

# Tarea 1 Historia de la programacion

Miguel Angel Aguirre Olvera

31 de agosto de 2018

Gottfried Wilheml von Leibniz (1646-1716), quien aprendió matemáticas de forma autodidacta (método no aconsejable en programación) construyó una máquina similar a la de Pascal, aunque algo más compleja, podía dividir, multiplicar y resolver raíces cuadradas.

Pero quien realmente influyó en el diseño de los primeros computadores fue Charles Babbage (1793-1871). Con la colaboración de la hija de Lord Byron, Lady Ada Countess of Lovelace (1815-1852), a la que debe su nombre el lenguaje ADA creado por el DoD (Departamento de defensa de Estados Unidos) en los años 70. Babbage diseñó y construyó la "máquina diferencial" para el cálculo de polinomios. Más tarde diseñó la "máquina analítica" de propósito general, capaz de resolver cualquier operación matemática. Murió sin poder terminarla, debido al escepticismo de sus patrocinadores y a que la tecnología de la época no era lo suficientemente avanzada. Un equipo del Museo de las Ciencias de Londres, en 1991, consiguió construir la máquina analítica de Babbage, totalmente funcional, siguiendo sus dibujos y especificaciones. Un hito importante en la historia de la informática fue-

ron las tarjetas perforadas como medio para "alimentar" los computadores. Lady Ada Lovelace propuso la utilización de las tarjetas perforadas en la máquina de Babbage. Para que se enteren todos esos machistas desaprensivos, el primer programador/a fue una mujer. En 1880 el censo en Estados Unidos tardó más de 7 años en realizarse. Es obvio que los datos no eran muy actualizados. Un asistente de la oficina del censo llamado Herman Holle-rit (1860-1929) desarrolló un sistema para automatizar la pesada tarea del censo. Mediante tarjetas perforadas y un sistema de circuitos eléctricos, capaz de leer unas 60 tarjetas por minuto realizó el censo de 1890 en 3 años ahorrando tiempo y dinero. Más tarde fundó la Tabulating Machine Company y en 1924 tras alguna que otra fusión nació la Internacional Bussines Machines, IBM. ¿Os suena ? Las computadoras de hoy en día se sustentan en la lógica matemática basada en un sistema binario. Dicho sistema se implementa sobre dispositivos electrónicos que permiten, o no, pasar la corriente, con lo que se consiguen los 2 estados binarios: 0 y 1. A mediados del siglo XX, cuando se empezaron a construir

las primeras computadoras digitales, se utilizaban tubos de vacío para implementar los 2 estados binarios, pero ¿ cómo aparecieron estos conceptos ? Alan Mathison Turing (1912-1954) diseñó una calculadora universal para resolver cualquier problema, la "máquina de Turing". Tuvo mucha influencia en el desarrollo de la lógica matemática. En 1937 hizo una de sus primeras contribuciones a la lógica matemática y en 1943 plasmó sus ideas en una computadora que utilizaba tubos de vacío. George Boole (1815-1864) también contribuyó al álgebra binaria y a los sistemas de circuitos de computadora, de hecho, en su honor fue bautizada el álgebra booleana.

La primera computadora digital electrónica patentada fue obra de John Vincent Atanasoff (1903-1995). Conocedor de las inventos de Pascal y Babbage, y ayudado por Clifford Berry (1918-1963), construyó el Atanasoff Berry Computer (ABC). El ABC se desarrolló entre 1937 y 1942. Consistía en una calculadora electrónica que utilizaba tubos de vacío y estaba basada en el sistema binario (sistema numérico en el que se combinan los valores verdadero y falso, o 0 y 1).

Entre 1939 y 1944, Howard Aiken (1900-1973) de la universidad de Harvard en colaboración con IBM desarrolló el Mark 1. Era una computadora electromecánica de 16 metros de largo y más de dos de alto. Tenía 700.000 elementos móviles y varios centenares de kilómetros de cables. Podía realizar las cuatro operaciones básicas y trabajar con información almacenada en forma de tablas.

Por desgracia, los avances tecnológicos suelen producirse gracias a los militares que se aprovechan de la ciencia para perfeccionar sus armas. En la Moore School de la Universidad de Pensilvania se estaba trabajando en un proyecto militar para realizar unas tablas de tiro para armas balísticas. Los cálculos eran enormes y se tardaban semanas en realizarlos. Parece ser que John W. Mauchly (1907-1980), quien dirigía el departamento de física del Ursine College de Filadelfia vivió en casa de Atanasoff durante cuatro días a partir del 13 de Junio de 1941, lo que seguramente aprovechó para conocer las ideas de Atanasoff. Junto a John Presper Eckert (1919-1995), Mauchly desarrolló una computadora electrónica completamente operacional a gran escala, para acelerar los complicados cálculos del proyecto militar de la universidad Moore. Se terminó en 1946 y se llamó Electronic Numerical Integrator And Computer (ENIAC). El ENIAC tenía 18.000 tubos electrónicos integrados en un volumen de 84 metros cúbicos. Pesaba unas 30 toneladas y consumía alrededor de 100.000 vatios. Su capacidad de cálculo era de 5.000 operaciones por segundo, aunque tenía que programarse manualmente conectándola a 3 tableros que contenían más de 6000 interruptores. Cargar un programa podía ser una tarea de varios días. El calor disipado por semejante monstruo debía ser importante, y se necesitaba una instalación de aire acondicionado. En definitiva, un ordenador portátil... más o menos.

Puede que no os suene, pero quien conozca de "los entresijos de la

informática”seguro que considera importante nombrar a Johann Ludwig Von Neumann (1903-1957), genio de las matemáticas, quien tuvo el honor de asistir a las clases de Albert Einstein en la universidad de Berlín. Autor de trabajos de lógica simbólica, matemática pura y aplicada, física y tecnología, publicó un artículo acerca del almacenamiento de los programas, en 1945. Proponía que los programas se guardaran en memoria al igual que los datos, en forma binaria. Esto tuvo como consecuencia el aumento de velocidad de los cálculos y la ausencia de errores producidos por fallos mecánicos al programar la máquina mediante cables.

En cuanto a la aparición de los lenguajes de programación, el archiconocido COBOL, que tantos problemas causó con el “efecto 2000”, fue el primer lenguaje en el que no había que programar directamente en código binario, y fue Grace Murray Hopper en 1952, una oficial de la Marina de Estados Unidos desarrolló el primer compilador, un programa que puede traducir enunciados parecidos al inglés en un código binario comprensible para la máquina llamado COBOL (COmmon Business-Oriented Language) A partir de ahí, los avances han sido vertiginosos.

La utilización del transistor en las computadoras en 1958, sustituyendo los tubos de vacío La aparición del circuito integrado de mano de Jack Kilby, también en 1958 La miniaturización de un circuito electrónico en un chip de silicio en 1961 El primer microprocesador, el 4004 de Intel, en 1971 Gary Kildall crea el sistema operativo CP/M en 1973 IBM

comercializa el primer PC en 1980

Recordando a los primeros tiempos del ENIAC, con enormes computadores, en 1998 se terminó el proyecto Blue Pacific. La “maquinita” tiene la nada despreciable cantidad de 5856 procesadores que en conjunto tienen una velocidad de 3’9 teraflops, 2’6 Terabytes de memoria, ocupa 2400 metros cuadrados y tiene un peso de 47 toneladas. Se utiliza para la simulación de explosiones nucleares, y “ha salido” por unos 13000 millones de pesetas... baratito.

Hay muchos más personajes que intervienen en la historia y que han realizado grandes aportaciones, pero no es cuestión de extenderse.

Los primeros lenguajes de programación preceden a la computadora moderna. En un inicio los lenguajes eran códigos.

La máquina del telar de Jacquard, creada en 1801, utilizaba los orificios en tarjetas perforadas para representar los movimientos de un brazo de la máquina de tejer, con el objetivo de generar patrones decorativos automáticamente.

Durante un período de nueve meses entre 1842 y 1843, Ada Lovelace tradujo las memorias del matemático italiano Luigi Menabrea acerca de la nueva máquina propuesta por Charles Babbage, la Máquina Analítica. Con estos escritos, ella añadió unas notas en las cuales especificaba en detalle un método para calcular los números de Bernoulli con esta máquina, el cual es reconocido por muchos historiadores como el primer programa de computadora del mundo.

Herman Hollerith se percató de

que podía codificar la información en tarjetas perforadas cuando observó a los conductores de trenes que identificaban a los pasajeros según el orificio que hacían en su respectivo ticket. En 1890 Hollerith codificó los datos del censo en tarjetas perforadas.

Los primeros códigos de computadora estaban especializados según sus aplicaciones. En las primeras décadas del siglo 20, los cálculos numéricos estaban basados en los números decimales. Eventualmente se fueron dando cuenta que la lógica podía ser representada con números, no sólo con palabras. Por ejemplo, Alonzo Church fue capaz de expresar el cálculo lambda a través de fórmulas. La máquina de Turing estableció las bases para almacenar programas como datos en la arquitectura de von Neuman de una computadora. Sin embargo, a diferencia del cálculo lambda, el código de Turing no serviría satisfactoriamente como base para lenguajes de más alto nivel: su principal uso es en el análisis riguroso en la complejidad algorítmica.

Como muchos “primeros” en la historia, el primer lenguaje de programación moderno es difícil de identificar. Desde un inicio, las restricciones de hardware definían el lenguaje. Las tarjetas perforadas permitían 80 columnas, pero algunas de estas serían utilizadas para una clasificación de cada tarjeta. FORTRAN incluía algunas palabras reservadas provenientes del Inglés, como “IF”, “GOTO” (go to) y “CONTINUE”. El uso del tambor magnético para la memoria implicaba que los programas informáticos tuvieran que estar intercalados con las rotaciones del

tambor. Por lo tanto los programas eran muy dependientes del hardware.

Para algunas personas, lo que sería el primer lenguaje de programación moderno depende de cuánto poder y legibilidad humana se requería antes de que se concediera el estado de “lenguaje de programación”. Tanto el telar de Jacquard como la Máquina Diferencial de Babbage, tenían lenguajes muy simples y extremadamente limitados para describir las acciones que estas máquinas realizaran. Se puede incluso considerar los agujeros perforados en los rollos de pianola como un limitado lenguaje de dominio específico, a pesar de no estar diseñado para el consumo humano. En la década de 1940 fueron creadas las primeras computadoras modernas, con alimentación eléctrica. La velocidad y capacidad de memoria limitadas forzaron a los programadores a escribir programas, en lenguaje ensamblador muy afinados. Finalmente se dieron cuenta de que la programación en lenguaje ensamblador requería de un gran esfuerzo intelectual y era muy propensa a errores.

En 1948, Konrad Zuse publicó un artículo acerca de su lenguaje de programación Plankalkül. Sin embargo, no fue implementado en su vida y sus contribuciones fueron aisladas de otros desarrollos.

Entre algunos lenguajes importantes que fueron desarrollados en este período se encuentran:

1943 - Plankalkül (Respetad a Conrado), diseñado, pero sin implementar durante medio siglo 1943 - se crea el sistema de codificación

ENIAC. 1949 - 1954 - una serie de conjuntos de instrucciones ne-motécnicas, como la de ENIAC, comenzando en 1949 con C-10 para BINAC (que luego evolucionaría en UNIVAC). Cada conjunto de instrucciones estaba destinado a un fabricante específico.

### Las décadas de 1950 y 1960

En los cincuenta, los tres primeros lenguajes de programación modernos, cuyos descendientes aún continúan siendo utilizados, son:

FORTRAN (1955), creado por John Backus. LISP (1958), creado por John McCarthy. COBOL (1959), creado por el Short Range Committee, altamente influenciado por Grace Hopper.

Otro hito a finales de 1950 fue la publicación, por un comité Americano y Europeo de científicos de la computación, de un nuevo “lenguaje para algoritmos”; el Reporte de ALGOL 60 (.ALGO<sup>r</sup>ithmic Language”). Este reporte consolidó muchas ideas que estaban circulando en aquel entonces, y proporcionó dos innovaciones importantes para los lenguajes de programación:

Estructuras de bloques anidadas: las secuencias de código y las declaraciones asociadas se pueden agrupar en bloques sin tener que pertenecer explícitamente a procedimientos separados; Ámbito léxico: un bloque puede tener sus propias variables, procedimientos y funciones, invisible al código fuera de dicho bloque, por ejemplo, ocultamiento de información.

Otra innovación, relacionada con

esto, fue cómo el lenguaje fue descrito:

Una notación matemática exacta, Backus-Naur Form (BNF), fue utilizada para describir la sintaxis del lenguaje. Todos los subsecuentes lenguajes de programación han utilizado una variante de BNF para describir la porción libre del contexto de su sintaxis.

Algol 60 influenció particularmente en el diseño de lenguajes posteriores, de los cuales algunos se hicieron más populares. Los grandes sistemas de Burroughs fueron diseñados para ser programados en un subconjunto extendido de Algol.

Las ideas fundamentales de Algol se continuaron, produciendo Algol 68:

la sintaxis y la semántica se hizo aún más ortogonal, con rutinas anónimas, un sistema de tipificación recursiva con funciones de orden superior, etc.; y no sólo la parte libre del contexto, sino que tanto la sintaxis como la semántica del lenguaje completo fueron definidos formalmente, en términos de una gramática de Van Wijngaarden, un formalismo diseñado específicamente para este propósito.

Las variadas pero poco usadas características de Algol 68 (por ejemplo, bloques simultáneos y paralelos) y su complejo sistema de atajos sintácticos y coerciones automáticas de tipo lo hicieron impopular entre los ejecutores y se ganó una reputación de ser difícil. Niklaus Wirth salió del comité de diseño para crear el sencillo lenguaje Pascal.

### 1968-1979: estableciendo

### **paradigmas fundamentales**

El período comprendido entre finales de 1960 y finales de 1970 trajo un gran florecimiento de lenguajes de programación. La mayoría de los paradigmas de lenguajes de programación más importantes y actualmente en uso se inventaron en este período:

Simula, inventado en la década de 1960 por Nygaard y Dahl como un superconjunto de Algol 60, fue el primer lenguaje diseñado para apoyar la programación orientada a objetos. C, en principio un lenguaje de programación de sistemas, fue desarrollado por Dennis Ritchie y Ken Thompson en los Laboratorios Bell entre 1969 y 1973. Smalltalk (mediados de los 70) proporcionaron un completo diseño de un lenguaje orientado a objetos. Prolog, diseñado en 1972 por Colmerauer, Roussel y Kowalski, fue el primer lenguaje de programación lógica. ML construyó un sistema de tipos polimórfico (inventado por Robin Milner en 1973) en el tope de Lisp, pionero en los lenguajes funcionales de programación con tipado estático.

Cada uno de estos lenguajes generó toda una familia de descendientes, y los lenguajes más modernos cuentan al menos uno de ellos en su ascendencia.

En los años 60 y 70 también suscitó un debate considerable sobre los méritos de la "programación estructurada", que esencialmente significaba programación sin el uso de GOTO. Este debate estaba estrechamente relacionado con el diseño del lenguaje: algunos lenguajes no incluían GOTO, lo que obligó a la programación estructurada en el programa-

dor. Aunque el debate se recrudecía acaloradamente en ese momento, casi todos los programadores están de acuerdo en que, incluso en lenguajes que ofrecen GOTO, es una mala práctica de programación usarlo, excepto en raras circunstancias. Como resultado de ello, las generaciones posteriores de diseñadores de lenguajes han encontrado el debate sobre programación estructurada tedioso e incluso desconcertante.

### **La década de 1980: consolidación, módulos, rendimiento**

La década de 1980 fueron años de consolidación relativa en los lenguajes imperativos. En vez de inventar nuevos paradigmas, se comenzó a trabajar a partir de las ideas inventadas en la década anterior. C++ combinaba la programación orientada a objetos y la programación de sistemas. El gobierno de Estados Unidos estandarizó Ada, un lenguaje de programación de sistemas destinado a ser utilizado por contratistas de defensa. En Japón y en otras partes, se gastaron enormes sumas investigando los llamados lenguajes de programación de quinta generación que incorporaban construcciones de la programación lógica. La comunidad de los lenguajes funcionales llevaron a cabo la estandarización de ML y Lisp. La investigación en Miranda, un lenguaje funcional, con evaluación perezosa, comenzó a tomar fuerza en esta década.

Una nueva tendencia importante en el diseño de lenguajes era un mayor enfoque en la programación de sistemas a gran escala a través

del uso de módulos, una organización a gran escala de unidades de código. Modula, Ada, y ML desarrollaron sistemas de módulos notables en la década de 1980. Los sistemas de módulos eran relacionados con frecuencia con construcciones de programación genéricas, que serían, en esencia, módulos parametrizados (véase también polimorfismo en programación orientada a objetos).

Aunque no aparecían nuevos paradigmas de lenguajes de programación imperativos, muchos investigadores expandieron las ideas de los lenguajes anteriores y las adaptaron a nuevos contextos. Por ejemplo, los lenguajes de los sistemas Argus y Emerald adaptaron la programación orientada a objetos para sistemas dis-

tribuidos.

La década de 1980 también trajo avances en la implementación de lenguajes de programación. El movimiento de RISC en la arquitectura de computadoras postulaba que el hardware debía estar diseñado para los compiladores más que para los programadores de ensamblador humanos. Con la ayuda de las mejoras en la velocidad del procesador, permitiendo técnicas de compilación cada vez más agresivas, el movimiento RISC despertó un mayor interés en la tecnología de compilación de los lenguajes de alto nivel.

Las tecnologías de los lenguajes continuaron sobre estas líneas entrando en la década de 1990.