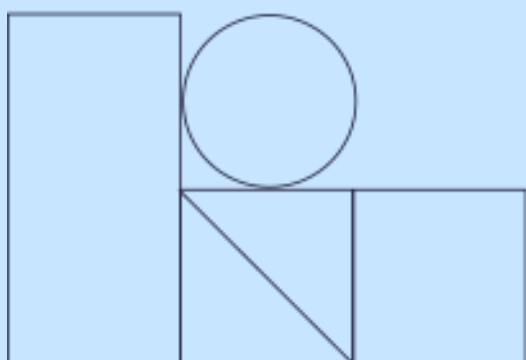


# Conceptos básicos de programación

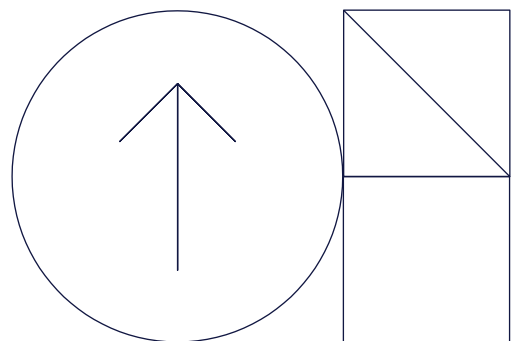
Qué es un lenguaje de  
programación



---

## Índice

Introducción	3
¿Qué es un lenguaje de programación?	4
¿Qué tipos de lenguaje de programación existen?	4
Lenguaje de programación de bajo nivel	5
Lenguajes de medio nivel	5
Lenguaje de programación de alto nivel	6
Generaciones	6
Traductores	7
Intérpretes	7
Compiladores	7
Intermedios	7
¿Para qué sirven los lenguajes de programación?	8

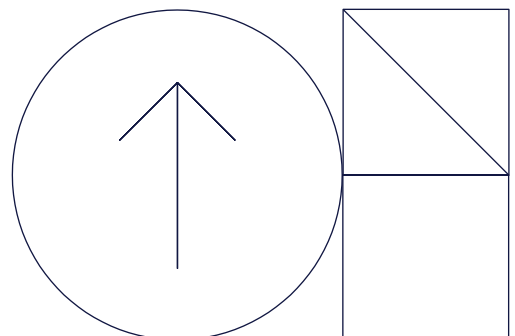


# Introducción

En los últimos años los lenguajes de programación han ido evolucionado en el desarrollo de sistemas o software, con el objetivo principal de facilitar al usuario las actividades que realiza día con día; por tal motivo, como programador, es importante conocer los conceptos básicos de programación, los tipos de lenguajes que se utilizan para el desarrollo y su funcionamiento para la interpretación de algoritmos, así como para dar solución a los problemas que pudieran presentarse.

Tener nociones de la programación de computadoras es indispensable en cualquier sector de la vida hoy en día, ya que, existen distintos problemas que como tales no es posible solucionarlos de la manera tradicional en la cual un ser humano los realiza. Es ahí donde entra el rol de la computadora, es decir nos ayuda a resolver estos problemas por así decirlo sin solución o con solución aparentemente muy complicada para un humano de manera rápida o con un margen de tolerancia suficientemente bueno.

Es por ello que se menciona en muchas ocasiones el concepto de programación en general.



## ¿Qué es un lenguaje de programación?

Un lenguaje de programación es un sistema con una estructura de comunicación similar al humano, este permite que los aparatos se entiendan entre sí y que a su vez interprete y ejecute las instrucciones de los usuarios.

La programación, es un conjunto de instrucciones ordenadas y sucesivas que tienen por objetivo ejecutar una tarea específica. A esas instrucciones se les denomina “código fuente”, que son únicos para cada lenguaje y están diseñados para cumplir una función específica.

Al igual que ocurre con los idiomas, los lenguajes de programación cuentan con un vocabulario que se rige por un conjunto de reglas gramaticales usando una sintaxis que, en este caso, nos permite dar instrucciones concretas a una máquina y que esta las interprete.

En términos generales, un lenguaje de programación es una herramienta que permite desarrollar software o programas para computadora. Los lenguajes de programación son empleados para diseñar e implementar programas encargados de definir y administrar el comportamiento de los dispositivos físicos y lógicos de una computadora. Lo anterior se logra mediante la creación e implementación de algoritmos de precisión que se utilizan como una forma de comunicación humana con la computadora.

A grandes rasgos, un lenguaje de programación se conforma de una serie de símbolos, reglas de sintaxis y semántica que definen la estructura principal del lenguaje y le dan un significado a sus elementos y expresiones.

Programación es el proceso de análisis, diseño, implementación, prueba y depuración de un algoritmo, a partir de un lenguaje que compila y genera un código fuente ejecutado en la computadora.

La función principal de los lenguajes de programación es escribir programas que permiten la comunicación usuario-máquina. Unos programas especiales (compiladores o intérpretes) convierten las instrucciones escritas en código fuente, en instrucciones escritas en lenguaje máquina (0 y 1).

Los intérpretes leen la instrucción línea por línea y obtienen el código máquina correspondiente.

En cuanto a los compiladores, traducen los símbolos de un lenguaje de programación a su equivalencia escrito en lenguaje máquina (proceso conocido como compilar). Por último, se obtiene un programa ejecutable.

## ¿Qué tipos de lenguaje de programación existen?

El lenguaje de programación es la base para construir todas las aplicaciones digitales que se utilizan en el día a día.

Hay que tener en cuenta que, en la práctica, la mayoría de lenguajes no pueden ser puramente clasificados en una categoría, pues surgen incorporando ideas de otros lenguajes y de otras filosofías de programación, pero no importa al establecer las clasificaciones, pues el auténtico objetivo de las mismas es mostrar los rangos, las posibilidades y tipos de lenguajes que hay.

Actualmente, existen más de 100 lenguajes de programación y se clasifican en dos tipos principales según el nivel de abstracción, es decir, según el grado de cercanía a la máquina:

- Lenguajes de programación de bajo nivel.
- Lenguajes de programación de medio nivel.
- Lenguajes de programación de alto nivel.

## Lenguaje de programación de bajo nivel

Este tipo de lenguaje de programación es utilizado para controlar directamente el hardware del dispositivo y dependen únicamente de la máquina, es decir, no pueden ser usados en aparatos diferentes de los que fueron creados originalmente.

Al estar prácticamente diseñados a medida del hardware, aprovechan al máximo las características del mismo.

Solo pueden ordenar las operaciones primarias para el funcionamiento del dispositivo, tienden a ser complejos por lo que lo suelen utilizar los fabricantes de hardware.

Con este lenguaje de programación de bajo nivel se puede reproducir audios y videos, mostrar imágenes, realizar operaciones matemáticas, seguir el movimiento del puntero, etc. Dentro de este grupo se encuentran:

**El lenguaje máquina:** este lenguaje ordena a la máquina las operaciones fundamentales para su funcionamiento. Consiste en la combinación de 0's y 1's para formar las órdenes entendibles por el hardware de la máquina. Este lenguaje es mucho más rápido que los lenguajes de alto nivel.

**El lenguaje ensamblador:** es un derivado del lenguaje máquina y está formado por abreviaturas de letras y números llamadas. Con la aparición de este lenguaje se crearon los programas traductores para poder pasar los programas escritos en lenguaje ensamblador a lenguaje máquina. Como ventaja con respecto al código máquina es que los códigos fuentes eran más cortos y los programas creados ocupaban menos memoria.

## Lenguajes de medio nivel

En la actualidad se suele omitir este grupo, quedando la clasificación dividida en lenguajes de bajo y alto nivel. Pero, dado que estos lenguajes de medio nivel tienen características diferenciadoras respecto a los de bajo nivel, merece la pena hacerles una mención.

Lenguaje de medio nivel es un lenguaje de programación como el lenguaje C, que se encuentran entre los lenguajes de alto nivel y los lenguajes de bajo nivel.

Suelen ser **clasificados muchas veces de bajo nivel**, pero permiten ciertos **manejos de alto nivel**. Son precisos para ciertas aplicaciones como la creación de sistemas operativos, ya que permiten un manejo abstracto (independiente de la máquina, a diferencia del ensamblador), pero sin perder mucho del poder y eficiencia que tienen los lenguajes de bajo nivel.

Una característica distintiva, por ejemplo, que convierte a C en un lenguaje de medio nivel y al Pascal en un lenguaje de alto nivel es que en el primero es posible manejar las letras como si fueran números (en Pascal no), y por el contrario en Pascal es posible concatenar las cadenas de caracteres con el operador suma y copiarlas con la asignación (en C es el usuario el responsable de llamar a las funciones correspondientes).

Otra de las características más peculiares del lenguaje de programación C; es el uso de “apuntadores”, los cuales son muy útiles en la implementación de algoritmos como **Listas ligadas, Tablas Hash y algoritmos de búsqueda y ordenamiento** que, para otros lenguajes de programación (como Java por ejemplo) suele ser un poco más complicado implementar.

Dos de los lenguajes de medio nivel más usados son: **C, C++**.

## Lenguaje de programación de alto nivel

Tienen como objetivo facilitar el trabajo del programador, ya que utilizan unas instrucciones más fáciles de entender.

Además, el lenguaje de alto nivel permite escribir códigos mediante idiomas que conocemos (español, inglés, etc.) y luego, para ser ejecutados, se traduce al lenguaje de máquina mediante traductores o compiladores.

Están dirigidos a solucionar problemas mediante el uso de Estructuras Dinámicas de Datos.

Se tratan de lenguajes independientes de la arquitectura del ordenador. Por lo que, en principio, un programa escrito en un lenguaje de alto nivel, lo puedes migrar de una máquina a otra sin ningún tipo de problema.

Estos lenguajes permiten al programador olvidarse por completo del funcionamiento interno de la máquina para la que se está diseñando el programa.

Suelen usar tipos de datos para la programación y hay lenguajes de propósito general (cualquier tipo de aplicación) y de propósito específico (como FORTRAN para trabajos científicos).

Algunos de los lenguajes de alto nivel más usados son: **Java, C#, Python, JavaScript.**

## Generaciones

- **Primera generación:** lenguaje máquina.
- **Segunda generación:** se crearon los primeros lenguajes ensambladores.
- **Tercera generación:** se crean los primeros lenguajes de alto nivel. Ej. C#, Pascal, Cobol
- **Cuarta generación:** son los lenguajes capaces de generar código por sí solos, son los llamados RAD. Son herramientas que permiten construir aplicaciones sencillas combinando piezas prefabricadas (librerías). Algunos proponen reservar el nombre de cuarta generación para la programación orientada a objetos, por sus siglas, POO.
- **Quinta generación:** aquí se encuentran los lenguajes orientados a la inteligencia artificial.

# Traductores

El traductor es una herramienta esencial en la programación o desarrollo, encargándose de convertir código fuente de un determinado lenguaje de programación a código máquina que puede «entender» directamente el ordenador.

De acuerdo al modo en que llevan a cabo el proceso de conversión, los traductores se dividen en dos conjuntos: intérpretes y compiladores.

## Intérpretes

Un intérprete es un traductor que ejecuta las líneas de código que conforman un programa una a una y directamente. Es un programa que va leyendo el código fuente de otro programa y lo va ejecutando según lo lee.

El código máquina generado en el momento de la traducción no es almacenado para su posterior ejecución cuando se solicite. Por tanto, el programa realizado con un lenguaje de programación traducido por un intérprete siempre necesitará a éste para poder ejecutarse. No existe independencia entre la fase de traducción y ejecución.

Cada vez que el programa necesite ejecutarse, el traductor tendrá que convertir el código fuente a código máquina (interpretar). Esta característica hace que, por regla general, los lenguajes que usan este tipo de traductor ofrezcan un menor rendimiento.

En contrapartida, ofrecen un mejor entorno de programación, depuración y mantenimiento que el otro conjunto de traductores: los compiladores.

## Compiladores

Los traductores de tipo compilador convierten el código fuente de un programa en código máquina, generando en el camino un recurso adicional (un ejecutable) que será el que utilizaremos para ejecutar el programa.

En realidad, el programa fuente (programa con el código fuente, valga la redundancia) será convertido, sentencia a sentencia, a código máquina, creando un programa objeto o código objeto.

Sin embargo, para crear el programa final, autoejecutable, será necesario un proceso adicional: el enlazado o montaje (realizado por el programa montador, enlazador o linker).

El resultado final, será un programa autoejecutable que no necesitará ninguna herramienta adicional para hacerlo. Simplemente le indicaremos que se ejecute cuando lo deseemos y así lo hará.

## Intermedios

Existen otros traductores que producen un código intermedio entre el código fuente y el código máquina. Son, por tanto, una mezcla de interpretados y compilados.

En el lenguaje Java por ejemplo, se genera un recurso **bytecode**: código precompilado que necesita interpretarse por la JVM o máquina virtual de Java para ejecutarse.

# ¿Para qué sirven los lenguajes de programación?

En general un lenguaje de programación sirve para programar. Sin embargo, cada lenguaje tiene un alcance y forma de comunicación diferente.

En resumidas cuentas, el lenguaje de bajo nivel permite la comunicación interna de la máquina, cada instrucción tiene su código único de operación.

Y el lenguaje de alto nivel facilita la captación de instrucciones que el programador le da a la máquina, mientras que este introduce datos en el idioma conocido la máquina lo va absorbiendo en lenguaje de máquinas mediante traductores o compiladores, permitiendo así:

- reducir el tiempo de programación;

- entender más fácilmente la tarea a realizar;

- permitir al programador desvincularse del funcionamiento interno de la máquina, entre otros.

Como resumen, el lenguaje de bajo nivel es cercano a los idiomas de las máquinas mientras que el lenguaje de alto nivel está más cerca del entendimiento e idioma humano.