

Lenguajes de programación
Arquitectura y Sistemas Operativos
Tecnicatura Superior en Programación.
UTN-FRA

Autores: *Prof. Martín Isusi Seff*

Revisores: *Prof. Marcos Pablo Russo*

Versión : 1



Esta obra está bajo una [Licencia Creative Commons Atribución-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-sa/4.0/).

Lenguajes de programación.

Para poder comenzar a hablar de los lenguajes de programación, sus características, los distintos tipos, etc., es necesario responder una pregunta fundamental: ¿Qué es un lenguaje de programación?

Un lenguaje de programación es un lenguaje especial, que especifica su propia sintaxis, reglas y un conjunto de palabras clave, utilizado para poder comunicar instrucciones a una máquina, más específicamente, a una computadora. Los lenguajes de programación, son los que nos permiten el desarrollo del software.

Durante la primera generación de computadoras (1938-1952) y hasta la creación de las primeras tarjetas perforadas, la lógica que ejecutaba cualquier sistema de computación se almacenaba de manera cableada en el equipo. La implementación de las tarjetas perforadas como medio de almacenamiento a principio de los años '50 (en el final de la 1ra generación de computadoras), da lugar a lo que se conoce como la primera generación de lenguajes de programación. En esta primera generación, los programas eran escritos directamente en lenguaje de máquina. No existía abstracción alguna, por lo que las instrucciones que escribía el programador se ejecutaban directamente sobre el hardware del equipo.

Coincidiendo con el inicio de la segunda generación de computadoras (1953-1962), surgen los primeros lenguajes de ensamblado. Esto da lugar a una segunda generación de lenguajes de programación.

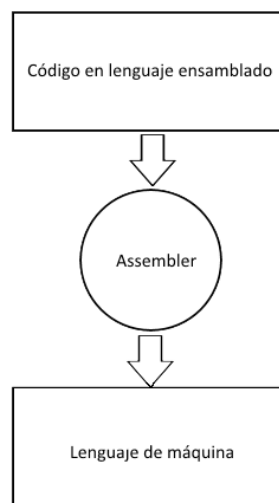
Por último, a finales de los años '50, el desarrollo de nuevos lenguajes de programación como C, COBOL, BASIC, y muchos de los lenguajes de programación actuales, corresponden a la tercera generación de lenguajes de programación. Su meta, fue abstraer al programador del hardware del sistema.

Niveles de abstracción en los lenguajes de programación

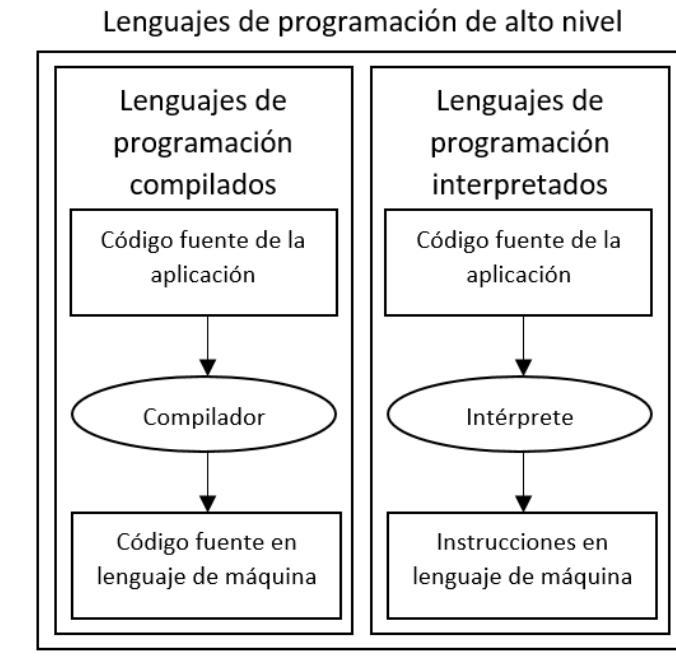
Los lenguajes de programación suelen clasificarse por su nivel de abstracción con respecto al hardware, por lo tanto, tendremos los lenguajes de programación de bajo nivel y los lenguajes de programación de alto nivel.

Dentro de los lenguajes de programación de bajo nivel tenemos al lenguaje de máquina y los lenguajes de ensamblado. El lenguaje de máquina es el lenguaje con el que trabaja el hardware y es, de hecho, el único con el que puede trabajar. Para el ser humano este lenguaje no es más que una secuencia de dígitos binarios, por lo que no es utilizado directamente por los programadores.

Siendo el lenguaje de máquina el código con el que trabaja el hardware directamente, podemos decir que no posee abstracción alguna, por lo tanto, el primer nivel de abstracción corresponde a los lenguajes de ensamblado. Estos lenguajes consisten de una serie de instrucciones que se ejecutan directamente sobre el microprocesador y sus componentes (registros, contadores, etc.). Generalmente, estas instrucciones se representan con una abreviación o palabra clave tales como MOV, JUMP, ADDL, siendo cada una, una instrucción distinta en el microprocesador. Dado que el hardware sólo puede ejecutar instrucciones almacenadas en lenguaje de máquina, debe existir un componente encargado de traducir las instrucciones en lenguaje de ensamblado a lenguaje de máquina. Este componente es el ensamblador o assembler



Por encima de los lenguajes de programación de bajo nivel, nos encontramos con los lenguajes de programación de alto nivel. Estos son lenguajes de programación que tienen como característica principal, la abstracción del hardware del equipo. Para poder escribir un programa en un lenguaje de programación de alto nivel, no es necesario conocer la arquitectura del sistema, algo que sí es necesario cuando trabajamos, por ejemplo, con un lenguaje de ensamblado. Dentro de los lenguajes de programación de alto nivel, también podemos identificar distintos niveles de abstracción. Esto quiere decir que algunos lenguajes de programación de alto nivel nos proveen herramientas para manejar aspectos del hardware directamente, como, por ejemplo, el manejo de punteros en C. Revisando el ejemplo anterior, podemos decir que C es un lenguaje de programación de alto nivel, pero dentro de este grupo se encuentra entre los que tienen el nivel de abstracción más bajo. Un ejemplo contrario podría Javascript, cuya sintaxis y funcionalidad se encuentra muy abstraída del hardware, por lo que no podemos utilizarlo para realizar desarrollos que controlen el hardware directamente.



Distintos tipos de lenguajes de programación: Interpretados y compilados

Debido a que el sistema sólo es capaz de ejecutar las instrucciones almacenadas en lenguajes de máquina, los lenguajes de programación de alto nivel necesitarán de un componente que “traduzca” las instrucciones escritas por el programador, al lenguaje de máquina. Este componente puede ser un compilador o un intérprete. Esto hace que dentro de los lenguajes de programación de alto nivel existan dos grandes tipos: Lenguajes de programación compilados y lenguajes de programación interpretados.

Los lenguajes de programación compilados son aquellos que se traducen enteramente al lenguaje de máquina para ser ejecutados directamente en el hardware, mientras que los lenguajes de programación interpretados son aquellos en los que necesitan de un software intermedio llamado intérprete que será el encargado de ejecutar las instrucciones. Generalmente, el intérprete sí está almacenado en lenguaje de máquina.

Lenguajes de programación compilados

Los lenguajes de programación compilados son aquellos que utilizan un compilador para transformar el código fuente¹ de la aplicación en código de máquina que pueda interpretar el sistema. El compilador tomará los archivos que contienen el código fuente y dará como

¹ Cuando hablamos de código fuente, nos referimos al código escrito por el programador, en un determinado lenguaje de programación y que se almacena en archivos cuya extensión dependerá del lenguaje de programación utilizado. Por ejemplo, el código fuente de una aplicación desarrollada en lenguaje C, corresponderá a los archivos con extensión .c que correspondan a la misma.

resultado, archivos con el código de máquina.

interpretar el sistema. El compilador tomará los archivos que contienen el código fuente y dará como resultado, archivos con el código de máquina.

La principal ventaja de los lenguajes compilados sobre los interpretados, es la velocidad de ejecución. Dado que el compilador traduce el código fuente y lo almacena en código de máquina, la ejecución de estos archivos resulta mucho más rápida y eficiente que la ejecución de instrucciones en un lenguaje interpretado.

Dentro de los lenguajes de programación de alto nivel, aquellos que tienen a tener un menor nivel de abstracción suelen ser compilados ya que, al almacenarlo finalmente en lenguaje de máquina, permiten ejecutar instrucciones directamente sobre el hardware. Algunos ejemplos de estos lenguajes son C, C++, etc.

Entre las desventajas que podemos mencionar de los lenguajes compilados, se encuentra la necesidad de compilar el código fuente antes de poder ejecutar la aplicación. Si bien una vez compilada, la ejecución es más rápida, el hecho de que haya que compilar siempre antes de ejecutar hace que los tiempos de desarrollo sean más lentos. Otra desventaja radica en el hecho de que, como cada arquitectura de hardware posee su propio lenguaje de máquina, las aplicaciones escritas en lenguajes compilados deben ser compiladas para cada arquitectura que se desee, por lo tanto, podemos decir que los lenguajes de programación compilados son dependientes de la plataforma en la que se ejecutan.

Algunos ejemplos de lenguajes de programación compilados son C, C++, C#, Java², Basic, etc.

Lenguajes de programación interpretados

En contraste con los lenguajes de programación compilados, los lenguajes de programación interpretados no necesitan ser traducidos a código de máquina para ejecutarse. O al menos, no directamente. En el caso anterior, el compilador traducía el código fuente completo y lo almacenaba en lenguaje de máquina. Aquí, tendremos un intérprete, que será el encargado de traducir el código fuente a código de máquina, pero en lugar de almacenarlo en un archivo, ejecutará las instrucciones directamente. Para poder hacer esto, el intérprete lee línea por línea cada instrucción del código fuente, y las ejecuta independientemente de manera

² A pesar de que Java es un lenguaje de programación compilado, una vez realizada la compilación del código fuente, los archivos resultantes no contienen código de máquina. El código almacenado en los archivos resultantes de la compilación, contienen un código de bytes o *bytecode* que será ejecutado por una "máquina virtual" Java. Será luego la máquina virtual Java la encargada de traducir ese *bytecode* a código de máquina nativo.

ordenada.

La principal desventaja de este tipo de lenguajes es la velocidad de ejecución. Dado que el código nunca se encuentra almacenado en lenguaje de máquina, cada ejecución requiere de la “traducción” de las instrucciones. De todas maneras, al no requerir una compilación cada vez que se ejecutan las aplicaciones, el desarrollo de las mismas se realiza de una manera más ágil.

Algunos ejemplos de lenguajes de programación interpretados son PHP, Perl, Ruby, JavaScript, Python, etc.

Clasificación de los lenguajes de programación dependiendo del paradigma

Dentro del ámbito del desarrollo de software, podemos encontrar distintos paradigmas o “formas” de programación. Existen paradigmas orientados a la metodología de trabajo, que imponen reglas o formas de trabajar desde lo práctico. Por otro lado, podemos identificar paradigmas de programación que dependen del lenguaje. Algunos ejemplos de paradigmas son la programación orientada a objetos, programación procedural, programación funcional, etc. Las características de cada lenguaje de programación serán las que ubiquen al mismo dentro de alguno de los paradigmas.

Resumen

Los lenguajes de programación son la forma en la que el programador puede dar las instrucciones a la computadora para que la misma realice distintas operaciones. Los lenguajes de programación pueden clasificarse por su nivel de abstracción, encontrando en el nivel más bajo el lenguaje de máquina y los lenguajes de ensamblado. Sobre los lenguajes de programación de bajo nivel, en un nivel superior, encontramos los lenguajes de programación de alto nivel. Estos lenguajes son aquellos que buscan abstraer al programador del hardware, es decir que la persona no necesite desarrollar para distintos dispositivos la misma aplicación. Dentro de los lenguajes de programación de alto nivel podemos identificar dos grandes grupos: Lenguajes de programación compilados y lenguajes de programación interpretados. Por último, dentro de la programación existen distintos paradigmas, que corresponden a distintas “formas” de programar. Las características de cada lenguaje de programación son las que ubican a cada uno en alguno de los paradigmas.