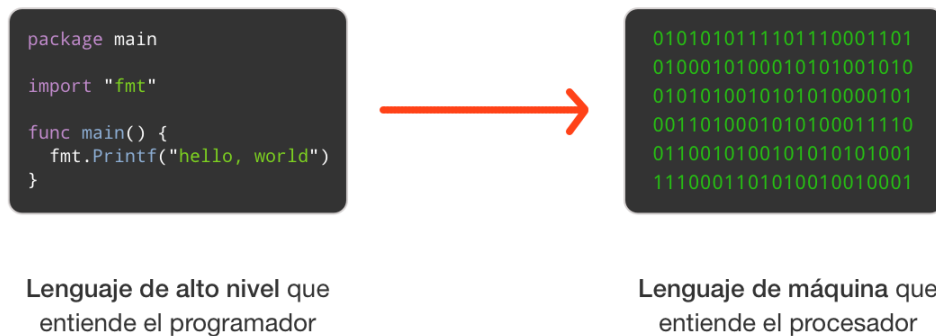


Lenguaje compilado vs Lenguaje interpretado

¿Cuál es la diferencia y las ventajas/desventajas de un lenguaje compilado versus uno interpretado?

Tanto **compiladores** como **interpretadores** son programas que convierten el código que escribimos a **lenguaje de máquina**.

Lenguaje de máquina son las **instrucciones que entiende la computadora** (el **procesador** para ser más exactos) **en código binario** (unos y ceros).



La principal diferencia entre un lenguaje compilado y uno interpretado es que el **lenguaje compilado** requiere un paso adicional antes de ser ejecutado, **la compilación**, que **convierte el código a lenguaje de máquina**. Un **lenguaje interpretado**, por otro lado, **es convertido a lenguaje de máquina a medida que es ejecutado**.

Ejemplos de **lenguajes compilados** incluyen **C, C++, Java, Go y Rust**, entre muchos otros. Ejemplos de **lenguajes interpretados** incluyen **Ruby, Python y JavaScript**, entre muchos otros. A todos estos lenguajes se les conoce como **lenguajes de alto nivel**.

Ventajas y Desventajas

En general, **el ciclo de desarrollo** (el tiempo entre el momento en que se escribe el código y el momento en que se realizan las pruebas) **es más rápido en un lenguaje interpretado**. Eso se debe a que para los lenguajes compilados es necesario realizar el proceso de compilación cada vez que se modifica el código fuente, aunque con herramientas adicionales se puede automatizar.

Otra **desventaja** de un **lenguaje compilado** es que **al compilar un programa se debe crear un ejecutable para cada uno de los sistemas operativos en donde se utilizará**, ya que por ejemplo un ejecutable creado para Linux no va a servir en Windows por ejemplo.

Sin embargo, **un lenguaje compilado es mucho más rápido que uno interpretado.** Esto se debe a que **cuando es ejecutado ya se encuentra en código de máquina y eso también le permite hacer algunas optimizaciones que no son posibles con un lenguaje interpretado.**

Además de la velocidad, otra desventaja de un lenguaje interpretado es que, **para ser ejecutado, debe estar instalado el interpretador.** Esto no es necesario en un lenguaje compilado que es convertido a lenguaje de máquina.

En general, **un lenguaje compilado está optimizado para el momento de la ejecución,** aunque esto signifique una carga adicional para el programador. Por otro lado, **un lenguaje interpretado está optimizado para hacerle la vida más fácil al programador,** aunque eso signifique una carga adicional para la máquina.

Conclusión

Se puede decir que los lenguajes de alto nivel se pueden dividir en tres ramas: **puramente compilados** (C++, Go, etc.), **parcialmente compilados** (Java, Elixir, etc.) e **interpretados** (Ruby, JavaScript, etc.).

Sin embargo, la brecha entre estas ramas cada vez es menor. Hoy existen lenguajes compilados a lenguaje de máquina como Go y Rust que se inclinan cada vez más en la productividad y facilidad para el programador mientras que los lenguajes interpretados son cada vez más rápidos en ejecución.