

ANÁLISIS Y DESARROLLO DE SOFTWARE

LÓGICA PROPOSICIONAL

REPRESENTACIÓN DE LOS ALGORITMOS

1. Diseño de un algoritmo mediante diagrama de flujo

Los algoritmos pueden representarse de varias maneras, entre ellas está la representación de diagramas de flujo. La representación en diagramas de flujo tiene como objetivo seguir paso a paso la solución de un problema mediante símbolos.

1.1 Definición Diagrama de flujo.

Un diagrama de flujo es un conjunto secuencial de figuras geométricas estándar conectadas lógicamente entre sí para dar solución a un problema específico, cada figura tiene un significado propio.

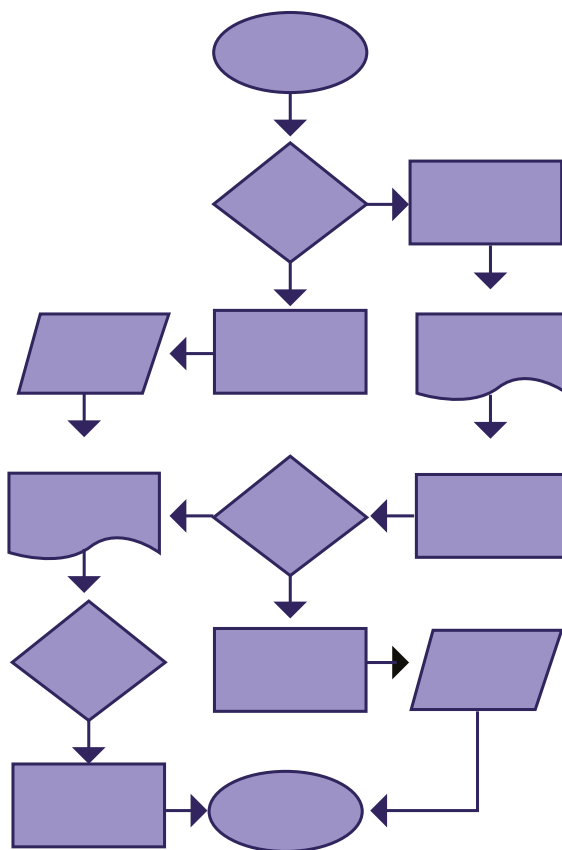
La secuencia lógica se da por medio de flechas llamadas líneas de flujo que indican el flujo lógico del algoritmo. Al ser un diagrama gráfico facilita la visión de la ejecución del algoritmo.

La simbología utilizada en estos diagramas ha sido estandarizada por las organizaciones ANSI (American National Institute) y por ISO (International Standard Organization).

SÍMBOLO	SIGNIFICADO	EXPLICACIÓN
	Paso de tipo operación	Representa cualquier tarea del proceso que lleve implícita una acción física o intelectual (excepto las de inspección o almacenaje).
	Paso de inspección	Se corresponde con tareas de verificación del trabajo realizado en determinada actividad del proceso. Sus acciones más comunes son: clasificar, observar, supervisar, auditar, probar, revisar, verificar, entre otras.
	Paso de decisión	Representa cualquier punto de decisión. Siempre tendrá al menos dos salidas.
	Paso de almacenaje	Se corresponde con una etapa del proceso que sitúa un producto, información o servicio en una zona de conservación (archivo, almacén o refrigerador) o posición (cola) para utilizarlo o proporcionar el servicio más adelante.
	Paso de demora	Corresponde a actividades que implican un retraso o pausa en el flujo del proceso.
	Línea de flujo	Muestra la dirección y sentido del flujo del proceso y representa el progreso de los pasos en la secuencia.
	Documento	Se utiliza con el objetivo de especificar los documentos confeccionados, corregidos o consultados en cada etapa.
	Conector de tareas	Se utiliza el caso de que el diagrama no se pueda hacer en una sola hoja.

Figura 10. Símbolos del Diagrama de Flujo.

Existen herramientas software que permiten realizar los gráficos mencionados anteriormente, una de las más utilizadas es DFD, día, entre otros, los cuales son útiles y de libre acceso en la web.



1.2 Reglas para la elaboración de un diagrama de flujo:

- Los diagramas se deben realizar de arriba hacia abajo y de izquierda hacia derecha.
- Los símbolos de inicio y final deben aparecer solo una vez
- La ejecución de un programa siempre empieza en la parte superior del programa.
- La dirección del flujo se debe representar por medio de flechas.
- Todas las líneas de flujo deben llegar a un símbolo o a otra línea.
- Se deben inicializar las variables que se utilicen o permitir la asignación de valores mediante la consulta a un usuario.

1.2.1 Diseño de un algoritmo mediante pseudocódigo

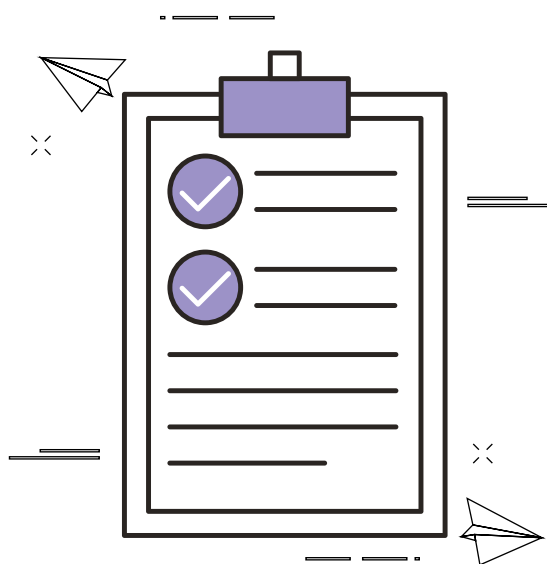
La representación de pseudocódigo sigue paso a paso la solución de un problema con lenguaje natural, pero recuerde que debe realizar la metodología resolución de problemas para poder iniciar su algoritmo con pseudocódigo. Podría recordarse la imagen de resolución de problemas explicada anteriormente.

Paso 1: inicio

Paso 2: los procesos que se van realizar....

Paso 3: si hay más pasos se debe continuar...

Paso n: fin



Ejemplo1:

Se retoma el ejemplo de José David para llevarlo en forma de algoritmo: José David, se encuentra ahorrando para comprar la maleta que vale 105.000 pesos. En su casa le han dado para sus gastos 24.000 pesos durante 4 semanas. Por atender la tienda recibió \$36.000 pesos. Su hermano Juan Antonio ganó 23.000 pesos por lavar la piscina y cuidar la tienda. ¿José David tiene ahorrado el dinero suficiente para comprar la maleta o aún le falta?

Paso 1: inicio

Paso 2: asignarle el valor del dinero ahorrado por José David a la Variable ValorAhorrado = 24.000+36.000

Paso 3: mostrar el ValorAhorrado, indicando si le alcanza o no para la maleta.

Paso 4: fin.

Ejemplo 2

Se retoma el ejemplo anterior también por continuidad.

Se necesita calcular el área de un triángulo rectángulo cuya Base mide 3 cm, la Altura 4 cm y la Hipotenusa 5 cm.

Paso 1: inicio

Paso 2: indicar que la variable Base tiene un valor de 3 cm

Paso 3: indicar que la variable Altura es de 4 cm

Paso 4: calcular el área $a = (Bases * Altura) / 2$

Paso 5: mostrar el área

Paso 6: fin.

Ejemplo 3

También se pueden resolver problemas cotidianos: por medio de pseudocódigo escriba un algoritmo para poder pasarse los semáforos, una vez usted se encuentra esperando pasar como peatonal.

Paso 1: inicio

Paso 2: ver el color del semáforo

Paso 3: si el semáforo esta en rojo : hay que detenerse

Paso 4: mostrar mensaje hay que detenerse.

Paso 5: si el semáforo esta en amarillo: alistarse para pasar

Paso 6: mostrar mensaje alistarse.

Paso 7: si el semáforo esta en verde: pasarse la calle.

Paso 8: mostrar mensaje puede pasarse y la persona se pasa la calle.

Paso 9: fin.

1.2.2 Expresar el algoritmo en un lenguaje de programación

Una vez el algoritmo este diseñado en diagrama de flujo o en lenguaje natural (pseudocódigo), se pone en marcha la “traducción” de este a un lenguaje de programación específico.

Existen diferentes tipos de lenguajes que permiten “traducir” el algoritmo para que pueda ser entendido por el computador entre los cuales son java, c#, Visual Basic, entre otros, los cuales son tratados específicamente más adelante.



2. Solución de problemas con diagramas de flujo y lenguaje natural (Pseudocódigo).

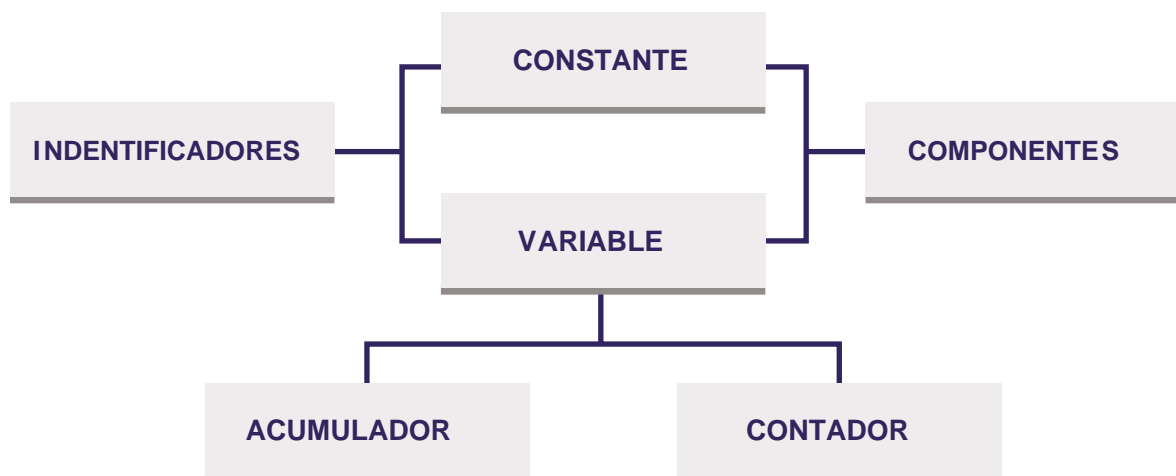


Figura 11. Estructura con identificadores.

Recuerde cuando se plantee un problema ya sabe que puede resolverlo por medio de diagrama de flujo o por lenguaje natural.

2.1 Variables.

Cuando se inicia con el análisis del problema se identifican los datos iniciales, estos datos se estructuran o se definen como variables; en ellas se pueden almacenar valores y son nombradas con identificadores, es decir nombres para poder identificarlas dentro del algoritmo. Por ejemplo, si en el problema de calcular el área de un triángulo tengo como datos iniciales la base y la altura, estas dos anteriores serían llamadas variables y sus nombres serán “BASE” y “ALTURA”.

Un aspecto importante de las variables es que pueden cambiar su valor durante la ejecución del algoritmo.

Se debe tener en cuenta que una variable puede ser declarada, asignada o solicitada mediante un algoritmo para poder ser utilizada.

2.1.1 Declaración de una variable

Para utilizar una variable tanto en pseudocódigo como en diagrama de flujo es necesario siempre declararla, es decir indicarle al algoritmo que va a utilizar una variable por ejemplo `Nom_Persona`, quiere decir que usted necesita una variable que almacene los nombres de las personas. Ejemplo `Nom_Persona=Nombre de las personas`.

2.1.2 Asignación de valor a una variable

Luego de declararla puede usted necesitar que esa variable inicie por defecto con un valor por ejemplo Saldo = 60000, quiere decir que utilizará la variable saldo pero que esta inicializada con un valor de 60000, puede inicializar una variable según el valor que necesite para solucionar el problema.

Asignarle un valor a una variable, constante, acumulador o contador, expresiones complejas o simples, por ejemplo.

Variable = expresión

Nom_Persona = Rita

Saldo = 4000

Area = (Base*altura)

2.1.3 Solicitar una variable

Cuando se necesita que el valor de la variable pueda ser ingresado por un usuario al computador, entonces debe solicitarse este valor de la variable.

2.2 Constantes

Almacenan datos al igual que las variables, pero su gran particularidad es que sus datos no cambian durante la ejecución del algoritmo, es decir siempre el valor de la constante va ser el mismo.

Las constantes se deben declarar e inicializar.


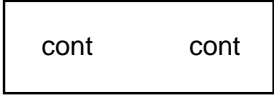
DIAGRAMA DE FLUJO	PSEUDOCÓDIGO
Declarar una constante e inicializarla <div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> Máximo 100 </div>	Declarar una constante e inicializarla Constante Máximo 100

Ejemplo de constante.

2.3 Contadores

Como su palabra lo dice permiten contar, para poder utilizar un contador es necesario inicializarlo en un valor y luego incrementar su valor de una manera constante para permitir realizar el conteo.

Es una variable cuyo valor se incrementa o decrementa en una cantidad constante cada vez que se produce un determinado suceso o acción. Los contadores se utilizan con la finalidad de contar sucesos o acciones internas de un bucle; deben realizar una operación de inicialización y posteriormente las sucesivas de incremento o decremento del mismo. La inicialización consiste en asignarle al contador un valor. Se situará antes y fuera del bucle.

DIAGRAMA DE FLUJO	PSEUDOCÓDIGO
Declarar una constante e inicializarla 	Declarar una constante e inicializarla cont = 0
Utilizar contador 	Utilizar un contador cont = cont + 1

Representación:

<nombre del contador> = <nombre del contador> + <valor constante>

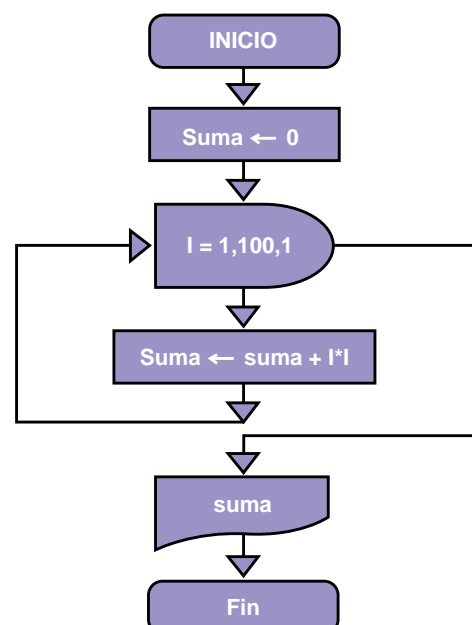
Si en vez de incremento es decremento se coloca un menos en lugar del más.

Ejemplo: $i = i + 1$

2.4 Acumuladores

Como su palabra lo dice permiten acumular el valor de una variable, para poder utilizar un acumulador es necesario inicializarlo en un valor y luego iniciar con la acumulación del valor.

El anterior algoritmo está representado mediante un Diagrama de Flujo y lo que hace es inicializar un valor en 1 hasta 100, hasta imprimir la suma de cada dígito acumulándolo hasta 100. En este caso sería $1+2+3+4+5+...+100$.

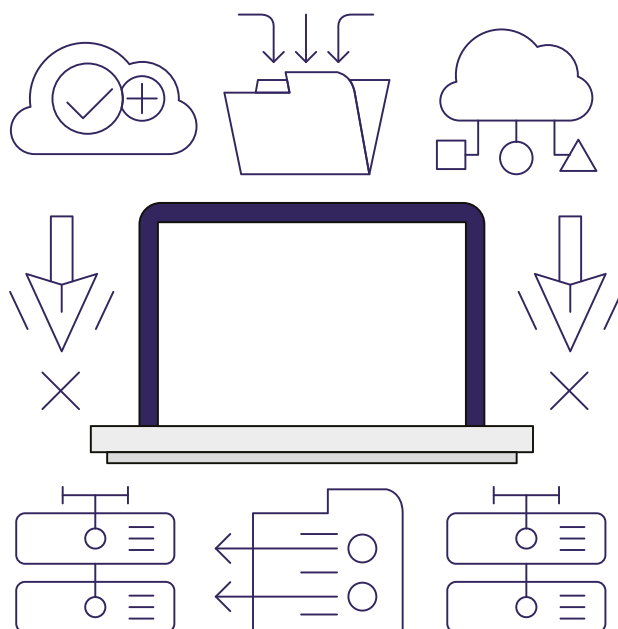


2.5 Identificadores

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: fechaInicial, baseTriángulo, etc.

3. 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: