



Universidad Nacional Autónoma de México

Facultad de Ingeniería

Fundamentos de Programación

Tarea #2| Historia de la Programación

Ramírez Pérez Daniela Itzel

Fecha (2/10/2020)

▪ **Breve historia del Cómputo**

En la historia del cómputo hay varios momentos clave en donde, con la ayuda de varias personas, se empezó a crear e innovar esta tecnología, De esto se nos quiere dar a entender que hay dos etapas en los avances científico-tecnológicos con el surgimiento de la idea y como esta idea se fue innovado durante los siglos.

La idea empieza a surgir desde el momento que el hombre tiene la necesidad de contar y esto viene con la invención de la escritura, diferentes civilizaciones encontraron sendas maneras de contar y más tarde, de efectuar operaciones.

Algunas de las escrituras creadas fueron:

- **La egipcia** adoptó un sistema de contar de diez en diez, de manera que marcaban las unidades con líneas verticales y las decenas con un signo similar a la U invertida.
- **La babilonia** contaba de 60 en 60, y aunque su sistema era muy poco práctico, se cree que de ellos provienen las bases del sistema actual de contar los minutos y los segundos.
- **La romana** era muy simple y permitía efectuar operaciones aritméticas, aunque no con mucha sencillez.
- **La maya** contó con un sistema numeral vigesimal muy avanzado y utilizaron el cero aun antes que en Europa o Asia.
- **La arábica** es la forma en que contamos actualmente, utilizando el sistema decimal, esta forma que hemos heredado de nuestros antepasados ha sobrevivido por ser simple y adecuada para nosotros debido a que tenemos diez dedos en las manos

El siguiente paso fue cuando se quiso dar la innovación de artefactos para efectuar operaciones matemáticas. Los artefactos que se construyeron fueron:

- **El ábaco** es la primera calculadora o computadora de que tenemos noticias. Se inventó hace aproximadamente 4 000 años, y aún tiene un uso muy amplio en algunos países de Asia, debido a que se trata de una calculadora muy rápida y práctica, útil para efectuar operaciones de adición, substracción, multiplicación y división.
- **Huesos de Napier o tablas de multiplicar** inventados en 1617 por el escocés John Napier inventa los. Consistía en una regla fija y otra móvil que se desliza sobre la primera, de manera que deja ver el producto de dos números cualesquiera. Posteriormente las tablas de Napier evolucionaron hasta llegar a la regla de cálculo.
- **Pascalina** inventada en 1642 por Blaise Pasca era una máquina sumadora mecánica con el objeto de facilitar los cálculos de su padre. Esta máquina consistía en un sistema con ruedas pequeñas acopladas entre sí y que, de derecha a izquierda correspondían a unidades, decenas,

centenas, etcétera. Estas ruedas se encontraban divididas en 10 partes iguales; de manera que para efectuar una suma se hacían girar manualmente un número de pasos acorde con el número que se deseaba introducir. La rotación completa de un círculo en la dirección positiva causaba automáticamente que el círculo a su izquierda avanzara una posición. Para efectuar sustracciones el proceso era el inverso.

Los siguientes avances significativos se logran hasta el siglo XIX, cuando Joseph Jacquard introduce la idea de programar máquinas mediante el uso de tarjetas perforadas. La invención de Jacquard consistió en un telar que utilizaba tarjetas perforadas para controlar de manera automática el diseño y los colores de los tejidos. Esta idea es adaptada a la computación, en la llamada "máquina analítica" por matemático Charles Babbage, quien vivió obsesionado con el diseño y construcción de máquinas calculadoras.

Babbage hizo su primer esbozo de una máquina calculadora a la que llamó "máquina diferencial", en 1822

La historia reciente de la computación se divide en *generaciones* de computadoras, cada una de las cuales está caracterizada por un desarrollo o una innovación importante:

➤ *Primera generación (Bulbos)*

Ésta es la era de las computadoras construidas con bulbos; se inicia en 1951 con la primera computadora industrial, la llamada UNIVAC 1. Las computadoras de esta generación eran muy grandes y de funcionamiento costoso. Los bulbos eran de gran tamaño y consumían mucha energía eléctrica, razón por la cual el calor que producían las fundía.

➤ *Segunda generación (el transistor)*

La 2da generación empezó a finales de los años cincuenta, con el remplazo de los bulbos por transistores. Los transistores son pequeños dispositivos electrónicos útiles para generar, amplificar y controlar señales eléctricas generaban menos energía para su operación. En esta época comenzó el auge de las computadoras desde el punto de vista comercial, se crearon muchas compañías dedicadas a su diseño y construcción.

➤ *Tercera generación: (las familias de computadoras y los circuitos integrados)*

En la 3ra generación se distingue por dos aspectos importantes: Las computadoras incluyen circuitos integrados (artefacto que contiene un alto número de transistores y otros componentes integradas e interconectadas), y forman de *familias* de computadoras (las familias fueron creadas en un intento

por lograr compatibilidad entre máquinas de modelos diferentes, construidas por un mismo fabricante).

➤ *Cuarta generación: (El microprocesador)*

La 4ta generación se caracteriza por el uso del microprocesador. Éste consiste en un solo circuito integrado, el cual contiene en su totalidad a la unidad central de procesamiento (CPU), o cerebro de la computadora

➤ *Quinta generación: (Procesamiento en paralelo, redes neuronales y computadoras ópticas)*

Esta nueva generación seguramente incluirá dos cambios importantes. El primero consiste en un cambio fundamental en la manera de efectuar el procesamiento de la información, y por tanto, implica el diseño de arquitecturas conceptualmente diferentes. El segundo cambio que consiste en el desarrollo y adaptación de nuevas tecnologías. Las nuevas tecnologías que se están desarrollando son las siguientes:

- **Las supercomputadoras** se han creado para llevar a cabo el procesamiento de la información *en paralelo*; esto es, constan de varios elementos procesadores, cada uno de los cuales efectúa tareas de manera independiente y simultánea. De esta forma, el trabajo se divide en subtrabajos independientes, los cuales son asignados a diferentes elementos procesadores.
- **Las redes neuronales**, se utilizan elementos electrónicos que emulan neuronas simplificadas, los cuales se conectan entre sí formando redes similares a las que se encuentran en el cerebro, aunque a una escala mucho menor.
- **Las computadoras ópticas**, utilizarán rayos láser, lentes, y fibras ópticas para transportar a los fotones a lo largo de trayectorias no rectas. Lo cual significa que si estos materiales se mantienen por debajo de una cierta temperatura no producirán calor.

▪ **Historia de la programación**

la programación está vinculada a la aparición de los computadores, y los lenguajes tuvieron también su evolución. Inicialmente se programaba en código binario, es decir en cadenas de 0s y 1s, que es el lenguaje que entiende directamente el computador, tarea extremadamente difícil; luego se creó el lenguaje ensamblador, que aunque era lo mismo que programar en binario, al estar en letras era más fácil de recordar. Posteriormente aparecieron lenguajes de alto nivel, que en general, utilizan palabras en inglés, para dar las órdenes a seguir, para lo cual utilizan un proceso intermedio entre el lenguaje máquina y el nuevo código llamado código fuente, este proceso puede ser un compilador o un intérprete.

Un compilador lee todas las instrucciones y genera un resultado; un intérprete ejecuta y genera resultados línea a línea. En cualquier caso, han aparecido nuevos lenguajes de programación, unos denominados estructurados y en la actualidad en cambio los lenguajes orientados a objetos y los lenguajes orientados a eventos.

▪ **Lista de los lenguajes de programación**

- **Java** Desarrollado en 1995 por Sun Microsystems, su principal característica es que el lenguaje puede usarse prácticamente para todo, desde aplicaciones webs a programas de ordenadores, apps de Android, juegos, etc. Otra de sus ventajas es que es independiente del hardware. Solo tienes que escribir la aplicación una vez, y funcionará en cualquier ordenador o móvil, mediante un intérprete de Java.
- **JavaScript** es el lenguaje más utilizado para escribir aplicaciones dentro de las páginas webs. Funciona directamente en el navegador, así que lo único que necesita para usar JavaScript es escribir el programa como un documento de texto, y abrirlo en el navegador como si fuese una página web.
- **Go** es un lenguaje que se divide en una serie de componentes o procedimientos, que se pueden utilizar en diferentes partes del programa. Este tipo de programación ayuda a localizar los errores, y permite crear procedimientos muy básicos que se pueden usar en muchas aplicaciones.
- **Kotlin** se utiliza para desarrollar aplicaciones Android, desarrollo web, aplicaciones científicas, y apps multiplataforma que van a funcionar tanto en iOS como en Android. El ambicioso objetivo de Kotlin es sustituir a Java, ofreciendo un lenguaje que puede hacer lo mismo, pero de forma más sencilla.
- **Python** es el lenguaje más utilizado en las finanzas, o cuando hay que realizar cálculos y aplicar fórmulas, o desarrollar interfaces. En su contra, que es un poco lento a la hora de ejecutar tareas, con respecto a otros lenguajes. Se adapta muy bien a las necesidades de un tipo de inteligencia artificial llamada aprendizaje de máquinas o aprendizaje automático, así que hoy en día tiene una gran demanda.
- **Lenguaje C / C++** es un lenguaje que se creó para diseñar sistemas operativos, como Unix y otros muchos. Pero al tratarse de un lenguaje muy rápido y eficiente también comenzó a usarse para escribir aplicaciones y programas.

- **Scala** se ejecuta en la máquina virtual de Java, y puede usar los miles de librerías de Java y JavaScript, así que resulta muy sencillo incorporarlo a las actualizaciones de aplicaciones ya existentes, para integrarlo en cualquier proyecto.
- **Ruby** es un lenguaje para principiantes, porque al contrario que otros, es un lenguaje orientado a los humanos, y no al hardware. Es decir, intenta que sea divertido de aprender y usar.
- **Swift** es un lenguaje de programación creado por Apple para diseñar apps para iOS, Mac, el Apple TV y el Apple Watch. Es el lenguaje *oficial* de la plataforma Apple, así que necesitas aprenderlo si quieres crear apps para estos sistemas. Es rápido y fácil de usar, y lógicamente optimizado para crear aplicaciones para iOS. Además, se puede integrar sin problemas el código escrito en Objective-C.
- **Dart** es otro lenguaje creado por Google, con el objetivo de superar algunas de las limitaciones de JavaScript. Se ha vuelto muy popular porque es un lenguaje pensado para desarrollar interfases de usuario: menús, botones de acceso, y otros comandos de entrada que son imprescindibles en las apps de móvil y en las páginas web.

▪ **Clasificación de los lenguajes de programación**

- **Lenguajes de bajo nivel.** Se trata de lenguajes de programación que están diseñados para un hardware específico y que por lo tanto no pueden migrar o exportarse a otros computadores. Sacan el mayor provecho posible al sistema para el que fueron diseñados, pero no aplican para ningún otro.
- **Lenguajes de alto nivel.** Se trata de lenguajes de programación que aspiran a ser un lenguaje más universal, por lo que pueden emplearse indistintamente de la arquitectura del hardware, es decir, en diversos tipos de sistemas. Los hay de propósito general y de propósito específico.
- **Lenguajes de nivel medio.** Este término no siempre es aceptado, que propone lenguajes de programación que se ubican en un punto medio entre los dos anteriores: pues permite operaciones de alto nivel y a la vez la gestión local de la arquitectura del sistema.

▪ **Tipos de paradigmas de programación**

- **Lenguajes imperativos.** Menos flexibles, dada la secuencialidad en que construyen sus instrucciones, estos lenguajes programan mediante órdenes condicionales y un bloque de comandos al que retornan una vez llevada a cabo la función.

✓ *Programación estructurada* El enfoque estructurado de programación constituye una forma simplificada de programación imperativa. La principal

modificación del principio básico radica en que, en lugar de instrucciones que provocan que el procesamiento no continúe con la siguiente instrucción, sino en otro lugar este paradigma de programación de software prevé el uso de bucles y estructuras de control.

✓ *Programación procedimental* El paradigma de programación procedimental amplía el enfoque imperativo al desglosar algoritmos en porciones manejables. Estos se denominan procedimientos, subprogramas, rutinas o funciones. El sentido y el propósito de esta distribución es hacer que el código de programa sea más claro y evitar las repeticiones innecesarias de código.

✓ *Programación modular* Adapta a los requerimientos de proyectos de software mayores y más amplios. En este sentido, el código fuente se divide específicamente en bloques parciales lógicos independientes los unos de los otros para proporcionar más transparencia y facilitar el proceso de *debugging* (resolución de errores).

➤ **Lenguajes funcionales.** También llamados procedimentales, estos lenguajes programan mediante funciones que son invocadas conforme a la entrada recibida, que a su vez son resultado de otras funciones.

✓ *Programación funcional* Los enfoques funcionales del desarrollo de software cuentan con funciones de una clase especial: Un programa de programación funcional consta de llamadas de función concatenadas en las que cada parte del programa se interpreta como una función. El tratamiento especial de funciones permite a los programadores que trabajan de forma funcional componer y aplicar nuevas normas de cálculo de gran alcance a partir de funciones.

✓ *Programación lógica* El paradigma de software lógico se basa en la lógica matemática. En lugar de una sucesión de instrucciones, un software programado según este principio contiene un conjunto de principios que se pueden entender como una recopilación de hechos y suposiciones. Todas las solicitudes al programa se procesan de forma que el intérprete recurre a estos principios y les aplica reglas definidas previamente para alcanzar el resultado deseado.

Fuentes de referencia:

- Castrillón Viana L. (1995). Historia de la Computación. Octubre 2,2020, de Instituto Latinoamericano de la Comunicación Educativa. Sitio web:
http://bibliotecadigital.ilce.edu.mx/sites/ciencia/volumen2/ciencia3/088/html/sec_5.html
- Cervantes N., Pineda C. (2017). Fundamentos de Programación. Octubre 2, 2020, de FunPrOO Sitio web:
http://www.utn.edu.ec/reduca/programacion/fundamentos/un_poco_de_historia.html
- Raffino, M. R. (2020, 11 julio). *Lenguaje de Programación*. Concepto.de.
<https://concepto.de/lenguaje-de-programacion/>
- Pascual, J. A. (2020, 9 mayo). *Los lenguajes de programación más populares del mundo*. ComputerHoy.
<https://computerhoy.com/listas/industria/lenguajes-programacion-mas-populares-633547>
- IONOS Inc. (2020, 22 mayo). *Paradigmas de programación*. IONOS Digital Guide.
<https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/paradigmas-de-programacion/>