

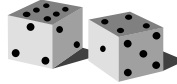
Programación Estructurada. Diagramas de Flujo y Pseudocódigo



Programación Estructurada

Diagramas de Flujo

- Es la representación gráfica de los pasos que deben seguirse para resolver un problema.
- El traducir una descripción narrada a diagrama de flujo agrega claridad y precisión a la descripción de una tarea.
- Además, al elaborar el diagrama de flujo, se descubren situaciones que no habían sido consideradas.



Programación Estructurada

Diagramas de Flujo

- En la elaboración de éstos, la simbología juega un papel muy importante, ya que debe estar adecuada a ciertos estándares, con el fin de que sea entendida por cualquier persona dedicada al campo de la computación.
- En los diagramas de flujo se utilizan figuras geométricas conectadas por líneas.
- Cada una de las figuras representa una etapa en la solución del problema; dentro de ellas se anotan indicaciones. Las líneas señalan el flujo de la información.

Programación Estructurada

Diagramas de Flujo

- En la actualidad se emplean poco, pero resultan muy útiles cuando se comienza en el estudio de la programación.
- El problema con los diagramas de flujo es que a medida que crece la complejidad de las proposiciones, también crece el detalle con el que hay que dibujarlas.
- Esto llegaba a convertirlos en figuras fraccionadas (pues de otro modo no cabrían en la hoja), y difíciles de seguir y entender.

Programación Estructurada

Diagramas de Flujo

- El equivalente simbólico de la flecha se llama goto (ir a) y está, en términos generales, excluido de la programación moderna debido a su poder "deestructurante" y caótico sobre los programas (aunque si se usa con cuidado puede llegar a ser de alguna utilidad).
- Los símbolos utilizados han sido normalizados por el Instituto Norteamericano de Normalización (ANSI) y los más frecuentes utilizados son:



Programación Estructurada

Diagramas de Flujo



INICIO O FIN DE PROCESO:

Indica el inicio o el fin de un Diagrama de Flujo. Dentro de la figura se debe escribir "inicio" o fin; según sea el caso.



ACCIONES U OPERACIONES:

Se utilizan para señalar las actividades, los pasos o las instrucciones en forma secuencial



ENTRADA / SALIDA DE INFORMACION:

Representa la entrada y salida de datos en la computadora.

Programación Estructurada

Diagramas de Flujo



LÍNEAS DE FLUJO:

Indican el sentido o dirección que lleva el diagrama de flujo desde su inicio hasta su fin.



DECISION:

Permite decidir entre 2 opciones o caminos a seguir.



CONECTOR :

Indica la continuidad del Diagrama De Flujo en una misma página. Dentro de la circunferencia se anota un número o una letra.

Programación Estructurada

Diagramas de Flujo



Salida (impresión):

Indica un resultado mostrado como consecuencia del proceso llevado a cabo.



CICLO REPETITIVO (for):

Indica la utilización de una estructura repetitiva



CONECTOR DE PAGINA:

Indica la continuación del diagrama de flujo de una página a otra. Se debe especificar con letra o número esta secuencia.

Programación Estructurada

Diagramas de Flujo



Reglas Básicas para la construcción de DF

- Todo diagrama debe tener un inicio y un fin
- Las líneas de conexión o de flujo deben ser siempre rectas, si es posible verticales y horizontales nunca cruzadas o inclinadas; para conseguir lo anterior es necesario apoyarse en conectores.
- Las líneas que enlazan los símbolos entre sí deben estar todas conectadas.

Programación Estructurada

Diagramas de Flujo

- Se deben dibujar todos los símbolos de modo que se pueda seguir el proceso visualmente de arriba hacia abajo (diseño de top-down) y de izquierda a derecha.
- Realizar un gráfico claro y equilibrado
- Evitar la terminología de un lenguaje de programación o máquina.

Programación Estructurada

Diagramas de Flujo

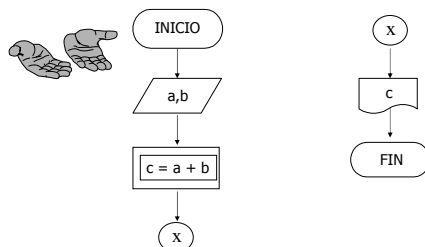
- Utilizar comentarios al margen (si es necesario) para que éste sea entendible por cualquier persona que lo consulte.
- A cada bloque o símbolo se accede por arriba y/o por la izquierda y se sale por abajo y/o por la derecha.
- Si el diagrama abarca más de una hoja es conveniente enumerarlo e identificar de donde viene y a donde se dirige.

Programación Estructurada

Diagramas de Flujo

• Ejemplo de un Diagrama de Flujo:

- **PROBLEMA:** Elaborar un programa que calcule la sumatoria de 2 números:

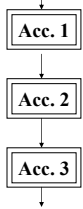


Programación Estructurada

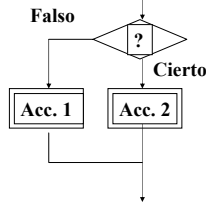
Diagramas de Flujo

Estructuras de control en Diagramas de Flujo

Secuencial



Alternativa

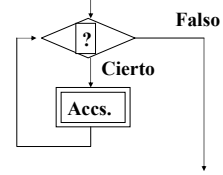


Programación Estructurada

Diagramas de Flujo

Estructuras de control en Diagramas de Flujo

Repetitiva

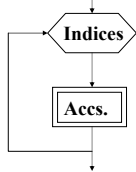


Programación Estructurada

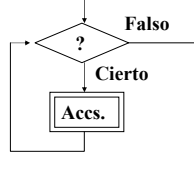
Diagramas de Flujo

Estructuras Repetitivas en Diagramas de Flujo

For



While

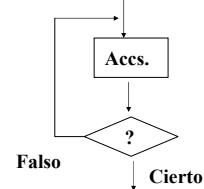


Programación Estructurada

Diagramas de Flujo

Estructuras Repetitivas en Diagramas de Flujo

Repeat



Programación Estructurada

Pseudocódigo

- Los algoritmos se deben describir en un lenguaje que se parezca más al lenguaje utilizado para escribir programas de computador.
- Es decir, un *lenguaje de pseudoprogramación*, una imitación del código de las computadoras al cual se le conoce como **pseudocódigo**.



Programación Estructurada

Pseudocódigo

- El **pseudocódigo** se concibió para superar las dos principales desventajas del diagrama de flujo:
 - el diagrama de flujo es lento de crear y
 - difícil de modificar sin un nuevo redibujo.
- Por otra parte el pseudocódigo es más fácil de utilizar ya que es similar al español -o al inglés, catalán, alemán o frances, dependiendo del caso.



Programación Estructurada

Pseudocódigo

- Al contrario que los lenguajes de programación de alto nivel, como pascal o Basic, no existe un conjunto de reglas que definan con precisión lo que es y lo que no es pseudocódigo.
- Varía de un programador a otro y de que tan próxima sea la descripción al lenguaje de programación.
- El pseudocódigo es una mezcla de lenguaje natural y símbolos, términos y otras características comunmente utilizadas en uno o más lenguajes de alto nivel.

Programación Estructurada

Pseudocódigo

- Tipicamente se encuentran las características en diferentes pseudocódigos que se pueden encontrar en libros de texto de programación.
- El pseudocódigo requiere de ciertos símbolos privilegiados que ya tienen significado preciso y establecido de antemano.
- A tales indicadores del pseudocódigo se les conoce como "**palabras clave**" (keywords).

Programación Estructurada

Pseudocódigo

- Es necesario que exista una palabra clave para la selección y otra para la iteración condicional, así como para las instrucciones adicionales y otras estructuras de control.
- Por ejemplo, la palabra "**escribe**" es una palabra clave que ya tiene significado predefinido, a diferencia de la palabra ALFA, que es una variable libre.



Programación Estructurada

Pseudocódigo

- Se pretenderá uniformizar el pseudocódigo utilizando la siguiente simbología :

Intrucción en Inglés

Begin
End
Read / Input
Write / Print
If ____ then
Else
For
While
Repeat
Until ____

Pseudocódigo en español

Inicio
Fin
Leer / Entrada de Datos
Escribir / Salida de Datos
Si ____ entonces
Sino / Caso Contrario
Desde
Mientras
Repetir
Hasta ____

Programación Estructurada

Pseudocódigo

- El algoritmo comienza con la palabra **inicio** y termina con la palabra **fin**. Entre estas palabras, se escribe una instrucción (acción) por línea o se separan con un punto y coma.
- La Línea encerrada entre llaves ({...}) se denomina *comentario* : es una información al lector del programa y no realiza ninguna instrucción ejecutable, sólo tiene efectos de documentación interna del programa.
- La asignación se llevara a cabo mediante el signo =
EJEMPLO: A = 10, a la variable A se le asigna el valor de 10.

Programación Estructurada

Pseudocódigo

- Por lo tanto, el **Pseudocódigo** a utilizar incluirá:
 - Nombre del programa
 - Sección de declaraciones (variables y constantes)
 - Algoritmo



Programación Estructurada

Pseudocódigo

• Ejemplo en Pseudocódigo:

- **PROBLEMA:** Elaborar un programa que calcule la sumatoria de 2 números:

Programa Suma dos numeros

Variables N1, N2, S enteros

Inicio

Leer N1

Leer N2

S = N1 + N2

Escribir S

Fin

- Donde :
- N1 = Variable que recibe el primer número
- N2 = Variable que recibe el segundo número
- A la variable S se le asigna la suma de los dos números

Programación Estructurada

Ejercicios

- Se desea obtener una Tabla con las depreciaciones acumuladas y los valores reales de cada año de un automóvil comprado en \$1.800.000 pesos en el año 1992, durante los seis años siguientes; suponiendo un valor de recuperación de \$120.000. Realizar el análisis del problema, conociendo la fórmula de la depreciación anual constante D para cada año de vida útil.

$$D = \frac{\text{costo} - \text{valor de recuperación}}{\text{vida útil}}$$

Programación Estructurada

Ejercicios

Año	Depreciación	Depreciación Acumulada	Valor anual
1 (1992)	280.000	280.000	1.520.000
2 (1993)	280.000	560.000	1.240.000
3 (1994)	280.000	840.000	960.000
4 (1995)	280.000	1.120.000	680.000
5 (1996)	280.000	1.400.000	400.000
6 (1997)	280.000	1.680.000	120.000



Programación Estructurada

Ejercicios

Entradas

- Costo original
- Vida útil
- Valor de Recuperación



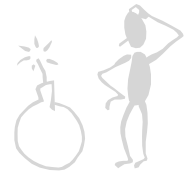
Procesos

- Depreciación en cada año
- Cálculo de la depreciación acumulada
- Cálculo del valor del automovil en cada año



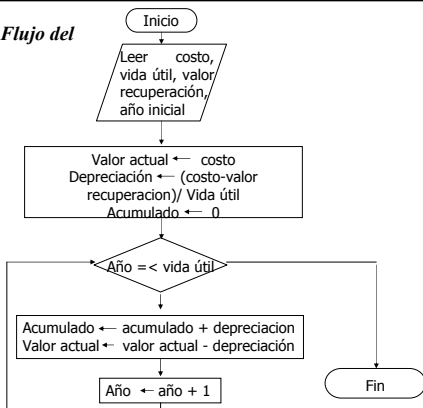
Salidas

- Depreciación anual por año
- Depreciación acumulada en cada año
- Valor del automóvil en cada año



Programación Estructurada

Diagrama de Flujo del Ejercicio



Programación Estructurada

Pseudocódigo del Ejercicio

Calculo de Depreciacion

Introducir Costo

Vida útil

Valor de Recuperacion

Imprimir Cabeceras de tabla

Establecer el valor inicial del Año

Calcular Depreciación

Mientras valor año ≤ vida util **hacer**

Calcular depreciacion acumulada

Calcular valor actual

Imprimir una linea de la tabla con los valores calculados

Incrementar el valor del año en uno

Fin de mientras



Programación Estructurada

Ejercicios

- Realice los siguientes ejercicios en Diagrama de Flujo y Pseudocódigo:
- Calcular el exponencial de un número (a^b), considerando todos los casos posibles:
 - Ingreso de números negativos
 - Ingreso de valores igual a 0 ($a=0$, $b=0$)
- Calcule e imprima los primeros n números primos, considere que n es ingresado por el usuario.

Ejercicios

- Se pide determinar el mayor de tres números ingresados por el usuario. Considere que pueden ser iguales
- Calcule el factorial de un número ingresado por el usuario.

