

Guía Teórica N°1

Introducción Teórica

¿Qué es un Lenguaje de programación?

Es un lenguaje formal que, mediante una serie de instrucciones, le permite a un programador escribir un conjunto de órdenes, acciones consecutivas, datos y algoritmos para, de esa forma, crear programas que controlen el comportamiento físico y lógico de una máquina.

Mediante este lenguaje se comunican el programador y la máquina, permitiendo especificar, de forma precisa, aspectos como:

- Cuáles datos debe operar un software específico;
- Cómo deben ser almacenados o transmitidos esos datos;
- Las acciones que debe tomar el software dependiendo de las circunstancias variables.

Para explicarlo mejor (en otras y con menos palabras), el lenguaje de programación es un sistema estructurado de comunicación, el cual está conformado por conjuntos de símbolos, palabras claves, reglas semánticas y sintácticas que permiten el entendimiento entre un programador y una máquina.

¿Qué tipos de lenguaje de programación existen?

El lenguaje de programación es la base para construir todas las aplicaciones digitales que se utilizan en el día a día y se clasifican en dos tipos principales: lenguaje de bajo nivel y de alto nivel.

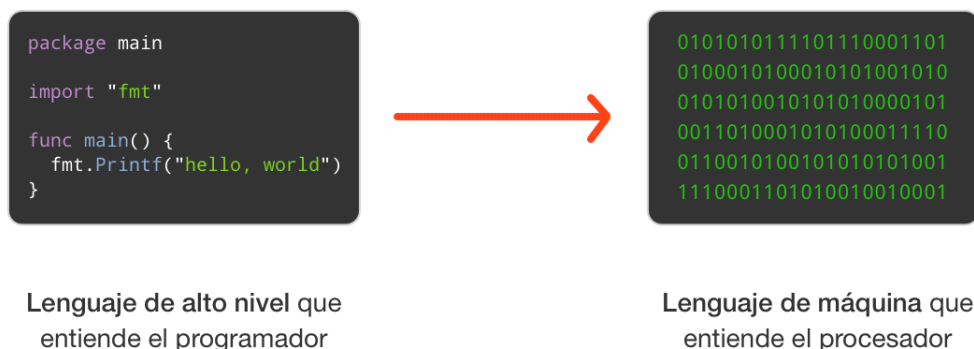
¡Continúa leyendo para aprender sobre ellos!

- **Lenguaje de bajo nivel:** Son lenguajes totalmente dependientes de la máquina, es decir que el programa que se realiza con este tipo de lenguajes no se puede migrar o utilizar en otras máquinas (lenguaje máquina o ensamblador)
- **Lenguaje de alto nivel:** Se tratan de lenguajes independientes de la arquitectura del ordenador. Por lo que, en principio, un programa escrito en un lenguaje de alto nivel, lo puedes migrar de una máquina a otra sin ningún tipo de problema. Estos lenguajes permiten al programador olvidarse por completo del funcionamiento interno de la maquina/s para la que están diseñando el programa. Tan solo necesitan un traductor que entiendan el código fuente como las características de la máquina (C/C++, Java, PYTHON, entre otros).

Lenguajes Compilados Vs Lenguajes Interpretados:

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

Lenguaje de máquina son las instrucciones que entiende el computador (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 que escribes 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 escribes el código y lo pruebas) es más rápido en un lenguaje interpretado. Eso se debe a que en lenguajes compilados es necesario realizar el proceso de compilación cada vez que cambias el código fuente, aunque con herramientas adicionales se puede automatizar.

Otra desventaja de un lenguaje compilado es que cuando compilas un programa debes crear ejecutables para cada uno de los sistemas operativos en los que lo vayas a utilizar. 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, debes tener 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.

¿Por qué aprender Python?

Python es uno de los lenguajes más demandados y de los más recomendados por varias razones, entre ellas su facilidad de uso y de aprendizaje. Además, resulta muy útil para introducirnos en la programación y se emplea en ámbitos más allá del desarrollo web o de la creación de aplicaciones, como por ejemplo el machine learning y el data science.

A continuación, repasamos varios motivos por los que deberías plantearte aprender Python, seas o no programador, quieras o no dedicarte a la programación.

- **Para iniciarse en programación:** La primera razón es obvia. Los principales elogios que se le hacen a Python hablan de sintaxis clara e intuitiva.

Además, es relativamente rápido de aprender. Hay quien considera que con 6 a 8 semanas, un programador medio puede aprender todo lo necesario para empezar con Python. También se recomienda iniciarse en programación con Python en lugar de con otros lenguajes como Java.

A todo esto, hay que añadir la gran cantidad de documentación disponible para aprender Python por nuestra cuenta y gratis.

- **Muchas salidas posibles:** La página oficial de Python nos da algunas pistas de qué podemos hacer con este lenguaje de programación: desarrollo web, desarrollo de software, administración de sistemas, entre otros.

- **Data science y machine learning:** Python y las librerías disponibles facilitan su aplicación a sectores como el data science o el machine learning. Con Python y librerías como Pandas, NumPy, SciPy y matplotlib es posible realizar tareas de análisis de información, obtención de datos, cleaning, refinamiento, generación de modelos (machine learning) y visualización de los datos obtenidos. La ciencia de datos o data science está muy relacionada con lo que conocemos como big data, y se aplica en la gran mayoría de sectores actuales, desde las finanzas a cualquier industria que puedas imaginar: automoción, aeronáutica, salud... Incluso se aplica a otros ámbitos como el deporte o la política.

Por su parte, el machine learning y el deep learning son otras expresiones de moda en los últimos años y tiene que ver con el aprendizaje automático de máquinas o algoritmos. Relacionado con la inteligencia artificial, esta disciplina consiste en que todo el trabajo lo realice la propia máquina (física o virtual), incluyendo el proceso de aprendizaje y adaptación para que tome sus propias decisiones. Algunas de sus aplicaciones son la detección de fraude financiero, la fijación de precios o la personalización de perfiles de clientes.

- **Automatización:** A la facilidad de uso de Python se le añade sus características de automatización que permiten aprovechar código mediante scripts y generar herramientas para realizar por nosotros tareas repetitivas.

Ámbitos como la gestión de bases de datos o la administración de redes o servidores encuentran en Python de gran ayuda para automatizar procesos con una porción de código que podemos escribir en unos minutos.

