Freiler Aleiro Ortega Estupiñan

correo : [aleiroestupinan1999@gmail.com](mailto:aleiroestupinan1999@gmail.com)

grupo:t2

* **diferencias entre compilado y intérprete**

### **Lenguajes compilados**

Los lenguajes compilados son convertidos directamente a código máquina que el procesador puede ejecutar. Como resultado, suelen ser más rápidos y más eficientes al ejecutarse en comparación con los lenguajes interpretados. También le dan al desarrollador más control sobre aspectos del hardware, como la gestión de memoria y el uso del CPU.

Los lenguajes compilados necesitan un paso de compilación (*build*) - primero necesita compilarse manualmente. Necesitas "recompilar" el programa cada vez que necesites hacer un cambio. En nuestro ejemplo del hummus, la traducción completa está escrita antes de que la tengas. Si el autor original decide que quiere usar un distinto aceite de oliva, toda la receta necesitaría ser traducida de nuevo y reenviada a ti.

Algunos ejemplos de lenguajes compilados puros son C, C + +, Erlang, Haskell, Rust y Go.

### **LENGUAJES INTERPRETADOS**

Estos lenguajes ejecutan línea por línea el programa y a la vez ejecuta cada comando. Aquí, si el autor decide que quiere usar un distinto aceite de oliva, podría borrar el anterior y agregar el nuevo. Tu amigo traductor puede decirte ese cambio a medida que sucede.

Los lenguajes interpretados alguna vez fueron significativamente más lentos que los lenguajes compilados. Pero, con el desarrollo de la compilación justo a tiempo, esa diferencia se está reduciendo.

Ejemplos comunes de lenguajes interpretados son PHP, Ruby, Python y JavaScript.

En un lenguaje compilado, la máquina de destino traduce directamente el programa. En un lenguaje interpretado, el código fuente no es directamente traducido por la máquina de destino. En cambio, un programa distinto, mejor conocido como intérprete, lee el código y lo ejecuta.

* **LENGUAJE TIPADO FUERTE Y DÉBIL**

**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. Presenta desventajas como:

Es muy rápido de desarrollar, pero favorece la aparición de errores.

Su flexibilidad ayuda a que los programas funcionen sin necesidad de utilizar tanto código. No obstante, presenta errores en el tiempo de ejecución, como el que se pierde al comparar números mezclando algunos tipados como los strings, que no son detectables anteriormente.

Es necesario castear de manera periódica el programa para conseguir el resultado esperado. Ello provoca que tengas que redactar más código para aumentar tu seguridad.

## 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. Se usa en lenguajes de programación como Python, C, C#, Java y Ruby. Sus desventajas son:

* A la hora de programar es mucho más estricto y necesitas escribir más código, lo que alarga el tiempo de trabajo y aumenta la necesidad de verificar lo redactado más exhaustivamente.
* El compilador debe ejecutar una disciplina de tipos concreta. Es decir, se siguen unas normas concretas de asignación de variables, argumentos de funciones o valores de retorno.