



ANÁLISIS Y DESARROLLO DE SOFTWARE

FUNDAMENTACIÓN DE LA TEORÍA DE SISTEMAS Y PROCESOS

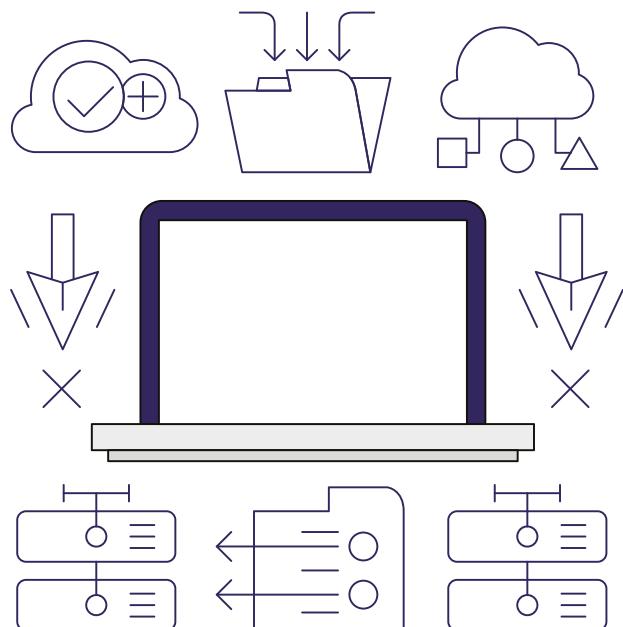
REPRESENTACIÓN DE LOS ALGORITMOS

5.5 Identificadores

Los identificadores son nombres que se dan a las variables, constantes, acumuladores y contadores para así poder diferenciarlos. Para asignar los nombres se debe tener en cuenta lo siguiente:

- Los nombres pueden estar formados por una combinación de letras y números (saldoMes, salario, fecha2, baseTriángulo, etc).
- El primer carácter de un nombre debe ser una letra.
- La mayoría de los lenguajes de programación diferencian las mayúsculas de las minúsculas.
- Los nombres deben ser nemotécnicos, con solo leerlos se puede entender lo que contienen. Deben ser muy descriptivos; no utilizar abreviaturas, a menos que se justifique plenamente.
- No utilizar caracteres reservados (% , + , / , > , etc).
- No utilizar palabras reservadas por los lenguajes de programación.
- Para cumplir con convenciones ampliamente utilizadas (Jiménez, 2002), los nombres de procedimientos, variables y constantes deben empezar con minúscula. Ejemplo, fecha, suma, etc. Si es un nombre compuesto por varias palabras, cada una de las palabras (con excepción de la primera) debe empezar con mayúscula. Ejemplo: fechahnicial, baseTriángulo, etc.

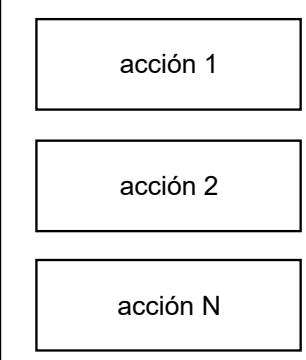
6. Estructuras algorítmicas o de programación



Las estructuras algorítmicas están creadas para orientarnos en la forma de diseñar algoritmos, cada una de ellas representa cierto concepto que permite lograr encontrar la solución del problema, entre ellas tenemos:

6.1 Estructura secuencial

Es una de las estructuras más sencillas, conocida también como estructura lineal y se compone de instrucciones que deben ejecutarse secuencialmente. Ejemplo:

DIAGRAMA DE FLUJO	PSEUDOCÓDIGO
	Inicio <acción1> <acción2> * * * <acciónN> Fin

6.2 Correspondencia de pseudocódigo a diagrama de flujo.

Consiste en llevar un algoritmo a un diagrama de flujo.

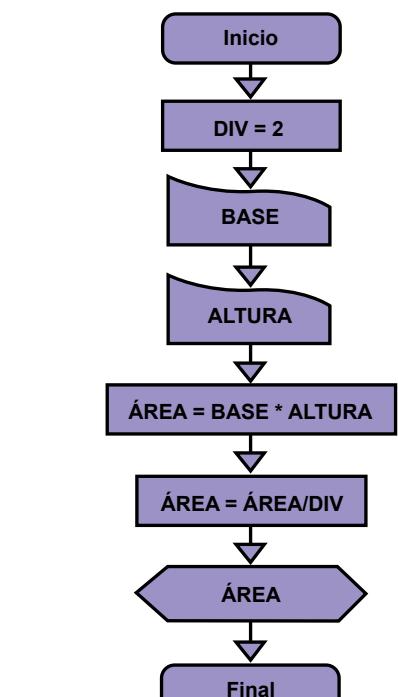
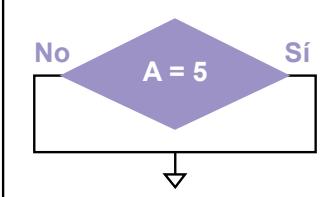
DIAGRAMA DE FLUJO	PSEUDOCÓDIGO
	Paso 1: inicio. Paso 2: asignar el número 2 a la constante DIV. Paso 3: saber la base del triángulo y guardarla en la variable “BASE”. Paso 4: saber la altura del triángulo y guardarla en la variable “ALTURA”. Paso 5: guardar en la variable “ÁREA” el valor de BASE * ALTURA Paso 6: guardar en la variable “ÁREA” la división entre el “ÁREA” sobre la variable “DIV”. Paso 7: mostrar el valor de área. Paso 8: fin.

Figura 14. Diagrama de Flujo con Pseudocódigo.

6.3 Estructuras de decisión

Son utilizadas para tomar decisiones lógicas, llamadas también estructuras selectivas o alternativas, en ellas se evalúa una condición y en función del resultado de la misma se realiza una opción u otra.

Ejemplo:

DIAGRAMA DE FLUJO	PSEUDOCÓDIGO
	Inicio Si condición entonces Fin - si Fin condición

Las estructuras de decisión pueden ser:

6.3.1 Simples.

Ejecuta una determinada acción cuando se cumple una determinada condición (llamada si-entonces). La selección si-entonces evalúa la condición y si la condición es verdadera, entonces ejecuta la acción; si la condición es falsa entonces no hace nada. Ejemplo:

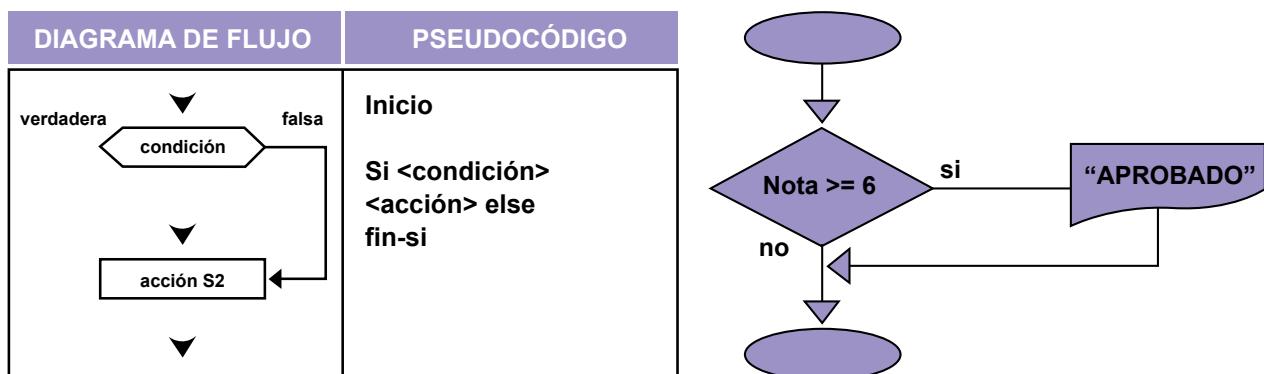
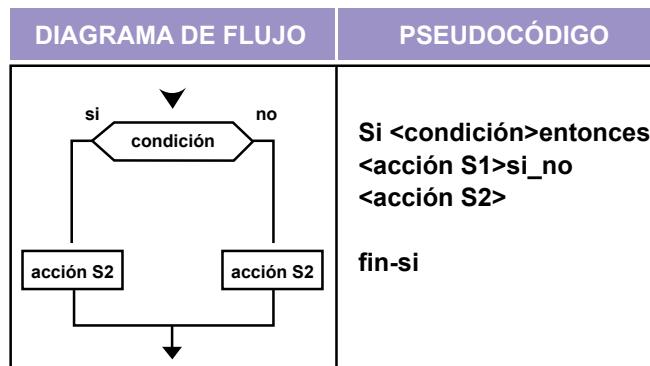


Figura 15. Estructura simple de un condicional.

En el ejemplo anterior existe una estructura de decisión simple dado que si la Nota es superior a 6 es considerado el aprendiz como un logro aprobado.

6.3.2 Dobles:

La estructura anterior es muy limitada y normalmente se necesitará una estructura que permita elegir entre dos opciones o alternativas posibles, en función del cumplimiento o no de una determinada condición. **Ejemplo:**

DIAGRAMA DE FLUJO	PSEUDOCÓDIGO
	Si <condición>entonces <acción S1> si_no <acción S2> fin-si

En el siguiente ejemplo dados 3 números definir cuál de ellos es mayor e imprimirlo. (Véase Figura 16).

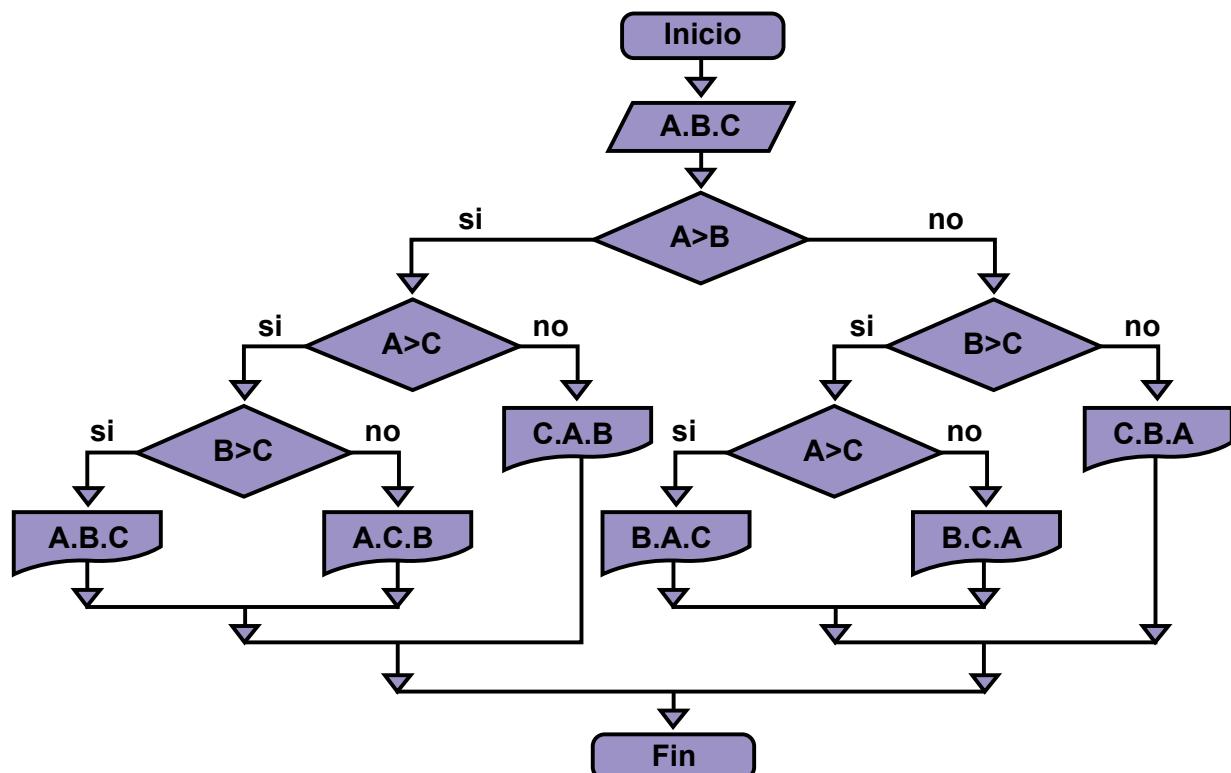


Figura 16. Estructura condicional anidada.

6.4 Estructuras cíclicas y/o repetitivas

Un tipo muy importante de estructura es el algoritmo necesario para repetir una o varias acciones un número determinado. Las estructuras que repiten una secuencia de instrucciones se denominan bucles, y se llama iteración al hecho de repetir la ejecución de una secuencia de acciones.

Y así sucesivamente para cada número de la lista. En otras palabras, el algoritmo repite muchas veces las acciones.

Las dos principales preguntas a realizarse en el diseño de un bucle son: ¿qué contiene el bucle? y ¿cuántas veces se debe repetir?

6.4.1 Estructura para

En muchas ocasiones se conoce de antemano el número de veces que se desean ejecutar las acciones de un bucle. Esta estructura permite ejecutar una o varias instrucciones un determinado de veces finita, fija. (Véase Figura 17).

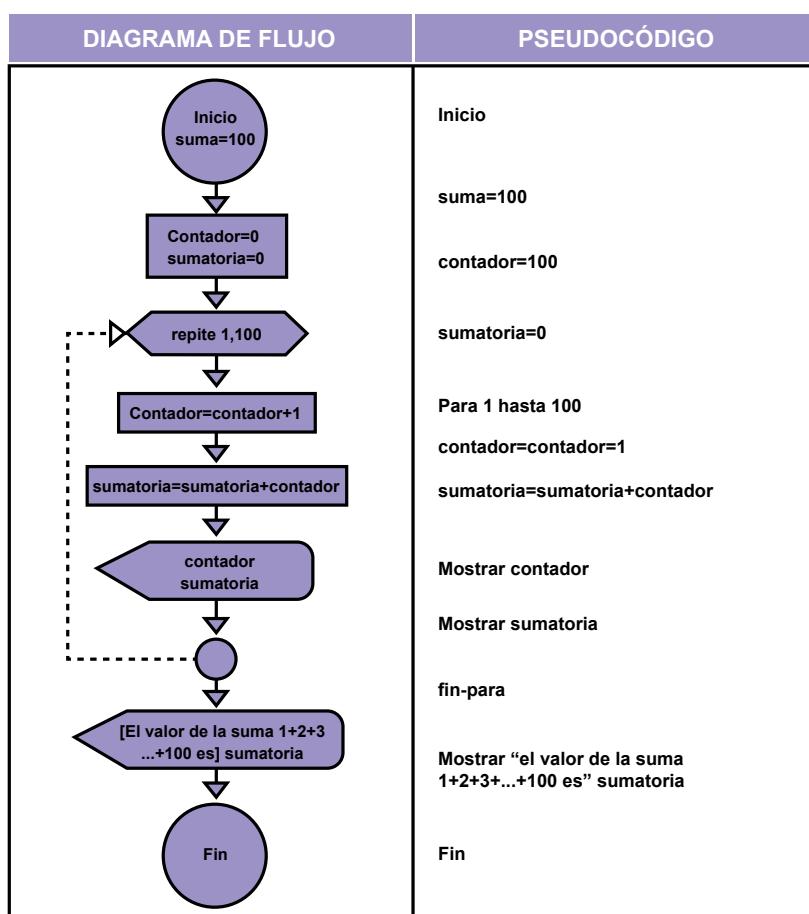


Figura 17. Diagrama con estructura cíclica y/o repetitiva

6.4.2 Estructura mientras

Se llama Mientras a la estructura algorítmica que se ejecuta mientras la condición evaluada resulte verdadera. Se evalúa la expresión booleana y, si es cierta, se ejecuta la instrucción especificada, llamada el cuerpo del bucle. Entonces se vuelve a evaluar la expresión booleana, y si todavía es cierta se ejecuta de nuevo el cuerpo. Este proceso de evaluación de la expresión booleana y ejecución del cuerpo se repite mientras la expresión sea cierta.

Cuando se ejecuta la instrucción mientras, la primera cosa que sucede es que se evalúa la condición (una expresión booleana). Si se evalúa falsa, ninguna acción se toma y el programa prosigue en la siguiente instrucción del bucle. Si la expresión booleana es verdadera, entonces se ejecuta el cuerpo del bucle, después de lo cual se evalúa de nuevo la expresión booleana. Este proceso se repite una y otra vez mientras la expresión booleana (condición) sea verdadera. (Véase Figura 18).

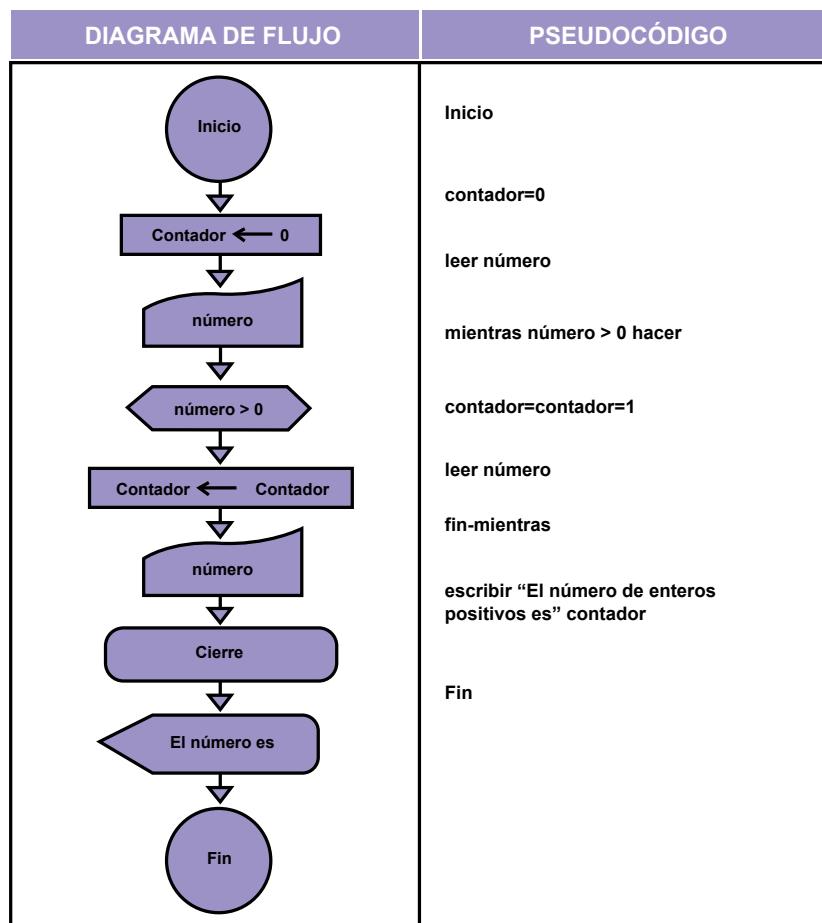


Figura 18. Estructura con la función Mientras.

6.4.3 Estructura Repita

La estructura cíclica REPITA, al igual que la estructura cíclica mientras, se ejecuta un número indeterminado de veces, estas dos estructuras tienen un comportamiento similar, presentando su principal diferencia en el lugar de estructura donde se evalúa la condición, dado que la estructura MIENTRAS evalúa la condición del ciclo al inicio del mismo y la estructura REPITA lo hace al final del mismo, de este modo, en la estructura cíclica REPITA, el programador garantiza que el ciclo se ejecuta al menos una vez. (Véase Figura 19).

