

1. Conceptos básicos de programación

Índice

[1. Lenguajes de programación](#)

[2. Herramientas de desarrollo](#)

1. Lenguajes de programación

Definición

Un lenguaje de programación es un **lenguaje formal** (o artificial, es decir, un lenguaje con reglas gramaticales bien definidas) que proporciona a una **persona**, en este caso el programador, la capacidad y habilidad de escribir (o programar) una serie de **instrucciones** o secuencias de órdenes en forma de **algoritmos** con el fin de controlar el **comportamiento** físico o lógico de un sistema informático, para que de esa manera se puedan obtener diversas clases de **datos** o ejecutar determinadas **tareas**.

(Wikipedia, consultado en 25 de septiembre de 2023)

Tipos de lenguajes

Alto y bajo nivel

Un lenguaje de programación se puede clasificar en función de su "nivel de abstracción", que es cuán cerca o lejos está del ser humano o de la máquina en cuanto a su sintaxis y los elementos que lo componen.

Por ejemplo, el lenguaje de más bajo nivel que existe es el llamado "código máquina", compuesto de meros ceros y unos, exactamente lo que la máquina entiende y procesa (diferencias de potencial en transistores eléctricos).

En cambio, un lenguaje de muy alto nivel sería aquel que tiene un enfoque declarativo, esto es, que al escribir o leer el código en ese lenguaje se parece al lenguaje natural que hablan las personas en un idioma humano concreto.

Compilados e interpretados

Se llama código fuente (o para abreviar código) al resultado de escribir un programa informático en un lenguaje determinado.

Los lenguajes compilados son aquellos que requieren de un proceso de compilación previo a la ejecución. En dicho proceso se verifica que todo el código escrito cumple unas reglas sintáticas, semánticas, léxicas y morfológicas (como cualquier otro lenguaje formal).

En caso de ser así se genera un fichero ejecutable (código máquina dependiente de la plataforma, esto es, la combinación de hardware y software concreta) o un código objeto (un paso intermedio entre el fuente y el ejecutable), dependiendo del lenguaje.

Si no se cumple las reglas, se generarán los mensajes de error correspondientes y estos deberán ser corregidos para lograr el caso satisfactorio anterior. Cuando se da este caso, ninguna de las operaciones del

programa es ejecutada.

Por otro lado, los lenguajes interpretados son aquellos que no tienen este proceso previo de revisión de todo el código del programa, sino que son ejecutados directamente línea a línea, de manera que si una de ellas da un error (independientemente del tipo que sea) se detiene la ejecución.

Sin embargo, las líneas que se hayan intepretado hasta ese momento sí habrán sido ejecutadas, lo que podría generar problemas a futuro por esta ejecución parcial del programa (por ejemplo, por usar recursos no liberados a posteriori, como un fichero o una base de datos).

2. Herramientas de desarrollo

La persona que codifica el programa requiere a su vez de aplicaciones que le permitan desarrollarlo.

Aunque cada vez existen más recursos en esta línea, se podría hablar de dos grandes categorías:

- Editores de código / IDEs: Aplicaciones de procesamiento de texto específicamente diseñadas para tratar con código fuente, con mayor o menor grado de funcionalidad y ayudas al programador según el caso.
- Compiladores / intérpretes: Herramientas que se encargan de realizar el proceso de compilado o interpretado del código fuente, respectivamente, comentados más arriba.

Referencias

[Lenguaje de programación](#)