

TIPOS DE LENGUAJES DE PROGRAMACIÓN

Cápsula interactiva

Lenguajes de programación

Un programa se escribe con instrucciones utilizando un **lenguaje de programación**, el cual está definido por su **sintaxis**, es decir, el conjunto de reglas que establece e indica cómo debe escribirse cada instrucción para considerarse correcto en ese lenguaje.

Existen diversos lenguajes de programación, por ejemplo:

C	Java	Python	C++	C#	Visual Basic	JavaScript	PHP	Go
SQL	MATLAB	Perl	Assembly	Ruby	Groovy	Lisp	Lua	Bash

Puede revisarse el índice de TIOBE (<https://www.tiobe.com/tiobe-index/>), el cual evalúa la popularidad de diferentes lenguajes de programación.

Si bien, existen varias formas de clasificar los lenguajes de programación, por ejemplo, según su paradigma o según su propósito; en este curso se abordarán según su evolución y según su implementación o ejecución. Así, se pueden clasificar los lenguajes de programación como:

Según su evolución	Según su implementación o ejecución
1. Lenguajes máquina	1. Lenguajes compilados
2. Lenguajes ensambladores	2. Lenguajes interpretados
3. Lenguajes de alto nivel	

El lenguaje máquina es entendido directamente por una computadora, es decir, es su “lenguaje natural” y está definido por el diseño del hardware de dicha computadora. Por lo general, los lenguajes máquina consisten en cadenas de números (que finalmente se reducen al lenguaje binario de ceros y unos) que instruyen a la computadora para realizar sus operaciones. Los lenguajes máquina son dependientes de la máquina, es decir, un lenguaje máquina en particular puede utilizarse solamente en un tipo de computadora. Los lenguajes máquina son difíciles de comprender para los humanos, por ejemplo, un código máquina en la memoria de una computadora podría verse de la siguiente forma:

0x100000f20 0x0afc45c7 0xc7000000 0x00234f845 0x458b0000

Dado que la programación en lenguaje máquina resultaba demasiado lenta y tediosa, los programadores comenzaron a utilizar abreviaturas del inglés para representar las operaciones básicas. De este modo, nacieron los **lenguajes ensambladores**. Por ejemplo, un programa en un lenguaje ensamblador que suma dos números podría escribirse como:

- LOAD NUMERO1
- LOAD NUMERO2
- ADD NUMERO1, NUMERO 2
- STORE RESULTADO

Aunque dicho código es más claro para los humanos, será incomprensible para la computadora, hasta que sea traducido al lenguaje máquina. El programa encargado de esa traducción se denomina **ensamblador**.

El uso de los lenguajes ensambladores, incrementó rápidamente el uso de las computadoras. Sin embargo, éstas aún requerían de muchas instrucciones para llevar a cabo tareas sencillas. Para acelerar el proceso de programación, se desarrollaron los **lenguajes de alto nivel**, que permiten a los programadores escribir instrucciones que se parecen mucho al inglés común y con una notación matemática más natural. Por ejemplo, un programa para sumar dos números, escrito en un lenguaje de alto nivel, podría ser como:

resultado = numero1 + numero2

El programa encargado de traducir el código escrito en un lenguaje de alto nivel a lenguaje máquina se le llama **compilador**.

El proceso de compilación de un programa escrito en lenguaje de alto nivel a un lenguaje máquina puede tardar un tiempo considerable dependiendo de su complejidad y extensión. Es por ello que, para facilitar la ejecución y corrección de códigos, se crearon los programas **intérpretes** que permiten ejecutar un programa de alto nivel sin necesidad de compilarlo completamente al lenguaje máquina, sino que va traduciendo cada instrucción, una a una, durante el proceso de ejecución.

Aunque la ejecución de los programas compilados es más rápida que los programas interpretados, los lenguajes interpretados son populares en ambientes de desarrollo de programas, en los cuales los programas se recompilan de forma frecuente conforme se agregan nuevas características y se corrigen errores. Una vez que se desarrolla un programa, una versión compilada puede ejecutarse de manera más eficiente.

En resumen, un **lenguaje compilado** requiere ser traducido completamente al lenguaje máquina mediante un compilador antes de su ejecución. Al realizar el proceso de compilación, generalmente, se genera un archivo ejecutable. Por ejemplo, C y C++ son lenguajes compilados.

Por otro lado, un **lenguaje interpretado** no requiere compilación, puesto que durante su ejecución cada instrucción se traduce, una a una, al lenguaje máquina a partir del código fuente. Por ejemplo, Python y JavaScript son lenguajes interpretados.



REFERENCIAS BIBLIOGRÁFICAS

- Deitel, H. y Deitel, P. (2004). Cómo programar en C/C++ y Java. Pearson Educación.
- TIOBE (enero de 2021). TIOBE Programming Community Index. Recuperado de: <https://www.tiobe.com/tiobe-index/>