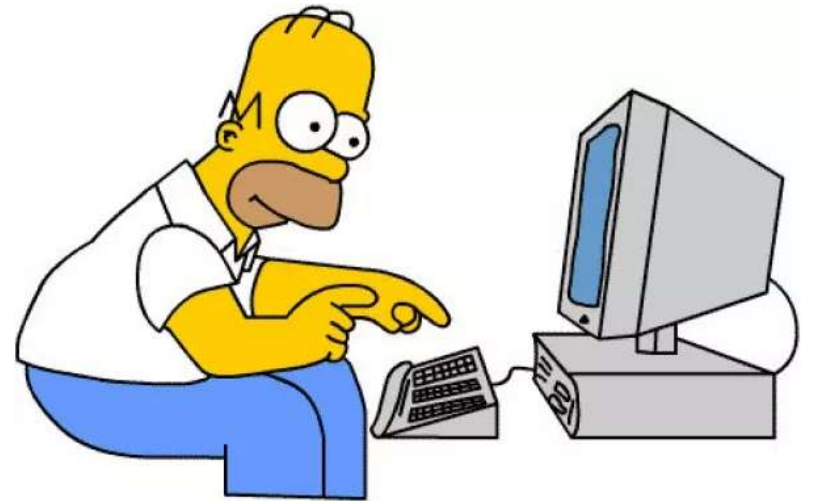


INTRODUCCIÓN A LA PROGRAMACIÓN

Lenguajes de programación.



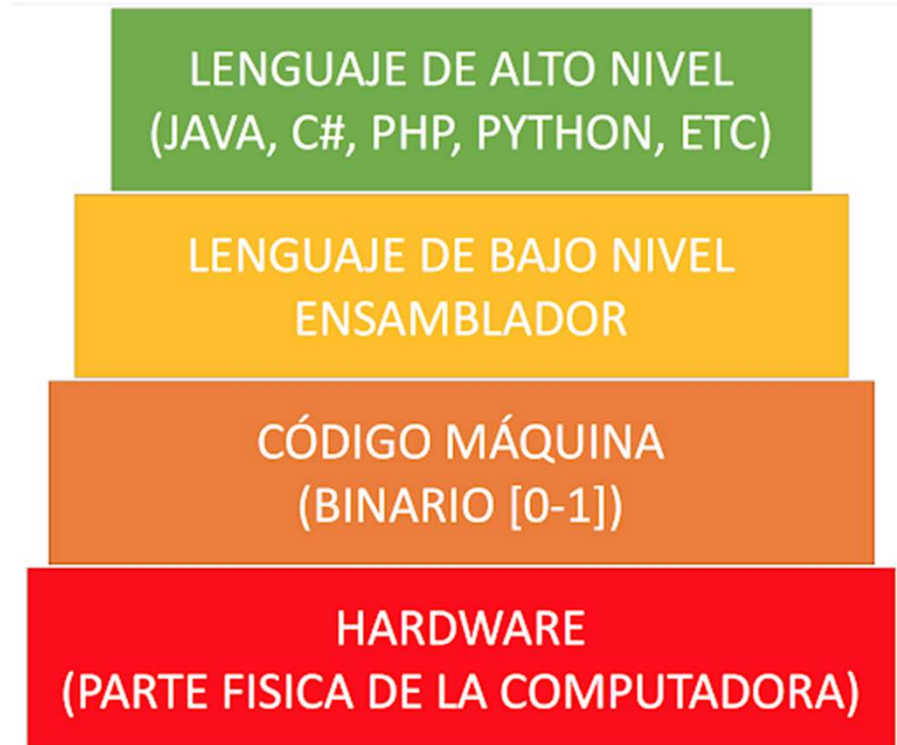
INTRODUCCIÓN A LA PROGRAMACIÓN

Lenguajes de programación.

Un **lenguaje de programación** es un lenguaje formal, es decir, un lenguaje con reglas gramaticales bien definidas, donde se especifican una serie de instrucciones entendibles por un ordenador.

Debemos diferenciar entre:

- ❖ Lenguaje **máquina**.
- ❖ Lenguaje de **bajo nivel**.
- ❖ Lenguaje de **alto nivel**.



INTRODUCCIÓN A LA PROGRAMACIÓN

Lenguajes de programación.

Lenguaje de máquina

- Secuencias de números (0's y 1's)
- Lenguaje propio de cada computadora

Lenguaje ensamblador

- Utiliza códigos parecidos al inglés
- Varía de acuerdo al tipo de procesador
- Ejemplo: `LOAD X ADD Y STORE Z`

Lenguajes de alto nivel

- Instrucciones en un lenguaje familiar
- Usa notaciones matemáticas conocidas
Ejemplo: $Z = X + Y$
- Independiente de la máquina

INTRODUCCIÓN A LA PROGRAMACIÓN

Lenguajes de programación.

Para que un ordenador pueda comprender las instrucciones que contiene un **programa de alto nivel**, el código fuente debe convertirse a **lenguaje máquina**. De este procedimiento, dependiendo del lenguaje de programación, se encarga un **compilador** o un **intérprete**.

Un **intérprete** es un programa que ejecuta directamente las instrucciones escritas en un lenguaje de programación dado.

Un **compilador** es un programa que transforma el código fuente de un programa a su equivalente en otro lenguaje de programación de más bajo nivel.

INTRODUCCIÓN A LA PROGRAMACIÓN

Lenguajes de programación.

Un **compilador** es un programa que lee un programa escrito en un lenguaje de alto nivel (el **lenguaje fuente**) y lo traduce antes de ejecutarlo a un programa equivalente en **código máquina** y listo por tanto para ejecutarse en la computadora.

El compilador informa a su usuario de la presencia de errores en el programa fuente.

Entre los lenguajes compilados puros se incluyen, entre otros, los veteranos C, C++ y Pascal.

INTRODUCCIÓN A LA PROGRAMACIÓN

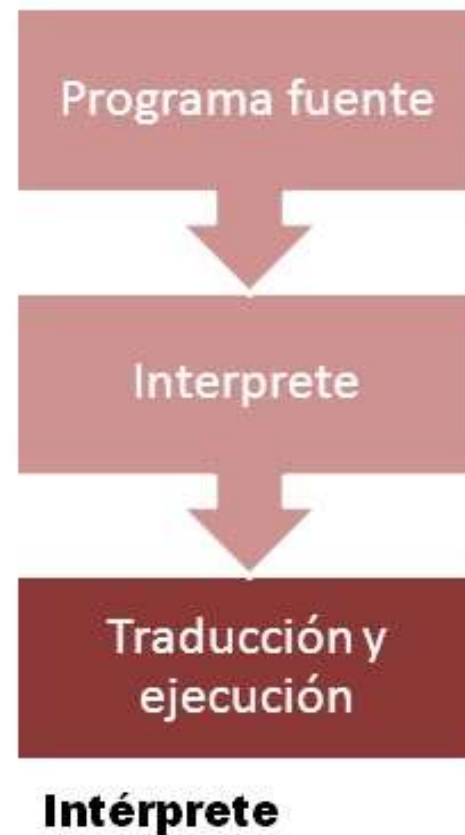
Lenguajes de programación.

Un **intérprete** analiza el **programa fuente** y lo **ejecuta directamente**, sin generar ningún código equivalente. Es un traductor que realiza la operación de compilación paso a paso. Para cada sentencia que compone el texto de entrada, se realiza una traducción, ejecuta dicha sentencia y vuelve a iniciar el proceso con la sentencia siguiente.

BASIC, Perl, Python, Ruby y PHP son algunos de los lenguajes de programación más famosos que dependen de un intérprete para ser traducidos de código fuente a código máquina. Por ello, también suelen llamarse **lenguajes interpretados**.

INTRODUCCIÓN A LA PROGRAMACIÓN

Lenguajes de programación.



INTRODUCCIÓN A LA PROGRAMACIÓN

Lenguajes de programación.

Tanto los **compiladores** como los **intérpretes** cumplen la función de convertir el código de software que se ha escrito a un formato ejecutable y legible por la computadora.

Un intérprete ejecuta el código **AHORA**. Un compilador lo prepara para ejecutarlo **LUEGO**

Un compilador genera un programa “stand-alone” es decir que tiene sentido y se puede ejecutar por sí solo, mientras que un programa interpretado siempre necesita su intérprete correspondiente para poder ejecutarse.

INTRODUCCIÓN A LA PROGRAMACIÓN

Lenguajes de programación.

Por lo tanto, es más fácil distribuir un programa compilado. El problema es que sólo podrás ejecutarlo en la plataforma para la que se ha compilado. Cada sistema operativo necesita **una versión compilada diferente**.

Y al revés, puedes distribuir la misma copia de un programa interpretado a usuarios en plataformas muy diferentes. Pero cada uno de ellos tendrá que disponer de un intérprete para su plataforma.

INTRODUCCIÓN A LA PROGRAMACIÓN

Lenguajes de programación.

Es más fácil escribir un programa que sea **multiplataforma** con lenguajes interpretados. De hecho lo que estarás haciendo es escribir el programa para una plataforma específica: el intérprete y dejarás que sean las diferentes versiones del intérprete quienes te den la capacidad de ser multiplataforma. En cambio, para programas compilados, el programador es responsable de gestionar las diferencias entre plataformas.

INTRODUCCIÓN A LA PROGRAMACIÓN

Lenguajes de programación.

También existe el llamado modelo de compilación en tiempo de ejecución (en inglés, *just-in-time-compiler*, o “*compilador justo a tiempo*”).

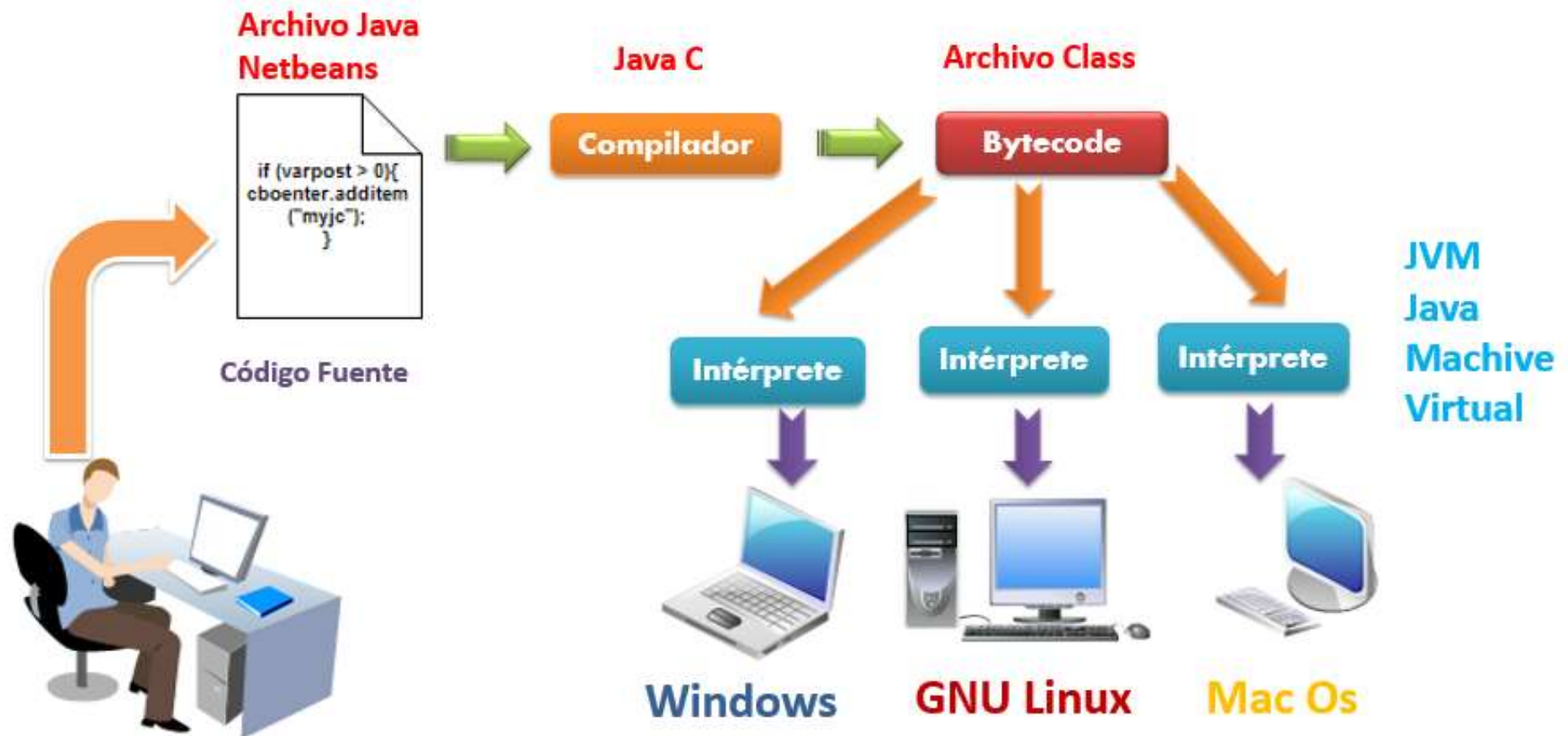
El lenguaje primero se compila a su representación intermedia (código de bytes) que luego se interpreta en tiempo de ejecución.

Ésta es la tendencia actual.

Java es uno de los ejemplos más conocidos de lenguaje basado en compilación en tiempo de ejecución.

INTRODUCCIÓN A LA PROGRAMACIÓN

Lenguajes de programación.



INTRODUCCIÓN A LA PROGRAMACIÓN

Lenguajes de programación.

“Fin del tema”

