Yessica Andrea Perez Machuca

[yp80905@gmail.com](mailto:yp80905@gmail.com)

Grupo:T2

DIFERENCIAS ENTRE COMPILADORES E INTÉRPRETES

| DIFERENCIAS | |
| --- | --- |
| COMPILADORES | INTÉRPRETES |
| Los programas que son compilados a un código máquina nativo suelen ser más rápido que los lenguajes interpretes. | Los lenguajes interpretes suelen ser más flexibles. |
| Traducen el código antes de ejecutar el software. | Ofrecen características como escritura dinámica y tamaño de programa más pequeño. |
| El código compilado se ejecuta más rápido. | Ejecutan el código fuente del programa ellos mismos,el código en sí en dependiente de la plataforma. |
| El compilador muestra todos los errores después de la compilación; | El intérprete muestra los errores cada línea uno por uno. |
| El compilador se basa en el modelo de carga y enlace de traducción. | El intérprete se basa en el método de interpretación. |

LENGUAJE DE TIPADO FUERTE

Es el utilizado para que el código incluya el tipo de dato al declarar la variable. Solemos decir que esta forma de programar es más expresiva al saber qué sucederá exactamente con las órdenes programadas. La ejecución es más veloz, ya que la inferencia de los tipos es anterior y el lenguaje no necesita verificarlos a la hora de ejecutarlos.

Suelen ser usados en lenguajes de programación como Python, C, C#, Java y Ruby.

LENGUAJE DE TIPADO DÉBIL

Se usa para indicar el tipo de variable a la hora de declarar. Permite asignarle un valor entero a una variable que con anterioridad tenía una cadena. En programación informática se usa en lenguajes como el JavaScript o PHP.

## Diferencias entre ambos tipados:

* El tipado débil te permite trabajar en menos tiempo y comprobar cuáles son los resultados del programa en tiempo real. Para la fase de ideación del proyecto esta opción es ideal. Se adapta muy bien a programas de menor capacidad.
* El tipado fuerte admite realizar operaciones con distintos tipos de variable. La consecuencia directa es que no vas a cometer tantos errores como con el lenguaje del punto anterior. Ahora bien, necesitas escribir más código. Este lenguaje es ideal para proyectos de mayor alcance y con un número mayor de especificaciones.