ASIGNAR numA ← A
7: ASIGNAR numB ← B
8: while numB ≠ 0 do
9:   ASIGNAR residuo ← numA MOD numB
10:  ASIGNAR numA ← numB
11:  ASIGNAR numB ← residuo
12: end while
13: ASIGNAR mcd ← numA
14: ASIGNAR mcm ← (A * B) / mcd
15: ESCRIBIR 'El MCD es: ', mcd
16: ESCRIBIR 'El mcm es: ', mcm
17: FIN
```

▷ Guardar originales para el mcm

Ejemplo 4: Máximo Común Divisor (MCD) y Mínimo Común Múltiplo (mcm).

