



Facultad de Ciencias
de la **Administración**

TECNICATURA
UNIVERSITARIA EN
**DESARROLLO
WEB**



PROGRAMACIÓN I

Unidad V – Estructuras de Control

Estructuras de control repetitivas/iterativas

Tecnicatura Universitaria en Desarrollo Web

Facultad de Ciencias de la Administración

Universidad Nacional de Entre Ríos

- **Objetivos**

- Identificar las distintas alternativas de las que se dispone para controlar el flujo de ejecución de programas.
- Entender como hacer combinaciones de las mismas.
- Comprender cómo pueden diseñarse/documentarse algoritmos a través de diagramas.

- **Temas a desarrollar:**

- Ejecución Secuencial de sentencias.
- Estructuras de control condicionales: if, elif, else, match.
- Estructuras de control iterativas: while, for, for in range, break, continue. Análisis de eficiencia.
- **Diagramas de flujo.**

Representación de algoritmos




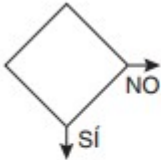
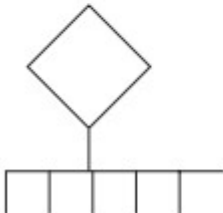
- Para **representar un algoritmo** debemos utilizar algún método que permita **independizar** dicho algoritmo del **lenguaje de programación elegido**.
- Ello permitirá que un algoritmo pueda ser codificado indistintamente en cualquier lenguaje.
- Para conseguir este objetivo se precisa que el algoritmo sea representado gráfica o numéricamente, de modo que la descripción pueda servir fácilmente para su transformación en un programa, es decir, su codificación.
- Los métodos usuales para representar un algoritmo son:
 - 1) Diagrama de flujo,
 - 2) Pseudocódigo,
 - 3) Diagrama Nassi-Schneiderman,
 - 4) Lenguaje coloquial,
 - 5) Fórmulas.

Definición

- Un **diagrama de flujo** (en inglés *flowchart*) es una técnica de representación de algoritmos.
 - Si bien los diagramas de flujo son antiguos, son la técnica más utilizada a la hora de diseñar y documentar.
- Un **diagrama de flujo** utiliza **símbolos** (cajas) estándar, normalizados por **ANSI (American National Standards Institute)**, y describe los pasos de un algoritmo escritos en esas cajas unidas por flechas, denominadas **líneas de flujo**, que indican la secuencia en que se debe ejecutar.

Símbolos

- Aquí vemos algunos símbolos más comunes de los diagramas de flujo:

Símbolos principales	Función
	Terminal (representa el comienzo, “inicio”, y el final, “fin” de un programa. Puede representar también una parada o interrupción programada que sea necesario realizar en un programa.
	Entrada/Salida (cualquier tipo de introducción de datos en la memoria desde los periféricos, “entrada”, o registro de la información procesada en un periférico, “salida”.
	Proceso (cualquier tipo de operación que pueda originar cambio de valor, formato o posición de la información almacenada en memoria, operaciones aritméticas, de transferencia, etc.).
	Decisión (indica operaciones lógicas o de comparación entre datos —normalmente dos— y en función del resultado de la misma determina cuál de los distintos caminos alternativos del programa se debe seguir; normalmente tiene dos salidas —respuestas SÍ o NO— pero puede tener tres o más, según los casos).
	Decisión múltiple (en función del resultado de la comparación se seguirá uno de los diferentes caminos de acuerdo con dicho resultado).

Símbolos (2)



Conector (sirve para enlazar dos partes cualesquiera de un ordinograma a través de un conector en la salida y otro conector en la entrada. Se refiere a la conexión en la misma página del diagrama).



Indicador de dirección o línea de flujo (indica el sentido de ejecución de las operaciones).



Línea conectora (sirve de unión entre dos símbolos).



Conector (conexión entre dos puntos del organigrama situado en páginas diferentes).



Llamada a subrutina o a un proceso predeterminado (una subrutina es un módulo independientemente del programa principal, que recibe una entrada procedente de dicho programa, realiza una tarea determinada y regresa, al terminar, al programa principal).



Pantalla (se utiliza en ocasiones en lugar del símbolo de E/S).



Impresora (se utiliza en ocasiones en lugar del símbolo de E/S).



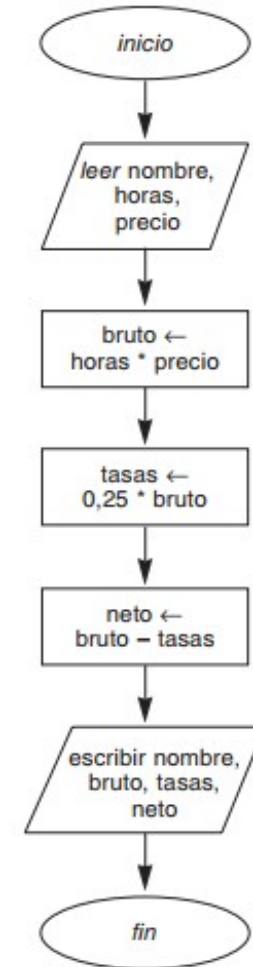
Teclado (se utiliza en ocasiones en lugar del símbolo de E/S).



Comentarios (se utiliza para añadir comentarios clasificadores a otros símbolos del diagrama de flujo. Se pueden dibujar a cualquier lado del símbolo).

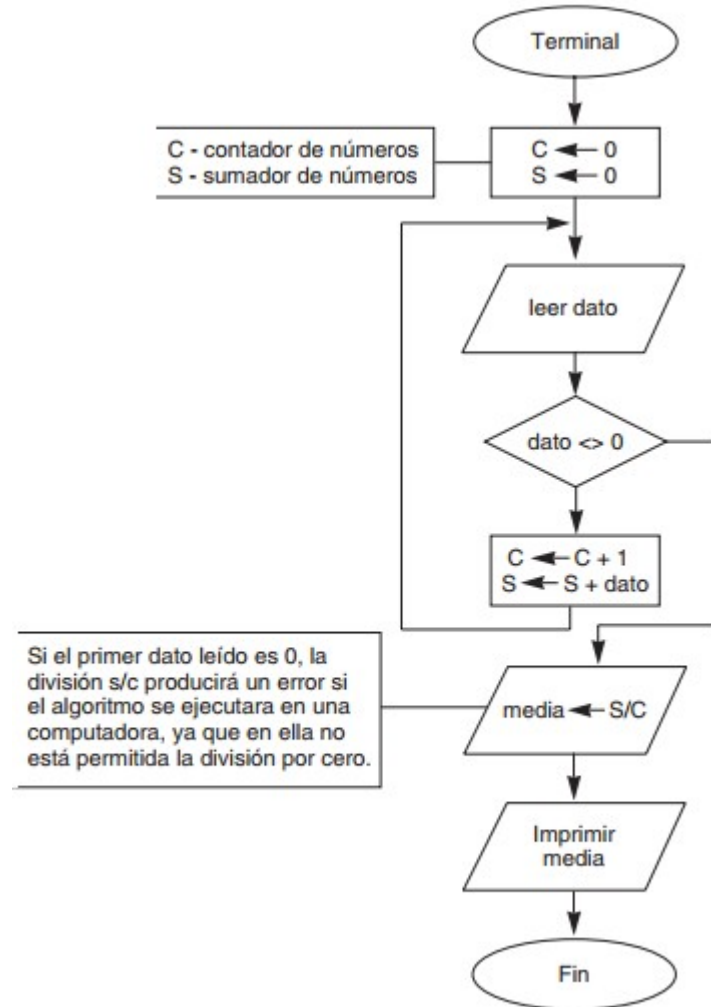
Ejemplo 1

- Este diagrama representa la resolución de un programa que deduce el salario neto de un trabajador a partir de la lectura del nombre, horas trabajadas, precio de la hora, y sabiendo que los impuestos aplicados son el 25 por 100 sobre el salario bruto.



Ejemplo 2

- Para calcular un promedio se necesita sumar y contar los valores.



Bibliografía

- Luis Joyanes Aguilar: ***“Fundamentos de programación, algoritmos, estructura de datos y objetos”***. Ed. Mc Graw Hill. 2008.