
Entornos de Desarrollo

LENGUAJES DE PROGRAMACIÓN



CONCEPTO DE LENGUAJE DE PROGRAMACIÓN

“Un lenguaje de programación es un conjunto de instrucciones, operadores y reglas de sintaxis y semánticas, que se ponen a disposición del programador para que éste pueda comunicarse con los dispositivos de hardware y software existentes”.

El idioma que constituyen los operadores, instrucciones y reglas tiene el objetivo de facilitar la tarea de crear programas, permitiendo con un mayor nivel de abstracción realizar las mismas operaciones que se podrían realizar utilizando código máquina.

CLASIFICACIÓN DE LOS LENGUAJES DE PROGRAMACIÓN

Se pueden clasificar atendiendo a 3 criterios principales: nivel de abstracción, forma de ejecución y el paradigma utilizado.

- **Nivel de abstracción:** que se caracteriza por el modo en que los lenguajes se alejan del código máquina y se acercan a un lenguaje similar al nuestro.
- **Forma de ejecución:** que depende de cómo el programa se ejecuta dentro del sistema.
- **Paradigma de programación:** que es el estilo de programación utilizado.

CLASIFICACIÓN DE LOS LENGUAJES DE PROGRAMACIÓN

Según el nivel de abstracción:

- Lenguajes de bajo nivel o primera generación: se trata del **código máquina**. Cadenas interminables de secuencias de 1 y 0 que conforman las operaciones que la máquina puede entender sin interpretación alguna.
- Lenguajes de medio nivel o segunda generación: se trata del **lenguaje ensamblador**, donde se definen unas instrucciones para realizar operaciones sencillas con datos simples y posiciones de memoria. Alta dependencia de la máquina.

CLASIFICACIÓN DE LOS LENGUAJES DE PROGRAMACIÓN

- Lenguajes de alto nivel: que se dividen en 3 generaciones.
 - **Tercera generación**: la mayoría de los lenguajes utilizados actualmente que siguen el paradigma de programación orientada a objetos. Por ejemplo, Java o Python.
 - **Cuarta generación**: los lenguajes creados con un propósito específico por lo que permiten reducir la cantidad de líneas de código que necesitamos. Por ejemplo, MatLab o SQL.
 - **Quinta generación**: se trata de los lenguajes naturales, que abstraen utilizando una base de conocimientos que produce otro sistema. Utilizado en inteligencia artificial y lógica. Por ejemplo, Prolog o Lisp.

CLASIFICACIÓN DE LOS LENGUAJES DE PROGRAMACIÓN

Según la forma de ejecución:

- Lenguajes compilados: donde un programa traductor (compilador) convierte el código fuente en código objeto y otro programa (enlazador) une el código objeto del programa con el código objeto de las librerías necesarias, para con todo ello producir un único programa ejecutable. Por ejemplo, C y C++.
- Lenguajes interpretados: no se genera código objeto, sino que el código fuente es ejecutado directamente por un programa intérprete instalado en la máquina. Se diferencia del anterior en que se traducen en tiempo real solo las instrucciones que se utilizan en cada ejecución, en lugar de traducir todo el código, se utilice o no. Por ejemplo, Python.

CLASIFICACIÓN DE LOS LENGUAJES DE PROGRAMACIÓN

- Lenguajes intermedios o virtuales: son una mezcla de los anteriores, debido a que se genera un **bytecode** que no es código objeto, sino que es un código que puede ser interpretado por cualquier máquina que tenga instalado el interprete de bytecode. La desventaja reside en que la ejecución es más lenta, pero presentan la ventaja de ser multisistema. Por ejemplo, Java.

CLASIFICACIÓN DE LOS LENGUAJES DE PROGRAMACIÓN

Según el paradigma de programación:

- Paradigma imperativo: se compone de una secuencia de instrucciones.
- Paradigma declarativo: especifica un conjunto de premisas y condiciones para indicar que hay que hacer, pero no necesariamente como hacerlo.
- Paradigma procedimental: el programa se divide en partes más pequeñas, denominadas funciones y procedimientos, que pueden comunicarse entre si. Permite reutilizar código ya programado y solventa el problema del código '*spaghetti*'.
- Paradigma orientado a objetos: encapsula el estado y las operaciones en objetos, creando una estructura de clases y objetos que emula un modelo del mundo real. Los objetos realizan acciones e interactúan con otros objetos.

CLASIFICACIÓN DE LOS LENGUAJES DE PROGRAMACIÓN

- Paradigma funcional: evalúa el problema realizando funciones de manera recursiva, evitando declarar datos y haciendo hincapié en la composición de las funciones y las interacciones entre ellas.
- Paradigma lógico: define un conjunto de reglas lógicas para ser interpretadas mediante inferencias lógicas. Permite responder preguntas planteadas al sistema para resolver problemas.

TIPOS DE CÓDIGO

El código de nuestro programa se elabora en programas externos asociados al lenguaje de programación en el que escribiremos dicho programa y la arquitectura de las máquinas donde se ejecutará.

- Código fuente: el conjunto de instrucciones que escribiremos en un lenguaje de programación determinado.
- Código objeto: es el código resultado de compilar el código fuente, en el caso de lenguajes de programación compilados, pero en el caso de lenguajes de programación virtuales, será el bytecode.
- Código ejecutable: es el resultado de enlazar nuestro código objeto con las librerías.

COMPILACIÓN Y SUS FASES

Se define como el proceso de obtener el código ejecutable, pase tanto por un compilador como por un enlazador.

Dicho proceso de compilación se realiza siguiendo las siguientes fases:

- Análisis Lexicográfico: se leen de manera secuencial todos los caracteres de nuestro código fuente, buscando palabras reservadas, operaciones, caracteres de puntuación y agrupándolos en cadenas de caracteres denominados 'lexemas'.
- Análisis Sintáctico-semántico: el proceso sintáctico agrupa todos los componentes léxicos, analizados en la fase anterior, en forma de frases gramaticales. El proceso semántico revisa la coherencia de las frases gramaticales, es decir si el significado es correcto, si los tipos de datos son correctos, si los arrays tienen el tamaño y tipo adecuados, etc.

COMPILACIÓN Y SUS FASES

- Generación de código intermedio: se genera una representación intermedia a modo de pseudo-ensamblador con el objetivo de facilitar la tarea de traducir a código objeto.
- Optimización de código: revisa el código pseudo-ensamblador para optimizarlo con el objetivo de que el código resultante sea mas fácil y rápido de interpretar por la máquina.
- Generación de código: se genera el código objeto de nuestro programa en un código de lenguaje máquina.
- Enlazador de librerías: se enlaza el código objeto con el de las librerías necesarias, produciendo el código final o código ejecutable.

ACTIVIDAD

Crea una tabla donde clasifiques según el nivel de abstracción, la forma de ejecución y el paradigma de programación, los siguientes lenguajes de programación:

C, C++, Java, Python, Ruby, Go, Javascript, SQL, PHP y Prolog.