

# Languages Analysis

Language	Type system	Learning curve	Null	Error handling	Concurrency	Immutability	Ecosystem	Speed	Overall score
C++	👉	👎	👎	👎	👎	👎	👎	👎	0.5
Java	👉	👎	👎	👎	👎	👎	👍	👉	2
Python	👎	👍	👎	👎	👎	👎	👍	👎	4
JavaScript	👎	👍	👎	👎	👉	👉	👍👍	👎	4.5

## C++

A pesar de que el nombre implique que C++ es una mejora de C, aun prefiero este ultimo para programar y la razón es que creo que C++ se esfuerza demasiado en hacer de todo. Se parece a C, pero usa el mismo paradigma que Java, lo que lo coloca en un extraño punto medio entre estos dos lenguajes. Una de sus debilidades es que su tiempo de compilación es lento, aun así, se desempeña bien en tiempo de ejecución; otro punto en contra son los mensajes de error que arroja el compilador pues son tan crípticos que incluso pueden generar frustración a los programadores mas experimentados. Aun así, es un lenguaje muy utilizado para el desarrollo de software, lo que significa que es fácil encontrar a alguien que brinde ayuda en caso de toparse con algún problema.

## Python

La verdad este es un lenguaje con el que no tenido gran contacto, aunque debo reconocer que tiene una sintaxis tan sencilla que hasta parece incorrecta, por esa razón se ha vuelto muy popular para enseñar conceptos básicos de programación. También por eso hay una gran cantidad de paquetes para extender las funciones de Python, muchas de ellas dedicadas a la ciencia de datos e inteligencia artificial. Aunque no muy útil si tu objetivo es que trabajar en entornos de bajo nivel, como los sistemas embebidos

## JavaScript

Originalmente diseñado para agregar interactividad a las paginas web, pero gracias a la baja curva de aprendizaje que tiene, este lenguaje de programación se ha vuelto el favorito de muchos programadores. Sus funcionalidades se han esparcido más allá del desarrollo web, permitiendo usarlo en ciencia de datos, desarrollo de software e inclusive “machine learning”

## Conclusión

Existen varios lenguajes de programación actualmente, tantos que la simple pregunta ¿con que lenguaje empiezo a programar? Resulta abrumador, y pareciera que solo hay dos grandes categorías: aquellos lenguajes que son fáciles de aprender y usar, pero que son ineficientes para cualquier trabajo que se realice a bajo nivel, y aquellos que son muy buenos para trabajar a bajo nivel pero que suelen ser difíciles de comprender o inclusive difíciles de leer.