

Intérprete

El programa es traducido cada vez que se vaya a ejecutar. Los intérpretes no producen un lenguaje objetivo como en los compiladores. Un intérprete lee el código como está escrito e inmediatamente lo convierte en acciones; es decir, lo ejecuta en ese instante.

Existen lenguajes que utilizan un intérprete (como por ejemplo **Java**) que traduce en el instante mismo de lectura el código en lenguaje máquina para que pueda ser ejecutado.

Estructura de un intérprete

Un intérprete es un programa que se encarga de leer un lenguaje de programación y traducirlo a un lenguaje de máquina que el computador pueda entender. Esta tarea es llevada a cabo por varias partes diferentes, cada una con su propia función.

Un intérprete generalmente está compuesto por un **traductor a representación interna**. Esta parte del intérprete es responsable de analizar el lenguaje de programación y traducirlo a una representación interna que sea entendible para el intérprete. Esta representación interna generalmente es un árbol de sintaxis abstracta o AST.

Otra parte del intérprete es una **tabla de símbolos**. Esta parte es responsable de almacenar todos los nombres de símbolos y variables del programa para su posterior uso. Esta tabla de símbolos también puede almacenar información sobre el tipo de datos, el ámbito y otra información relevante.

También hay una parte llamada **evaluador de representación interna**. Esta parte del intérprete es responsable de recorrer el AST y ejecutar los comandos correctos. El evaluador también debe controlar los flujos de control, como el bucle while o el ciclo for.

La última parte de un intérprete es el **tratamiento de errores**. Esta parte del intérprete se encarga de detectar errores y mostrar mensajes de error con información relevante para el usuario. Esto ayuda al usuario a identificar y corregir errores de programación.

Tipos de intérpretes

Existen dos tipos principales de intérpretes: los **iterativos** y los **recursivos**. Los intérpretes iterativos se encargan de analizar el código línea por línea, mientras que los intérpretes recursivos se encargan de analizar el código línea por línea y luego volver a la línea anterior para realizar cálculos.

Además de estos dos tipos principales, también hay varios tipos de intérpretes especializados. Estos incluyen intérpretes **puros**, que ejecutan el código línea por línea sin almacenar datos intermedios, intérpretes **avanzados**, que almacenan datos intermedios para optimizar la ejecución del programa, y los intérpretes **incrementales**, que se ejecutan en varias etapas para optimizar el tiempo de ejecución.