

## Elementos fundamentales de los lenguajes de programación

El estudio de los lenguajes de programación permite comprender cómo se construyen y ejecutan los programas, así como los mecanismos internos que garantizan su funcionamiento. A continuación, se describen los principales elementos que forman parte de estos lenguajes:

### 1. Nombres

En programación, los nombres son identificadores utilizados para referirse a variables, constantes, funciones, clases o cualquier otro objeto definido en el programa.

Cumplen con reglas sintácticas específicas (por ejemplo, no iniciar con números).

Son esenciales para la legibilidad del código.

Están asociados a un tipo de dato y un valor en memoria.

### 2. Marcos de activación

Un marco de activación (activation record o stack frame) es una estructura de datos que se crea en la pila de ejecución cada vez que se invoca un subprograma. Contiene:

Dirección de retorno.

Parámetros pasados al subprograma.

Variables locales.

Información de control necesaria para reanudar la ejecución.

Su correcta administración permite la recursión y el control del flujo de subprogramas.

### 3. Bloques de alcance

El alcance (scope) determina la región del programa donde un nombre es válido. Existen dos tipos principales:

Ámbito estático o léxico: el alcance se define por la estructura del código fuente.

Ámbito dinámico: el alcance depende del orden en que se ejecutan los subprogramas.

Los bloques de alcance permiten organizar el código y evitar conflictos entre nombres.

### 4. Administración de memoria

La memoria en los programas puede clasificarse en:

Memoria estática: reservada en tiempo de compilación (ej. variables globales).

Memoria automática: asignada y liberada automáticamente en la pila (ej. variables locales).

Memoria dinámica: gestionada por el programador o por un recolector de basura (heap).

La administración de memoria es crucial para optimizar recursos y evitar errores como fugas de memoria o accesos inválidos.

## 5. Expresiones

Una expresión es una combinación de operandos (valores, variables) y operadores (aritméticos, lógicos, relacionales, etc.) que produce un resultado. Ejemplo:  $x + y * 2$ . Las expresiones son fundamentales porque representan cálculos y condiciones dentro del programa.

## 6. Comandos

Los comandos o sentencias son instrucciones que ejecuta el programa, como:

Asignación: `x = 5;`

Entrada/Salida: `print(x)`

Declaración: `int edad;`

Permiten controlar el comportamiento del software paso a paso.

## 7. Control de secuencia

Los lenguajes de programación incluyen mecanismos para alterar el flujo normal de ejecución:

Selección (decisión): permite elegir entre alternativas. Ejemplo: `if, else, switch`.

Iteración (repetición): permite ejecutar un bloque de código varias veces. Ejemplo: `for, while, do-while`.

Recursión: un subprograma se invoca a sí mismo directa o indirectamente. Es útil para problemas que se definen en términos más pequeños del mismo problema (como factorial, Fibonacci, recorridos de árboles).

## 8. Subprogramas

Un subprograma es un bloque de código que realiza una tarea específica y puede ser invocado desde otras partes del programa. Tipos:

Funciones: retornan un valor.

Procedimientos: no retornan valor directamente.

Facilitan la reutilización de código, la modularidad y la abstracción.

## 9. Tipos de datos

Los tipos de datos especifican qué clase de valores pueden almacenarse en una variable y qué operaciones se pueden realizar sobre ellos.

Primitivos: enteros, reales, caracteres, booleanos.

Compuestos: arreglos, registros, listas, pilas, colas.

Abstractos: definidos por el programador (clases, estructuras).

Son la base para representar información en un programa.