

LENGUAJES DE PROGRAMACIÓN

AUTOR: WALTER MADRIGAL CHAVES

NOVIEMBRE: 2020



San Marcos

Contenido

INTRODUCCIÓN.....	2
LENGUAJES INTERPRETADOS	4
LENGUAJES COMPILADOS.....	4
ELECCIÓN DEL LENGUAJE.....	5
CONCLUSIONES Y RECOMENDACIONES	7
REFERENCIAS BIBLIOGRÁFICAS.....	7



INTRODUCCIÓN

Existe una paleta extensa con diferentes tipos de software en el mercado, estos se agrupan según su funcionalidad, por ejemplo, existe el software de sistema operativo, el software de aplicación y el software de desarrollo, este último de gran relevancia por su naturaleza de engendrar a otros.

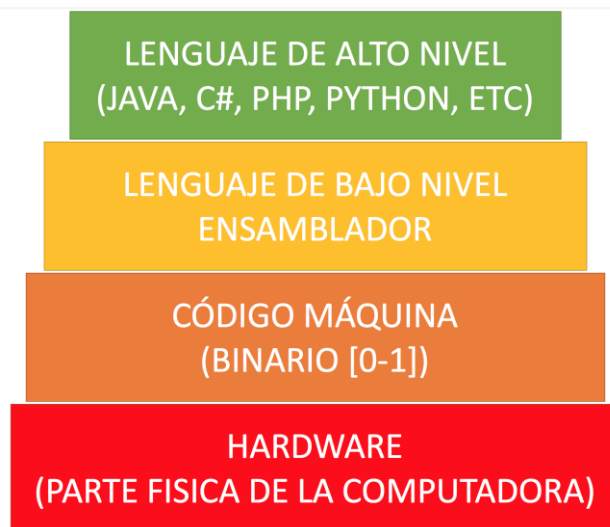
Un lenguaje de programación es un idioma que es entendible por las computadoras y que mediante instrucciones ejecutan tareas y procesos. Está compuesto por símbolos, reglas sintácticas y semánticas, expresadas en forma de instrucciones y relaciones lógicas, mediante las cuales se construye el código fuente de una aplicación.

En la siguiente lectura se analizará las diferentes categorías de lenguajes de programación, clasificados según la forma en que interpretan el código fuente, con la finalidad de establecer bases sólidas en una elección de un lenguaje en específico.

LENGUAJES DE BAJO NIVEL

Son lenguajes bastantes complejos porque su código está más cercano al código de la máquina que a los pseudocódigos. Para un desarrollo efectivo es necesario que el programador conozca en demasía los comandos, el hardware y como estos se interrelacionan con el sistema operativo.

Imagen 1 Tipos de lenguajes de programación



Fuente: <http://conogasi.org/>

La imagen anterior muestra las capas en que se distribuye la producción de software, en la base de la pila encontramos la parte física, el hardware, después encontramos el código de máquina que es un lenguaje basado en 1 y 0, posteriormente tenemos el lenguaje de bajo nivel llamado ensamblador que como ya mencionamos es un lenguaje de estructura compleja y por último los lenguajes de alto nivel, que analizaremos a continuación.

LENGUAJES DE ALTO NIVEL

Después de conocer las complicaciones que lenguajes como el Ensamblador genera, se crearon los lenguajes de alto nivel, estos tienen códigos, algoritmos y expresiones más cercanos al pseudocódigo, por ende, son más sencillos y

entendibles para los seres humanos. Los lenguajes de alto nivel procesan las instrucciones y las transforman en código entendible para las máquinas.

Imagen 2 Ciclo de interpretación de código



Fuente: Elaboración propia.

Como lo muestra la figura anterior, al desarrollar sistemas sobre plataformas de alto nivel, el código digitado es interpretado para generar nuevo código, este de bajo nivel y entendible por la computadora. Este proceso de interpretación de código provoca reducción de velocidad en todo el proceso.

LENGUAJES INTERPRETADOS

Los lenguajes de alto nivel se categorizan en dos ramas: los interpretados y los compilados. En los primeros las líneas de código son interpretadas en tiempo real, el encargado de realizar esa función es un intérprete que pasa línea a línea a código máquina, lo cual permite maximizar la eficiencia de los programas. Algunos ejemplos de estos tipos de programas son: Forth, Java, Python y Ruby.

Por lo general los lenguajes interpretados se utilizan para el desarrollo de aplicaciones o sitios web que van acompañados de framework (estructuras base, cargadas de rutinas preestablecidas, sirven como punto de partida para elaborar un proyecto) que facilitan en gran medida su programación.

LENGUAJES COMPILADOS

La otra cara de la moneda son los lenguajes compilados que al contrario del anterior que hace interacciones por línea, este toma todo el código fuente y lo traslada a un programa llamado compilador, el cual lo convierte a código

máquina únicamente después de que él verifique que no hay errores en su estructura. Algunos ejemplos de este tipo de lenguajes son: Basic, Delphi, C y Foxpro.

Este tipo de lenguajes los vemos más en software de escritorio ya que requieren de mayores recursos y de acceso a archivos determinados. También por el peso mayor que estos suelen tener en sus archivos ejecutables.

Tabla 1 Ventajas y desventajas de los lenguajes

Tipo de lenguaje	Ventajas	Desventajas
Lenguaje interpretado	<ul style="list-style-type: none"> -Poseen gran independencia de la plataforma donde se ejecutan. -Ocupa menos recursos. 	<ul style="list-style-type: none"> La interpretación línea a línea es más lenta que la compilación. Necesita un tercero que sirva como interprete las instrucciones
Lenguaje compilado	<ul style="list-style-type: none"> -Son más rápidos - Permite corregir todos los errores de codificación antes de obtener un programa completamente ejecutado 	<ul style="list-style-type: none"> Son muy dependientes de su plataforma. Son más pesados por sus ejecutables.

Fuente: Elaboración propia

ELECCIÓN DEL LENGUAJE

Es casi imposible estar al día con todos los lenguajes y sus variantes, de ahí



que la elección que realicemos para desarrollar un proyecto es muy importante para enfocar esfuerzos. Lo primero que se debe tener presente es el entorno que rodea el proyecto, ejemplo si es para una empresa con políticas establecidas para el desarrollo de sistemas o es un desarrollo independiente.

Otro punto muy importante es el tipo de software que vamos a desarrollar, anteriormente se mencionó que se recomienda la utilización de lenguajes interpretados para el desarrollo de aplicaciones y sitios web y lenguajes compilados para software de escritorio.

El licenciamiento o valor económico de utilización del lenguaje es muy importante, debido a que aumenta los costos del proyecto y puede convertirlo en poco viable. Recuerde que el mercado ofrece lenguajes gratis y de pago.

Por último, su conocimiento y experiencia en un lenguaje en específico suma en mucho a la hora de seleccionar una opción para el proyecto.

Para ahondar más en el tema de tipos de lenguajes de programación debe realizar la lectura de las páginas 95 a la 107 del libro: Lógica de programación (2017) de Omar Iván Trejos Buriticá.

CONCLUSIONES Y RECOMENDACIONES

- Los lenguajes de alto nivel llegaron para quedarse, su versatilidad y facilidad los convierten en los preferidos por muchos.
- Los lenguajes de interpretación son versátiles y ejecutables en distintas plataformas.
- Los lenguajes de compilación son más robustos y respaldos por plataformas estables.
- El crecimiento de las variantes de lenguajes de programación hace complicado conocerlas en detalle, es importante enfocar esfuerzos eligiendo inteligentemente.
- Estudie el mercado y el entorno para seguir su rumbo en los lenguajes de programación.

REFERENCIAS BIBLIOGRÁFICAS

Buriticá, O. T. (2017). *Lógica de programación*. Bogotá: Ediciones de la U.

José, G. E.-A. (2016). *Lógica: temas básicos*. Grupo Editorial Patria .

Morales, J. H. (2017). *Introducción a la lógica de programación*. Armenia: ELIZCOM S.A.A.



www.usanmarcos.ac.cr

San José, Costa Rica