Historia de los Lenguajes de Programación

AÑOS 1940-1959

MANUEL ÁNGEL RUBIO JIMÉNEZ

Historia de los Lenguajes de Programación

Años 1940-1959

Manuel Angel Rubio Jiménez

Historia de los Lenguajes de Programación
Años 1940-1959

Manuel Angel Rubio Jiménez

Resumen

Los lenguajes de programación surgieron para facilitar la forma de interactuar con los ordenadores y crear programas. Hoy en día todos empleamos programas en dispositivos electrónicos diariamente como nuestro móvil, televisión, videoconsola y sobre todo en nuestro ordenador. Los programas están en la mayoría de electrodomésticos y dispositivos electrónicos y la programación se ha convertido en una de las profesiones más demandadas.

Este libro te ofrece una visión de cómo surgieron los primeros lenguajes de programación a partir de los años 40 haciendo un repaso no solo a los lenguajes sino también a quienes los crearon, el contexto alrededor el cual fueron creados, las necesidades que cubrían, las limitaciones y aspiraciones de sus creadores.

Durante la década de 1950 se sucedieron grandes avances y veremos la creación de lenguajes tan populares como FORTRAN o COBOL, las raíces de LISP, otros menos conocidos como BACAIC, IT, Superplan o el más antiguo de todos Plankalkül. Veremos el primer ensablador ARC, el primer interpretado Brief Code, el primer uso del verbo programar, _bug_ y las subrutinas. Los primeros pasos del cálculo lambda, los diagramas de flujo y la Inteligencia Artificial. Grandes avances hechos en tan solo una década.

Ilustraciones: Carlo Gilmar Padilla Santana.

Depósito legal CO-1423-2021.

ISBN 978-84-945523-3-5

[1] http://books.altenwald.com/elixir-i
[2] https://creativecommons.org/licenses/by-nc-sa/3.0/deed.es

Capítulo 1. Tres amigos

Si la gente no piensa que las matemáticas son simples, es sólo porque no se dan cuenta de lo complicada que es la vida.
— John von Neumann

El diseño de las primeras máquinas en Estados Unidos, Reino Unido y Alemania necesitó de la especificación de un lenguaje basado en instrucciones concretas para llevar a cabo cálculos complejos y la implementación de los primeros algoritmos.

Estos primeros algoritmos para estas computadoras se desarrollaron en ***lenguaje máquina***. Este lenguaje es concreto de la arquitectura de la máquina, se escribe en código binario a través de cableado, tarjetas perforadas o una terminal y las máquinas obtienen un resultado.

Antes de comenzar a fabricarse máquinas especializadas en el cálculo, los matemáticos eran contratados como computadores humanos. El trabajo consistía en tomar un papel, normalmente una tarjeta donde se encontraba una fórmula y resolver la fórmula matemática dados algunos números de entrada. Finalmente debían escribir los resultados en otra tarjeta de salida.

Los analistas escribían las fórmulas necesarias para resolver un problema concreto y los computadores se encargaban de procesar las fórmulas y obtener el conjunto de todos los datos necesarios. El analista obtenía en unas horas, días o semanas más tarde todos los números y podía continuar su trabajo.

Dependiendo del sector, estos cálculos podían pertenecer a distintas necesidades, una de las más demandadas en tiempo de guerra eran las tablas balísticas para el lanzamiento de proyectiles. También cálculos específicos para almanaques astronómicos o cálculo de nóminas.

En Princeton a mediados de los años 30 un joven profesor de gran talento fue invitado al Instituto de Estudios Avanzados[1] donde se reunían otras personalidades como Einstein y se le ofreció la membresía como profesor. John von Neumann, un estudiante avanzado llegado desde Hungría con unas excelentes notas y una capacidad de cálculo mental impresionante comenzó así su carrera en los Estados Unidos.

[1] IAS, Institute of Advanced Study.

Figura 1.1. John von Neumann

Me gustaría resaltar algunos hechos sobre von Neumann para entender la inteligencia de este hombre. He encontrado muchas citas acerca de su cociente intelectual (el cual no se conoce) y muchas hazañas acerca de su memoria. Comenzaré por esta cita de George Pólya, profesor de matemáticas en el ETH de Zúrich:

> ***Hubo un seminario para estudiantes avanzados en Zúrich que yo estuve enseñando y von Neumann estuvo en esta clase. Llegué a cierto teorema. Dije que no estaba probado y que podría ser difícil probarlo. Von Neumann no dijo nada pero pasados 5 minutos levantó su mano. Cuando lo llamé a la pizarra procedió a escribir la prueba. Después de esto tuve miedo a von Neumann.***

Eugene Wigner un físico teórico y matemático que recibió el Premio Nobel de Física en 1963 decía de von Neumann:

> ***Cada vez que hablaba con von Neumann, siempre tenía la impresión de que solo él estaba completamente despierto.***

El físico y creador del primer reactor nuclear, Enrico Fermi, le dijo en una ocasión al profesor de Física de la Universidad de Chicago, Herbert Anderson, acerca de las capacidades matemáticas de von Neumann:

> ***Sabes, Herb, ¡Johnny puede hacer cálculos en su cabeza diez veces más rápido que yo! Y yo puedo hacer los cálculos diez veces más rápido que tú. Entonces Herb, ¡Puedes ver lo impresionante que es Johnny!***

Se decía que von Neumann podía leer y memorizar hasta 50 instrucciones de código máquina en su cabeza y mantenía además los primeros 100 números primos hasta su sexta potencia.

Por último, me gustaría compartir la última comparativa hecha por Eugene Wigner acerca del genio de von Neumann:

> ***He conocido a muchas personas inteligentes en mi vida. Conocía a Planck, von Laue y Heisenberg. Paul Dirac era mi cuñado; Leo Szilard y Edward Teller estaban entre mis amigos más cercanos; y Albert Einstein también era un buen amigo. Pero ninguno de ellos era tan agudo como John von Neumann. A menudo he mencionado esto en presencia de estos hombres y nadie me ha corregido nunca.***

En esa misma época, otros jóvenes brillantes se le unirían en Princeton donde podrían trabajar juntos. El primero Alonzo Church, que tras graduarse en 1924 y obtener su doctorado en 1927 continuó como profesor en Princeton a partir de 1929.

1. Cálculo Lambda y Alonzo Church

Church elaboró el cálculo lambda como un sistema formal diseñado para investigar la definición de función, la noción de aplicación de funciones y la recursividad. Este estudio se empleó con el objetivo de definir una función computable.

Figura 1.2. Alonzo Church

En 1935, Church estaba trabajando en la demostración de la teoría de la computabilidad, uno de los problemas más estudiados por los matemáticos en la época propuesto por David Hilbert y Wilhelm Ackermann en 1928 bajo el nombre de Entscheidungsproblem (Problema de Parada), pero se remonta hasta Gottfried Leibniz, quien en el siglo XVII, después de construir exitosamente una máquina mecánica de cálculo, soñaba con construir una máquina que pudiera manipular

símbolos para determinar si una frase en matemáticas es un teorema. El enunciado consistía en:

> ***¿Existe un algoritmo tal que pueda decidir si la proposición es cierta (y por tanto es un teorema del sistema) o por el contrario es falsa?***

Lo primero que sería necesario es un lenguaje formal claro y preciso para su resolución. En este sentido el cálculo lambda parecía encajar a la perfección y Church lo empleó para en 1936 publicar la solución bajo el nombre ***A note on the Entscheidungsproblem*** (Una nota sobre el Problema de Parada) [AC36] dando una respuesta negativa a la pregunta propuesta.

La línea de investigación de Church procedía de las ideas de distintos matemáticos desde Leibniz para resolver la duda y problema planteado por este hasta llegar al enunciado de Hilbert y Ackermann y en base al trabajo de Stephen Kleene, Church pudo continuar y obtener el resultado final. Sin embargo, otro joven llegado también a la Universidad de Princeton en esas fechas y alumno de Church optó por otro enfoque completamente diferente.

2. La Máquina Lógica y Alan Turing

En 1936, Alan Turing corroboró el resultado obtenido por Church a través de otro método diferente. Especificó el problema a través del uso de una máquina lógica [2] para explicar los límites del cálculo mecánico. Emitiría más tarde una corrección titulada ***Sobre Números Computables, con una Aplicación al Problema de Decisión. Una Corrección*** [AT38].

Figura 1.3. Alan Turing

[2] La máquina lógica elaborada por Turing fue nombrada más adelante como Máquina de Turing.

Se intuye que la corrección pudo ser motivada por el contenido del artículo de Church haciendo más asequible la explicación ofrecida por Turing gracias a la Máquina Lógica en lugar del cálculo lambda. No en vano, ambos colaboraron en diferentes trabajos juntos y compartieron muchas ideas a lo largo del tiempo que Turing permaneció en Princeton.

El primer trabajo a resolver para Turing fue definir ¿qué es un algoritmo? La respuesta corta dada por Turing fue ***todo algoritmo es equivalente a una máquina de Turing*** y aunque es una afirmación formalmente indemostrable se ha aceptado de forma universal.

La tesis Church-Turing por lo tanto muestra tres formalismos aceptados y en los que se ha trabajado de forma independiente pero equivalentes entre sí: las máquinas de Turing, los lenguajes formales y el cálculo lambda.

El artículo de Turing asentó las bases para la certificación de máquinas o lenguajes de programación diciendo si son Turing completos o no. La teoría nos dice que ***un lenguaje de programación Turing completo es capaz de implementar cualquier cálculo que cualquier otra máquina (o lenguaje) es capaz de hacer***. No obstante no dice nada del esfuerzo para escribir ese programa o el tiempo que puede tomar el cálculo.

Esta descripción de la Máquina de Turing fue muy importante y asentó las bases de la construcción de distintas máquinas así como de los lenguajes implementados sobre estas máquinas. El código de ejemplo especificado por Turing para su Máquina Lógica Universal se usó de base para lenguajes como ENIAC Short Code (1946) y Glennie Autocode (1952) principalmente. Estos dos lenguajes a su vez influenciarían a muchos otros más como veremos más adelante.

Consta que von Neumann le ofreció a Turing un puesto de asistente en IAS para continuar trabajando con él. Turing rechazó esta propuesta para volver a Inglaterra y alistarse en el ejército. Había comenzado la Segunda Guerra Mundial.

3. Las bases de la computación

Von Neumann obtendría la nacionalidad estadounidense en 1937, se interesó mucho por la matemática aplicada y se convirtió rápidamente en un experto en materia de explosivos que le llevó a diversas consultorías con la Marina de los Estados Unidos.

El trabajo de von Neumann y por el que más se le conoce en la comunidad informática no llegaría aún pero en estos años del IAS comenzó asentando las bases de una colaboración y consultoría que más adelante le abriría las puertas de otros laboratorios, compartir ideas con otros personajes y liberar su potencial creador.

Sin embargo, no podemos descartar el potencial de las mentes y trabajo de Church y Turing. Por un lado, **Church asentó con el cálculo lambda las bases de la programación funcional**.

Por otro lado, **Turing es considerado el padre de la computación**. Como hemos visto, su demostración no solo llevó a una forma de identificar la computabilidad de los algoritmos de forma diferente sino también a definir en qué consiste un algoritmo y la forma interna de una entidad computable a través de la máquina de Turing.

Puedes encontrar el resto de este texto en:

https://books.altenwald.com/

Además de muchos otros libros de Erlang, Elixir, Phoenix Framework y muchas otras tecnologías.