**INSTITUTO POLITÉCNICO NACIONAL**

Escuela Superior de Cómputo

TAREA 2

¿POR QUÉ DEBERÍAN EXISTIR MÁS LENGUAJES DE PROGRAMACIÓN?

NAVA ROMO EDGAR ADRIÁN

Profesor: ROBERTO TECLA PARRA

Grupo: 3CM8

¿POR QUÉ DEBERÍAN EXISTIR MÁS LENGUAJES DE PROGRAMACIÓN?

El mundo de la programación puede ser fascinante, pero también puede llegar a agobiarte en ocasiones. Al igual que todo lo relacionado con la tecnología se trata de un ámbito en constante cambio. Existen una gran cantidad de lenguajes de programación que son mejorados día a día para adaptarse a las nuevas necesidades. Y no solo eso. A medida que se plantean nuevos retos y necesidades se crean también nuevos lenguajes específicos.

La lista de lenguajes de programación existentes es muy larga. Y aunque todos tienen particularidades que han de escribirse de forma diferente en cada código afortunadamente todos suelen contar con gran cantidad de cosas en común. Precisamente debido a esta gran variedad de lenguajes es fundamental tener claro cuáles son los más demandados actualmente por las empresas. No importa que sea para proyectos web, aplicaciones móviles o para programas de ordenador, lo fundamental es conocer los más utilizados si se quiere encontrar un hueco en el mundo profesional como programador. Es cierto que no podemos hablar de una lista definitiva o de una lista exacta que tomar como referente. Sin embargo, sí que se pueden distinguir algunos por su menor o mayor relevancia en función de los proyectos que pueden encontrarse actualmente.

Lo cierto es que en el ámbito de lenguajes de programación tenemos de todo, desde los que sirven para que nos partamos la caja hasta los más adecuados para iniciar a niños y adolescentes según su edad pasando, claro está, por los más populares del mercado actual. El único idioma que realmente comprende un ordenador es el código máquina, que básicamente son instrucciones que se envían al microprocesador y que se representa por valores numéricos. La primera forma de abstracción de esta programación fue el ensamblador-desensamblador que convertía texto a instrucciones y viceversa. Los ensambladores tenían una correspondencia uno a uno entre comando e instrucción máquina, por lo que seguía siendo el mismo lenguaje, pero adornado.

El caso es que la vida del programador es muy dura y programar en ensamblador era muy tedioso y muy difícil de detectar los errores, además, cada procesador tenía un juego de instrucciones distintas, por lo que no se podía escribir programas para distintos procesadores a la vez. Es por ello por lo que se decidió subir un peldaño más y definir lenguajes de alto nivel que, mediante una compilación, permitiesen generar código máquina para distintas máquinas y, además, fuese más sencillo de programar para un humano.

Entrando al tema, directamente, se puede decir que hay tantos lenguajes de programación como necesidades no bien satisfechas han tenido los programadores. Crear un lenguaje de programación no es más que definir una gramática, una semántica y una traducción de esta gramática a código máquina. Luego han surgido teorías, como la orientación a objetos, los lenguajes funcionales, los lógicos, los multitarea, etc. que lo que permiten es facilitar alguna parte del proceso de realizar los programas. Como da la casualidad de que los mismos programadores eran los que definían sus propios lenguajes, siempre se encontraba alguno insatisfecho con los lenguajes disponibles y decidía inventar uno.

Los llamados lenguajes imperativos, este tipo de lenguajes, cuyo origen está ligado a la propia arquitectura de Von Neumann, la arquitectura consta de una secuencia de celdas, llamadas memoria, en la cual se pueden guardar en forma codificada, lo mismo datos que instrucciones; y de un procesador, el cual es capaz de ejecutar de manera secuencial una serie de operaciones, principalmente aritméticas y booleanas, llamadas comandos. En general, un lenguaje imperativo ofrece al programador conceptos que se traducen de forma natural al modelo de la máquina. Los lenguajes imperativos más destacados de la historia han sido: FORTRAN, Algol, Pascal, C, Modula-2.

Lenguajes Funcionales, los matemáticos desde hace un buen tiempo están resolviendo problemas usando el concepto de función. Una función convierte ciertos datos en resultados. Si supiéramos cómo evaluar una función, usando la computadora, podríamos resolver automáticamente muchos problemas. Así pensaron algunos matemáticos, que no le tenían miedo a la máquina, e inventaron los lenguajes de programación funcionales. Además, aprovecharon la posibilidad que tienen las funciones para manipular datos simbólicos, y no solamente numéricos, y la propiedad de las funciones que les permite componer, creando de esta manera, la oportunidad para resolver problemas complejos a partir de las soluciones a otros más sencillos. También se incluyó la posibilidad de definir funciones recursivamente.

Lenguajes lógicos, otra forma de razonar para resolver problemas en matemáticas se fundamenta en la lógica de primer orden. El conocimiento básico de las matemáticas se puede representar en la lógica en forma de axiomas, a los cuales se añaden reglas formales para deducir cosas verdaderas a partir de los axiomas. Gracias al trabajo de algunos matemáticos, de finales de siglo pasado y principios de éste, se encontró la manera de automatizar computacionalmente el razonamiento lógico particularmente para un subconjunto significativo de la lógica de primer orden que permitió que la lógica matemática diera origen a otro tipo de lenguajes de programación, conocidos como lenguajes lógicos. También se conoce a estos lenguajes, y a los funcionales, como lenguajes declarativos, porque el programador, parar solucionar un problema, todo lo que tiene que hacer es describirlo vía axiomas y reglas de deducción en el caso de la programación lógica y vía.

Lenguajes concurrentes, paralelos y distribuidos, la necesidad de ofrecer concurrencia en el acceso a los recursos computacionales se remonta a los primeros sistemas operativos. Mientras que un programa realizaba una operación de entrada o salida otro podría gozar del tiempo del procesador para sumar dos números, por ejemplo. Aprovechar al máximo los recursos computacionales fue una necesidad apremiante, sobre todo en la época en que las computadoras eran caras y escasas, el sistema operativo tenía que ofrecer la ejecución concurrente y segura de programas de varios usuarios, que desde distintas terminales utilizaban un solo procesador, y así surgió la necesidad de introducir algunos conceptos de programación concurrente para programar los sistemas operativos.

Muchos de los lenguajes de programación que hoy en día conocemos y utilizamos en el desarrollo de nuestros proyectos, se desarrollaron con un propósito y función específica pues sus contemporáneos no satisfacían completamente la necesidad atacada, es por esta razón y la más atinada que deben desarrollarse nuevos lenguajes de programación constantemente, las necesidades de la sociedad van cambiando conforme van siendo resueltas. Nuevas tecnologías, nuevo hardware, lógicamente requieren nuevos paradigmas y nuevos lenguajes de programación. Si bien se pueden clasificar los lenguajes de programación a través de distintos parámetros, es necesario destacar que independientemente del paradigma que utilice, de la sintaxis, si es compilado o interpretado, cada uno de estos lenguajes da soporte a un nicho de problemas relacionados con un tipo de software. Los lenguajes de programación se crean definiendo el nivel de abstracción que se quiere obtener, el tipo de aplicación al cual va dirigido y el alcance que se espere de este, por más optimista que llegué a ser. Entre más lenguajes de programación existan, potenciales programadores conocerán el lenguaje ideal que se acople a sus proyectos y aficiones.