

# 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 interpretador 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



**Historia de los Lenguajes de Programación: Años 1940-1959** por Manuel Ángel Rubio Jiménez<sup>1</sup> se encuentra bajo una Licencia Creative Commons Atribución-NoComercial-CompartirIgual 3.0 No portada (CC BY-NC-SA 3.0)<sup>2</sup>.

---

<sup>1</sup> <http://books.altenwald.com/elixir-i>

<sup>2</sup> <https://creativecommons.org/licenses/by-nc-sa/3.0/deed.es>

---

---

# Capítulo 2. Plankalkül y Konrad Zuse

*Aprendí a jugar ajedrez solo porque lo consideré un juego que podría usar para desarrollar y probar mi máquina de cálculo computacional. [...] El ajedrez me permitió estudiar una mezcla compleja de reglas, diferentes casos y similares, con poco espacio y pocos componentes.*  
— Konrad Zuse

Konrad Zuse, fue un ingeniero alemán que construyó varios ordenadores de relés en los años 40 incluyendo el primer computador electromecánico. Durante la Segunda Guerra Mundial sus computadores fueron destruidos y decidió mudarse a Hinterstein, un pequeño pueblecito de montaña en los Alpes al sur de Alemania y colindante con Austria. Allí siguió con su tesis comenzada en 1943 llamada Plankalkül (Programa de cálculo).



Figura 2.1. Konrad Zuse

El documento de Petrocelli [CP19] sobre Zuse y Plankalkül nos da un marco suficiente para comprender a este genio y su motivación para crear computadores y un lenguaje de programación para resolver problemas matemáticos. Lo veremos en diferentes etapas.

## 1. Creando el Z1

En los años 1930 Zuse comenzó sus estudios en ingeniería civil en la Technische Universität de Berlín. Zuse se encontró así mismo

involucrado en difíciles y repetitivos cálculos que amargaban la vida del estudiante. Zuse tuvo en esos momentos el impulso de construir máquinas calculadoras controladas por un programa.

A diferencia de Aiken (ver el Capítulo 5, *Laboratorio de Computación de Harvard y Howard Aiken*), Zuse no había oído hablar nunca de Charles Babbage y tampoco tenía conocimiento sobre tecnología. Intentó delinear algunas formas, algún marco de trabajo para datos entrantes. Zuse imaginó una superfórmula universal que pudiera procesar cualquier tipo de problema, un tipo de máquina universal muy similar a la propuesta por Turing en su documento de 1936 (ver Capítulo 1, *Tres amigos*).

Haciendo uso de relés de aparatos de telefonía, los cuales pueden adoptar dos posiciones (abierto o cerrado), reconoció no solo los avances técnicos y constructivos de los sistemas de dos valores, sino que también comprendió que todas las operaciones de cálculo algebraico podían resolverse usando esos componentes.

En su documento *Einführung in die allgemeine Dyadik* (Introducción a las didácticas generales) de 1938, el término *Dyadik* fue inspirado por Leibniz. Zuse propuso una lógica matemática que resultó ser sorprendentemente equivalente al cálculo proposicional y el álgebra de Boole, que aprendió más tarde gracias a uno de sus profesores de matemáticas.

Zuse decidió montar en el modesto salón de su casa familiar en Berlín un laboratorio para dedicarse a construir su máquina binaria programable. Su característica más significativa fue la clara distinción entre memoria y procesador. Su solicitud de patente presentada en 1936 muestra que Zuse ya había desarrollado muchos de sus conceptos sobre máquinas digitales varios años antes de la publicación del *Preliminary Discussion of the Logical Design of an Electronic Computing Instrument* [AB47] por von Neumann e incluso había ido más allá del concepto de computadora puramente secuencial, ahora conocida como computadora de von Neumann.

Equipado con una memoria completamente mecánica y una unidad aritmética diseñada para procesar números binarios de coma flotante, el Z1 iba acompañado de una unidad adicional que convertía el formato binario en decimal y viceversa para facilitar la entrada y lectura de datos.

El Z1 recibía las instrucciones mediante una cinta perforada de película cinematográfica de 35mm. Zuse obtuvo dicha cinta de un tío suyo que trabajaba en el principal estudio cinematográfico de Alemania. Los resultados del procesamiento eran enviados a un panel de bombillas eléctricas. En tamaño era comparable a una gran mesa de comedor y fue

descrito por aquellos que lo vieron como: *algo indefinible, compuesto de láminas de metal, placas de vidrio, manivelas, engranajes y discos.*

Más tarde, su trabajo se vio amenazado por el desarrollo de los acontecimientos de la guerra, fue reclutado y enviado al frente oriental. Con la ayuda de algunos colegas, logró montar una nueva máquina derivada de las partes del Z1 y Z2 que no fueron destruidos en la guerra, obteniendo el Z3.

La exitosa demostración del Z3 le garantizó a Zuse un préstamo de 25.000 marcos por parte de la Oficina de Investigación del Espacio Aéreo de Alemania. En 1942 inició la construcción de un modelo posterior: *tuve que cambiar el lugar donde construir la máquina tres veces.* Zuse pudo dotarla de una memoria mecánica mayor y una unidad de control operada por dos cintas perforadas que se podían usar alternativamente.



### Curiosidad

El Zuse Z4 fue adquirido por el ETH<sup>1</sup> de Zúrich en 1950 dos años después de que Eduard Stiefel fundase el Instituto de Matemáticas Aplicadas en el ETH de Zúrich con la idea de dedicarse a la construcción del ERMETH, el primer computador suizo junto con Rutishauser y Speiser (ver el Capítulo 8, *ETH y Superplan*).

## 2. Plankalkül

En 1943, Zuse discutió su tesis doctoral titulada *Ansätze einer Theorie des allgemeinen Rechnens unter besonderer Berücksichtigung des Aussagenkalküls dessen und Anwendung auf Relaisschaltungen* (Aproximaciones a una teoría de la aritmética general con especial consideración de su cálculo proposicional y su aplicación a circuitos de retransmisión). En su documento describía su proyecto para la formalización de un lenguaje de programación para una máquina de propósito general.

En sus tesis, Zuse empleó la palabra *Angaben* (declaraciones) para referirse a los datos a ser computados y *Vorschrift* (órdenes) fue la palabra para indicar el método de computación, las *órdenes* que hoy se conocen como *algoritmo*. También usó *Rechenplan* (plan aritmético) para referirse al conjunto de instrucciones para realizar operaciones para el procesamiento de los datos.

Por tanto, la computación era posible mediante un algoritmo que actúa sobre los datos de entrada para lograr resultados. Era necesario

---

<sup>1</sup>ETH: Eidgenössische Technische Hochschule; o Escuela Politécnica Federal.

escribir reglas para representar estos algoritmos en máquinas que pudieran realizar las operaciones deseadas de una manera totalmente *transparente*. Dicho de otra forma, era necesario *proporcionar una descripción puramente formal para cada procedimiento computacional* que permitiera escribir un *Rechenplan* correcto y estructurado. Por estos motivos, Zuse concibió su Plankalkül como un lenguaje universal, algorítmico y de alto nivel, que se prestaba muy bien a representar y dar solución a problemas de alta complejidad.

Consta que Zuse terminó su tesis en 1945 pero no fue presentada hasta 1972 [KZ72]. Su definición del lenguaje Plankalkül dispone de una cantidad de elementos que hacen pensar en el lenguaje como si se tratara de un lenguaje actual en lugar de un lenguaje diseñado en 1945. La construcción de los equipos Z1 (1938), Z3 (1941) y Z4 (1945) le ayudó a tener claro cómo debía ser un lenguaje de programación para facilitar la creación de programas informáticos.

### 3. El lenguaje adelantado a su tiempo y el ajedrez

En 1948 Zuse realizó una presentación en la reunión anual del GAMM y publicó un artículo en la revista científica *Archiv der Mathematik* [KZ48]. Una revista con publicaciones de alta calidad en matemáticas, aunque no tuvo mucha relevancia ni despertó mucho interés. En la publicación se recogían las ideas de Zuse para un lenguaje algorítmico de propósito general.

*Un muy primer intento de divisar un lenguaje  
algorítmico fue realizado en 1948 por Zuse. Su  
notación era bastante general, pero la propuesta  
nunca alcanzó la merecida consideración.  
— Heinz Rutishauser*

Zuse inicialmente trabajó solo, casi obsesionado con este nuevo aspecto de la computación hasta el punto de aprender a jugar al ajedrez con el objetivo de llegar a intentar formular las reglas en términos de cálculo lógico.

La formalización del juego de ajedrez impuso la necesidad de estructurar pensamientos desordenados con fórmulas estrictas. Hacer que la partida siga las reglas habría requerido inicialmente operaciones a partir de valores binarios que se introdujeran en series de valores binarios y luego en largas listas de pares binarios que surgirían de estos valores anteriores. Esto habría llevado a matrices cada vez más complejas que uno tendría que estructurar de acuerdo con las reglas creadas a partir de datos elementales.

Die Punkte sind benachbart

$$\begin{array}{l} V \\ \mathbb{K} \end{array} \left| \begin{array}{cccccc} V \neq V \wedge |V - V| \leq L \wedge |V - V| \leq L \Rightarrow R \Delta . 17 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ & & 0 & 0 & 1 & 1 \end{array} \right.$$

Koordinaten-Darstellung:

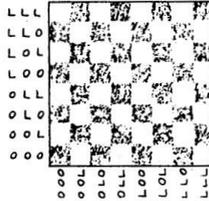


Figura 2.2. Código de Ajedrez en Plankalkül (pequeño extracto)

El ajedrez ofrecía una gran cantidad de estructuras de datos en un espacio limitado. Un lenguaje simbólico<sup>2</sup> que pudiera describir los problemas de ajedrez sería adecuado para todos los problemas a resolver por las computadoras.

El poder expresivo del lenguaje fue probado por el ingeniero alemán con aplicaciones relacionadas con el juego de ajedrez, que, como hemos visto, se prestaban bien al uso de tipos especiales de estructuras de datos, condiciones anidadas y cálculos generales. En su libro dedicado a la descripción del lenguaje destinó un capítulo de 44 páginas a la programación de este juego, satisfecho en su visión de que las máquinas derrotarían a los humanos.

Sin embargo, este lenguaje nunca se implementó realmente. La versión definitiva del lenguaje se terminó en Hinterstein (Allgäu, Bavaria) en 1945, pero las condiciones de posguerra en Alemania impidieron su difusión. En una conferencia en 1957, Zuse comentó:

*Yo, por lo tanto, espero que, últimamente, mi Plankalkül sea despertado del sueño de la Bella Durmiente [...] Me alegraría si esto fuera posible con una colaboración con universidades.*

La publicación completa de su lenguaje de programación no fue posible hasta 1972 [KZ72]. En el documento Zuse escribió:

*Después de algún tiempo, como la bella durmiente, aún está por llegar a la vida.*

<sup>2</sup>La expresión *lenguaje algorítmico* era desconocida para Zuse en ese momento.

## 4. Características del lenguaje

Plankalkül fue diseñado para una nueva máquina lógica. Zuse había descrito las máquinas lógicas básicamente como máquinas mínimas provistas de una memoria con una longitud de palabra de solo un bit y un procesador que solo podía calcular las funciones lógicas elementales Y, O y NO.

Debido a que la memoria se configuró como una cadena de bits, se podrían agrupar en cualquier forma deseada para representar números, caracteres, matrices y cualquier otro tipo de datos.

Plankalkül dispone de distintas estructuras de datos: el bit, el número entero y real, matrices y registros. Las variables se definen como V (para paso de parámetros), Z (para los valores intermedios) y R (para los resultados).

El principal problema de Plankalkül fue su difícil sintaxis. El programador tomaría mucho tiempo para poder descifrar las secuencias lógicas escritas y utilizadas en el programa. No obstante Zuse escribió varios programas: un programa de clasificación, otro para probar la conectividad de grafos, otros de aritmética que incluían el cálculo de la raíz cuadrada e incluso un juego de ajedrez.

El código de este lenguaje tiende a escribirse de forma matemática a través de expresiones y matrices no solo ocupando una línea, sino en varias líneas en caso de ser necesario como puede verse en la Figura 2.3, "Código en Plankalkül".

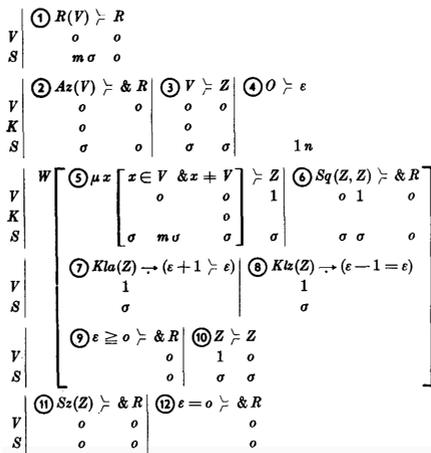


Figura 2.3. Código en Plankalkül

Podemos ver un ejemplo traducido a texto plano de código para calcular el máximo de tres números a continuación:

```
P1 max3 (V0[:8.0],V1[:8.0],V2[:8.0]) -> R0[:8.0] ❶
max(V0[:8.0],V1[:8.0]) -> Z1[:8.0]
max(Z1[:8.0],V2[:8.0]) -> R0[:8.0]
END
P2 max (V0[:8.0],V1[:8.0]) -> R0[:8.0] ❷
V0[:8.0] -> Z1[:8.0] ❸
(Z1[:8.0] < V1[:8.0]) -> V1[:8.0] -> Z1[:8.0]
Z1[:8.0] -> R0[:8.0] ❹
END
```

- ❶ P1 es el primer procedimiento definido, con tres parámetros de entrada y un parámetro de salida. Se encarga de obtener el número mayor de esos tres elementos.
- ❷ El código se compone de dos llamadas a otra función que calcula el máximo de dos elementos.
- ❸ Aquí podemos ver el empleo de las variables V para los parámetros.
- ❹ Las variables Z dentro de las funciones para contener los valores intermedios y R para el retorno de los resultados.

Por ejemplo una operación matemática de suma de dos datos almacenados previamente en dos variables podría especificarse así:

```
V1 + V2 -> R1
```

De esta forma el valor contenido en V1 es sumado al valor contenido en V2 y el resultado se almacena en R1.

No obstante todo esto fue completamente teórico puesto que Zuse no pudo completar una implementación funcional de su lenguaje.

## 5. Conclusiones

Petrocelli concluye su documento diciendo: *Plankalkül tendría gran importancia también para resolver problemas actuales: programación lógica y bases de datos relacionales, pero también el desarrollo de formas estandarizadas en inteligencia artificial se habría beneficiado enormemente y probablemente se habría desarrollado primero. Plankalkül ya muestra muchos de los principios básicos de los lenguajes de programación de los años 60, aunque en ocasiones ocultos bajo una notación viciada. Algunas características, como la estructuración de*

*objetos, solo recientemente han encontrado un lugar en los lenguajes de programación. Quizás Zuse hubiera merecido un reconocimiento diferente en la historia de la informática.*

Podemos decir sin duda alguna que Plankalkül se adelantó a su tiempo. No obstante, sus dificultades se observan más allá de su época. Su sintaxis era compleja y esto llevó a no disponer de un compilador teórico hasta la disertación de Joachim Hohmann en 1975 con título *El Plankalkül comparado con lenguajes algorítmicos* [JH75].

Zuse falleció el 18 de diciembre de 1995. No pudo ver nunca una implementación funcional de su lenguaje ni ninguna de las aplicaciones desarrolladas de forma teórica para él.

El primer compilador funcional de Plankalkül fue desarrollado en 1998. La primera aplicación se desarrolló en la Universidad Libre de Berlín en el año 2000 [RR00]. Aún hoy en día sigue siendo un lenguaje no empleado para el desarrollo de software pero sí un lenguaje con un contexto histórico importante.

Debemos reconocer el mérito de Zuse en la especificación del lenguaje pero también entender su baja utilidad actual. Las necesidades de la época llevaron el desarrollo de los lenguajes de programación por otros caminos diferentes y hoy en día sería difícil adoptar Plankalkül como lenguaje para desarrollo de soluciones.

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.