

ESTRUCTURAS DE SECUENCIA

La primera estructura básica de la programación estructurada es la secuencial. Consiste en una sucesión de instrucciones que se ejecutan en el mismo orden en que fueron escritas, una a continuación de la otra.

Un algoritmo secuencial se expresa como una sucesión de instrucciones que se ejecutan TODAS, una a continuación de la otra, teniendo un único punto de inicio y un único punto de fin. Esas instrucciones pueden ser de distintos tipos:

- Operaciones aritméticas
- Operaciones de asignación
- Ingreso de datos
- Salida / Mostrar información

Almacenamiento de Datos

Antes de poder escribir las instrucciones que forman parte de los algoritmos, debe saber cómo se guardan los datos internamente en la computadora, ya que las instrucciones serán operaciones que se realizarán sobre dichos datos.

Una variable es un espacio de memoria reservado para almacenar un valor que corresponde a un tipo de dato soportado por el lenguaje de programación. Una variable, es representada y usada a través de su identificador (nombre de la variable).

Por ejemplo, se define una variable con el nombre **num**, en la cual se almacena el número 8, **num** es el identificador y ocho es su contenido.

Tipos de Datos

Los tipos de datos indican que es lo que se puede almacenar en una variable. Por ejemplo, un número entero, un carácter, una cadena de caracteres, un número con decimales, etc.

Cada tipo de dato ocupa en la memoria de la computadora una cantidad distinta de bytes. Por lo tanto, para optimizar un programa, por ejemplo, si se necesita trabajar con números pequeños, se puede utilizar una variable con un tipo de dato que permita almacenar números pero hasta cierta cantidad haciendo que esa variable ocupe menos espacio en memoria.

Tipos de Datos en C++

Tipo de Dato	Descripción	Número de bytes	Rango
(signed) short	Entero corto	2	-32768 a 32767
unsigned short	Entero corto sin signo	2	0 a 65535
(signed) int	Entero	4	-2147483648 a +2147483647

unsigned (int)	Entero sin signo	4	0 a 4294967295
(signed) long	Entero largo	4	-2147483648 a +2147483647
unsigned long	Entero largo sin signo	4	0 a 4294967295
(signed) char	Carácter	1	-128 a 127
unsigned char	Carácter sin signo	1	0 a 255
float	Real (Número en coma flotante)	4	Positivos: 3.4E-38 a 3.4E38 Negativos: -3.4E-38 a -3.4E38
double	Real doble(Número en coma flotante de doble precisión)	8	Positivos: 1.7E-308 a 1.7E308 Negativos: -1.7E-308 a -1.7E308
long double	Real doble largo	10	Positivos: 3.4E-4932 a 1.1E4932 Negativos: -3.4E-4932 a -1.1E4932

Identificadores

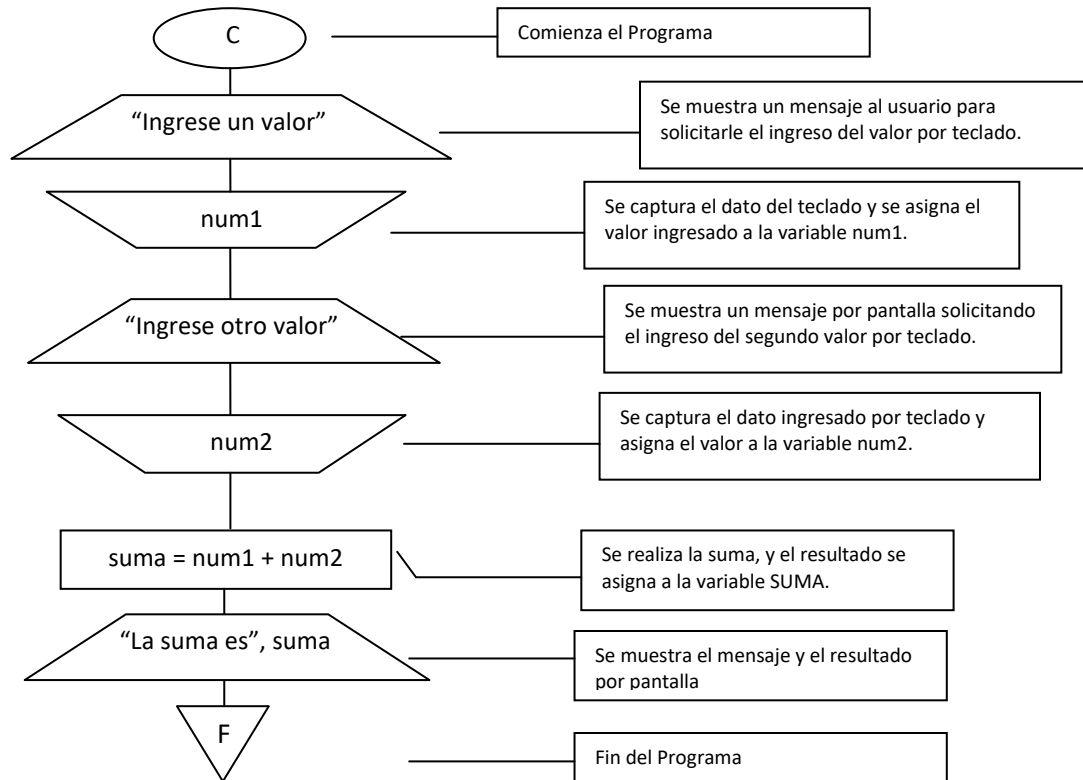
Los nombres de variables o identificadores deben ser nemotécnicos, es decir, que con solo leer el nombre de la variable se pueda entender o determinar con facilidad lo que ella significa o contiene. En el ejemplo anterior la variable con el nombre **num** almacena un número por lo que se ve a simple vista, pero si a esta variable se le hubiese dado el nombre **x** o **ds**, estos nombres pueden significar muchas cosas o, tal vez, no significar absolutamente nada.

Los identificadores tienen ciertas reglas de escritura que van a depender del lenguaje de programación que se esté utilizando. Como regla general, un identificador tiene que cumplir con las siguientes reglas:

- No puede comenzar con números.
- No puede tener espacios.
- No puede tener tildes, ni letra ñ.
- No puede tener símbolos (excepto el guión bajo que es el único permitido).

Ejemplo:

Realizar un programa que permita ingresar dos números enteros por teclado, realice la suma, y muestre por pantalla el resultado. El diagrama para el algoritmo es el siguiente:

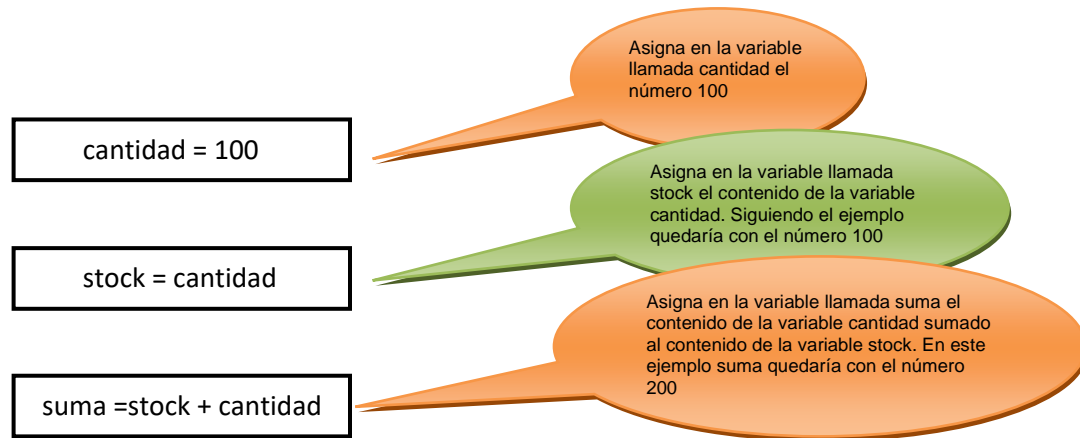
Operación de Asignación

La asignación consiste en guardar un dato en una variable. El dato puede ser:

- Otra variable
- Una constante
- El resultado de una operación matemática

La asignación se realiza de derecha a izquierda, es decir, que a la izquierda se escribe el identificador de la variable en la cual desea guardar el dato, luego se escribe el símbolo = (igual) que indica la operación de asignación, y a la derecha se coloca la constante, variable u operación matemática.

Gráficamente la operación de asignación se representa con un rectángulo. A continuación pueden verse distintos ejemplos de asignación.



Operaciones matemáticas

Sobre los datos se pueden realizar distintas operaciones. En la tabla 2 se detallan las operaciones posibles y el símbolo que se utiliza para expresarlas. A estos símbolos se los denomina operadores matemáticos.

Operadores Matemáticos	
Operación	Operador matemático
Suma	+
Resta	-
Multiplicación	*
División	/
Resto de la división entera	%

Las operaciones se pueden realizar tanto sobre variables (tomando el contenido que tienen almacenado), como sobre constantes.

Los operadores tienen precedencia al igual que en matemática, por ejemplo, el operador suma, separa términos y se ejecuta luego de realizar las operaciones de mayor precedencia como la multiplicación o la división. Muchas veces es necesario cambiar dicha precedencia y, por lo tanto, debemos agrupar las operaciones. Para ello se utilizan paréntesis. En una expresión pueden utilizarse todos los niveles de paréntesis que sean necesarios (NO corchetes). Por ejemplo, en la siguiente instrucción:

$$\text{suma} = 3 + 2 * 5$$

La variable `suma` quedará con el valor 13 ya que primero realiza la multiplicación y luego la suma. En cambio, si la misma instrucción se escribiera de la siguiente manera:

$$\text{suma} = (3 + 2) * 5$$

La variable `suma` quedará con el valor 25 ya que se utilizaron los paréntesis para resolver la suma primero que la multiplicación.