

1.

Introducción

“Hice mi más humilde reconocimiento a esta ilustre persona, por ser tan comunicativo; y le prometí que ‘si alguna vez tuviera la fortuna de regresar a mi país, le haría justicia, como único inventor de esta máquina maravillosa’ [...]. Le dije, ‘aunque sea costumbre de nuestros eruditos en Europa robarse los inventos unos a otros, para así tener al menos la ventaja de que se genere la controversia sobre quién es el verdadero autor, tomaré tales precauciones, que tendrá todo el honor, sin rival alguno’”.

Jonathan Swift
“Gulliver's Travels”

El estudio de un computador se divide en dos partes: la física —procesadores, memorias etc. — y la lógica —programas que lo componen, convenciones, etc. —. La parte física es el hardware y la parte lógica es el software. Hablando crudamente, el hardware constituye la máquina y el software los programas.

El hardware de un computador, a su vez, se divide en tres grandes partes: el procesador, la memoria y los periféricos. Las tres interactúan entre sí, las tres son necesarias para tener un computador funcional (ver fig. 1.1)

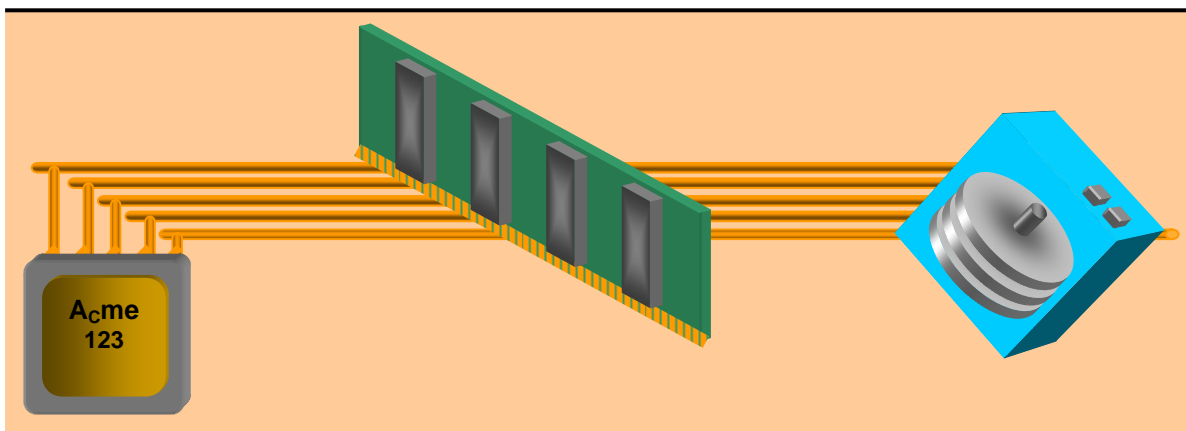


Fig. 1.1. Partes de un computador

El procesador es la parte encargada de ejecutar las instrucciones, es él quien dirige a los otros componentes. La memoria, por su lado, es la encargada de almacenar datos e instrucciones. Sin embargo, su papel es pasivo, puesto que no sabe qué es dato y qué es instrucción; se limita a entregarle al procesador lo que este le pide y a almacenar lo que le envía. Podemos imaginar la memoria como una lista numerada de objetos; el procesador le dice a la memoria "envíe el n -ésimo ítem", la memoria busca la posición n en la lista y envía lo que allí se encuentre.

El procesador y la memoria constituyen la parte operativa del computador; son los encargados de ejecutar los programas. La labor de los periféricos es otra: son el medio por el cual el procesador se comunica con el mundo exterior. Algunos ejemplos de periféricos son el teclado, la pantalla y la impresora.

Algunos periféricos sirven para almacenar datos, como es el caso para los discos duros, las cintas y los CD. Se les llama *memoria secundaria*, puesto que almacenan datos como la memoria principal pero, a diferencia de esta, los datos no son directamente asequibles al procesador; deben ser trasladados a la memoria principal antes de que el procesador pueda usarlos.

La anterior es una descripción general de los componentes de un computador, pero no es suficiente para entender su funcionamiento. Para esto es necesario conocer la estructura y la arquitectura del procesador, incluyendo en esta última el lenguaje utilizado para describir las operaciones (*lenguaje de la máquina*). La estructura es una descripción de los componentes del procesador, de sus interacciones y de su funcionamiento. En cuanto a la arquitectura, es la interfaz que ofrece la máquina para su programación: los tipos de operandos que maneja, las operaciones que puede efectuar, los formatos de instrucciones, etc.

Hasta ahora hemos tratado la parte física, pero, como mencionamos anteriormente, hay una parte lógica compuesta de programas y convenciones. Algunas de estas convenciones surgen del hecho de que los circuitos del procesador son binarios; es necesario establecerlas para poder representar entidades tales como: números, caracteres e incluso las mismas instrucciones de la máquina. Por esto empezamos por estudiar el sistema binario y los métodos de representación de la información.

De nada sirve un computador sin programas que lo hagan funcionar. Estos se pueden dividir en dos grupos: los del usuario y los del sistema. Los programas del sistema se encargan de operar la máquina continuamente. En efecto, cuando un computador se activa, algún programa debe empezar a ejecutar, al menos un programa que reciba y ejecute órdenes del operador. Además, en los sistemas que admiten varios usuarios al mismo tiempo, es necesario tener algún programa que administre los recursos del computador entre todos los programas que están ejecutando. Este "administrador" de la máquina, así como los programas utilizados para desarrollar otros programas, son los llamados *programas del sistema*.

Todos los elementos antes descritos constituyen la base de un computador, pero, antes de proceder a su estudio, daremos una mirada a algunos fundamentos de la computación, lo cual también nos proveerá de un marco histórico.

1.1 FUNDAMENTOS DE LA COMPUTACIÓN

Diferentes personas y procesos han contribuido en la creación del computador como lo entendemos hoy en día. En las secciones siguientes, analizaremos algunos de esos aportes y su relevancia en el proceso. No se trata de una historia de la computación como tal, sino de identificar las ideas fundamentales que dan origen al desarrollo del computador tal como lo conocemos hoy en día.

La máquina de Pascal

En 1.642, Pascal construyó una sumadora mecánica, la “pascalina”; solo podía sumar, la resta se hacía por técnicas de complemento a 9.

No fue la primera y tampoco funcionó muy bien. Sin embargo, es importante porque efectuaba operaciones que, en ese entonces, se consideraban propias de los seres humanos. Pascal hizo caer en cuenta de que no es así; que esas actividades son, en realidad, mecánicas; que si se dispone de un algoritmo para algo, es posible, en principio, pensar en construir una máquina que lo implemente de manera mecánica.

Hasta antes de las calculadoras mecánicas, todos los dispositivos de cálculo (ábacos, etc.) eran esencialmente algorítmicos, pero el operador se encargaba de ejecutar el algoritmo. El algoritmo está en el operador, no en el aparato; este solo sirve de soporte.

En cambio, las calculadoras mecánicas son aparatos que pueden realizar las operaciones por ellos mismos; es decir, el algoritmo está en el aparato, no en el operador. Así el cálculo se vuelve, literalmente, mecánico.

Se debe enfatizar que estas máquinas no son computadores; les falta un concepto fundamental: el programa.

La máquina de Leibniz

Leibniz construyó una calculadora mecánica: la calculadora por pasos. Quería una máquina capaz de hacer las cuatro operaciones, por eso, aunque conocía el trabajo de Pascal, desarrolló un nuevo mecanismo, el cual implementó en 1.674.

Sin embargo, Leibniz tiene un aporte de mayor importancia: estudió y describió formalmente el sistema binario en un ensayo publicado en 1.701. Aunque no sería justo decir que lo descubrió —antes de él ya había sido usado de una manera u otra—, sí vislumbró su potencial para el cálculo. En particular, pensó que se podrían construir máquinas basadas en el sistema binario.

Leibniz también deseaba crear un sistema para expresar las verdades científicas; este sistema estaría compuesto por dos partes: un lenguaje que permitiera expresarse de una manera precisa (*Characteristica universalis*) y un cálculo del pensamiento para expresar las “leyes del pensamiento” (*Calculus ratiocinator*), lo que hoy llamaríamos lógica simbólica. Era una aproximación formal, es decir, basada en la forma de los argumentos: los símbolos no se operan basados en lo que quieren decir, sino en la forma que tienen; esto abre la posibilidad a una manipulación mecánica de dichos símbolos.

¿Qué es ...

... una máquina de Turing? Una máquina abstracta que dispone de una cinta de entrada dividida en cuadrados donde puede escribir, o leer, un conjunto finito de símbolos. La máquina siempre se encuentra en un cierto estado —“configuración”, lo llamaba Turing— elegido entre un conjunto finito de estados.

En todo momento, la máquina puede examinar un solo símbolo de la cinta, y, basada en el símbolo y en el estado en el que se encuentra, puede escribir un símbolo, mover la cinta un lugar a la derecha o a la izquierda y cambiar de estado.

Además, Leibniz ligó esta idea con el sistema binario: el uno y cero eran el sí y el no, el cierto y el falso. Interesante visión, porque se usa en el diseño de los computadores modernos: puesto que son binarios, el uno y el cero se interpretan como “cierto” y “falso”, y se diseñan los circuitos usando lógica simbólica.

La máquina de Turing

En 1.900, durante el Congreso Internacional de Matemáticos, el matemático alemán David Hilbert presentó una propuesta de 23 problemas que deberían ser estudiados por los matemáticos en el siglo que empezaba. El segundo de estos problemas trataba de la compatibilidad de los axiomas de la aritmética. En él, Hilbert proponía que se demostrara que los axiomas de la aritmética no pueden conducir a resultados contradictorios.

Posteriormente, en 1.920, amplió este planteamiento en lo que se ha dado en llamar el Programa de Hilbert. El programa consistía en una formalización axiomática de las matemáticas¹, y en probar que eran completas —toda afirmación puede ser probada o refutada—, consistentes —no se puede demostrar algo y su negación— y decidibles —hay un método algorítmico, mecánico, para saber si una afirmación puede o no ser demostrada—.

Kurt Gödel se ocupó de probar que no se podían hacer las dos primeras demostraciones. Por su lado, el matemático inglés Alan Turing se interesó por la tercera demostración. En 1.936, publicó el artículo “*On computable numbers, with an application to the Entscheidungsproblem*”, donde demostraba que no se podía resolver el problema de la decidibilidad.

Para hacer esto, Turing necesitó formalizar el concepto de realizar mecánicamente una tarea por medio de una máquina abstracta.

En sus propios términos

“Es mi opinión que estas operaciones incluyen todas aquellas que se usan en el cómputo de un número. La sustentación de esta opinión será más fácil cuando el lector esté familiarizado con la teoría de las máquinas. En la próxima sección, en consecuencia, procederé al desarrollo de la teoría y supondré que se entiende lo que se quiere decir con ‘máquina’, ‘cinta’, ‘examinado’, etc.”.

A. M. Turing

“*On computable numbers, with an application to the Entscheidungsproblem*”

¹ En realidad, se trataba de axiomatizar la lógica, y de demostrar que la teoría de números y la teoría de conjuntos se podían reducir a la lógica.

La máquina no emprende las acciones de manera arbitraria, sino que dispone de un conjunto de reglas que especifican cómo debe comportarse. Aquí debe entenderse un punto importante: el conjunto de reglas hace parte de la máquina misma; no se puede cambiar. En ese sentido, no es como un programa, sino que modela el comportamiento mecánico que tendría un tal aparato. Así, una máquina de Turing específica corresponde a un solo algoritmo².

Como puede verse, es una máquina bastante sencilla, y, sin embargo, ya captura toda la esencia de la computación. Turing la utilizó para demostrar diversos resultados —entre ellos, la imposibilidad de resolver el problema de la decidibilidad—, pero hay uno que nos interesa particularmente:

Se pueden definir máquinas de Turing para resolver diversos problemas algorítmicos, pero, se preguntó Turing, ¿se puede definir una máquina de Turing que sea capaz de emular cualquier otra máquina de Turing? O, dicho en otros términos, ¿el problema de ejecutar, seguir, un algoritmo es algorítmico? Su respuesta es afirmativa: se puede definir una máquina de Turing que reciba en su cinta de entrada la descripción —codificada— de cualquier otra máquina de Turing, junto con los datos de entrada de la misma, y que calcule el mismo resultado que calcularía la máquina codificada. Esta es la máquina universal de Turing —La Máquina de Turing, por antonomasia—.

El resultado es importante porque indica que se puede construir una sola máquina para resolver todos los problemas algorítmicos. Dicha máquina, en lugar de procesar los datos directamente, procesa descripciones de los algoritmos (los programas). Esto, en última instancia, es un computador. Nótese que, en consecuencia, los datos de entrada que procesa un computador, son, en realidad, los programas.

La Tesis de Church-Turing

El matemático estadounidense Alonzo Church también se ocupó del problema propuesto por Hilbert. Igual que Turing, necesitó formalizar el concepto de algoritmo, pero definió una propuesta diferente: el cálculo lambda.

Estado actual	Símbolo actual	Escribir	Moverse a la	Nuevo estado
Par	0	—	Derecha	Par
Par	1	1	Derecha	Impar
Par	—	P	Derecha	Fin
Impar	0	—	Derecha	Impar
Impar	1	1	Derecha	Par
Impar	—	I	Derecha	Fin

Fig. 1.2. Ejemplo de una máquina de Turing. Recorre un número binario de izquierda a derecha. Borra los ceros (el símbolo “—” denota el blanco), pero deja los unos. Al final escribe si el número de unos es par (“P”) o impar (“I”). Empieza en estado “Par”.

² Siendo consecuentes, sería mejor no hablar de “la” máquina de Turing, sino de “las” máquinas de Turing.

¿Qué es ...

... el cálculo lambda? Un sistema que permite definir funciones formalmente, y tiene el poder de una máquina de Turing. Está compuesto de muy pocas reglas:

- Las funciones se definen así:

$(\lambda x. exp)$ donde x es el parámetro de la función y exp es una expresión que puede involucrar la variable x .

- La aplicación de una función en un argumento se expresa así:

$(exp_1 exp_2)$ donde exp_1 es una λ -función y exp_2 es el argumento.

- La evaluación de funciones se define así:

$((\lambda x. exp_1) exp_2) \equiv exp_1 [x/exp_2]$ donde esta última notación quiere decir: reemplace todas las ocurrencias de x en exp_1 por exp_2 .

Sencillo como se ve, permite definir los números y la aritmética sin ningún otro tipo de primitiva.

El lenguaje de programación LISP está inspirado en el cálculo lambda.

Tanto él como Turing encontraron que los dos sistemas eran igualmente poderosos: podían resolver la misma clase de problemas.

Ha habido otras formalizaciones del concepto de algoritmo —funciones recursivas, sistemas de Post, etc.—, pero se ha encontrado que todas ellas son equivalentes: lo que se puede expresar en uno de los sistemas, se puede expresar en los otros. Esto nos conduce a la Tesis de Church-Turing: las máquinas de Turing —o las funciones recursivas de Church— marcan el límite de las funciones que son efectivamente calculables —o, si se quiere, que pueden ser calculadas mecánicamente—.

Otra forma de verlo es que el hecho de que un problema se pueda resolver mecánicamente está ligado al problema y no al sistema o método que se use para expresar el algoritmo. Es decir, hay dos tipos de problemas: los que se pueden resolver mecánicamente y los que no.

Una tercera forma de verlo: no puede haber un sistema para calcular mecánicamente que sea más poderoso que las máquinas de Turing.

Por supuesto, se trata de una tesis porque no puede ser demostrada, y no puede ser demostrada porque el concepto de “algorítmico” o “calculable mecánicamente” es informal. De cierta manera, podría llegar a ser refutada, si se encuentra un contraejemplo; sin embargo, incluso esto se prestaría para grandes discusiones.

El resultado es importante porque nos dice que basta con construir una máquina que tenga el poder de la Máquina Universal de Turing para resolver cualquier problema algorítmico. Es decir, existe un mínimo hardware con el cual se puede resolver cualquier problema algorítmico; ahora bien, en la práctica, los

computadores no se limitan a ese mínimo hardware, puesto que entran en consideración cuestiones de eficiencia³.

Por otro lado, tanto las máquinas de Turing como el cálculo lambda nos hacen ver que basta con sistemas muy sencillos para tener el poder de lo expresable algorítmicamente.

Las máquinas de Babbage, Zuse y Aiken

Ha habido bastante discusión sobre quién inventó el computador. Por supuesto, parte de la discusión es asunto de definición: ¿“inventó” quiere decir “ideó” o quiere decir “construyó”? ¿qué requerimientos se imponen para considerar a una máquina como un computador? Por otro lado, está el aspecto de qué tanto influyó el trabajo en cuestión en desarrollos posteriores.

Aquí no mediremos en la discusión sino que miraremos diversos trabajos de pioneros en el campo de la computación.

En el siglo XIX, el inglés Charles Babbage trabajó en el desarrollo de dos máquinas: la máquina diferencial y la máquina analítica. La primera era una máquina de propósito específico —calculaba polinomios por el método de las diferencias—; la segunda era un computador de propósito general: estaba basada en tecnología mecánica, tenía una memoria para almacenar los resultados de los cálculos, era decimal y se podía programar —aunque el programa no se almacenaba en memoria—. Aunque no concluyó ninguna de las dos, es claro que tenía la idea esencial de un computador.

Sin tener conocimiento de los trabajos de Babbage, desde 1.934, el alemán Konrad Zuse empezó a trabajar la idea de desarrollar ingenios computacionales. Para 1.938 Zuse había terminado de construir su primera máquina: el Z1. Se trataba de una máquina basada en tecnología mecánica, binaria —pero con entrada-salida en decimal—, números de punto flotante y programable, aunque el programa no se almacenaba en memoria⁴.

La máquina no era muy confiable y Zuse decidió rediseñarla usando relés. La segunda máquina de Zuse, el Z2, era más bien un prototipo para ensayar la idea. El control y la unidad aritmética estaban basados en relés, pero la memoria seguía siendo mecánica.

En sus propios términos

“La Máquina Analítica es, en consecuencia, una máquina de naturaleza lo más general. Cualquiera sea la fórmula que se necesite desarrollar, la ley de su desarrollo se le debe comunicar por medio de dos conjuntos de tarjetas. Una vez puestas, la máquina es especial para esa fórmula en particular”.

Charles Babbage

“Passages from the Life of a Philosopher”

³ De hecho, la frontera entre el hardware y el software es móvil, y es un punto de discusión entre diseñadores de computadores.

⁴ Zuse contempló la idea, pero no la implementó debido a que eso implicaba construir una gran memoria que pudiera contener los programas.

Para 1.941, Zuse había construido su tercera máquina, el Z3, enteramente basada en relés. Sus características eran similares a las anteriores: binaria con entrada-salida en decimal, números de punto flotante⁵ y programable, pero el programa no se almacenaba en memoria.

Desafortunadamente, estos trabajos de Zuse se desarrollaron durante la Segunda Guerra Mundial, por lo cual no tuvieron mucha trascendencia fuera de Alemania; además, máquinas, planos y fotos fueron destruidos durante los bombardeos de Berlín. Sin embargo, en 1.941, Zuse fundó una compañía (*Zuse Apparatebau*), y, en 1.942, empezó el diseño de una nueva máquina (Z4).

Después de la guerra, Zuse fundó otra compañía (*Zuse KG*) en 1.949, y se dedicó a comercializar computadores. Diseñó diversos modelos hasta la desaparición de la compañía en 1.969.

En los Estados Unidos, Howard Aiken también se interesó por el diseño de máquinas automáticas para calcular. En 1.937, Aiken entró en contacto con IBM con el propósito de construir una máquina: la *Automatic Sequence Controlled Calculator* (ASCC), que posteriormente se llamaría Mark I. La máquina fue concluida a finales de 1.943, y presentada a la Universidad de Harvard en 1.944.

El Mark I estaba basado en relés, era decimal y programable con programas no almacenados en memoria. Posteriormente, Aiken trabajó en el desarrollo de otros computadores electrónicos (Mark II a Mark IV).

Es de anotar que Aiken conocía los trabajos de Babbage, pero no supo de Zuse. Su trabajo tuvo amplia difusión, por lo cual durante un tiempo se creyó que el Mark I era el primer computador programable que se había construido.

De las máquinas antes mencionadas, se puede destacar que sus constructores cayeron en cuenta de la importancia de los conmutadores en la computación, fueran estos mecánicos o electromecánicos —relés—; efectivamente, sabemos que un computador puede ser enteramente construido usando solo conmutadores. Igualmente, todos tenían la noción de una máquina programable.

En el caso de Zuse, también es destacable que cayó en cuenta de usar el sistema binario para facilitar los cálculos y simplificar la máquina; también cayó en cuenta de usar la lógica como herramienta de diseño para los circuitos binarios.

***Colossus* y las máquinas de Atanasoff y Eckert y Mauchly**

Entre 1.937 y 1.942, John Atanasoff y Clifford Berry construyeron un computador electrónico de propósito específico —no era programable—; podía resolver sistemas de hasta 29 ecuaciones lineales simultáneas con 29 incógnitas por medio de la eliminación Gaussiana. El *Atanasoff-Berry Computer*, o ABC, funcionaba con tubos al vacío, a diferencia de las máquinas antes mencionadas, que eran total o parcialmente mecánicas. El ABC usaba el sistema binario y su diseño estaba basado en la lógica.

⁵ Notablemente, Zuse incorporó el manejo de excepciones aritméticas: indefinido e infinito.

Nótese que, aunque era electrónico, su diseño seguía estando basado en conmutadores: tubos al vacío⁶. Solo que la tecnología electrónica permite una velocidad de conmutación más alta, y, por ende, máquinas más veloces.

Durante la Segunda Guerra Mundial, los británicos diseñaron una máquina electrónica (*Colossus*) con el propósito de decodificar mensajes alemanes; a principios de 1.944 la máquina era operacional. Funcionaba con tubos y era programable aunque en forma limitada. Su diseño se mantuvo en secreto, así que no influyó en desarrollos posteriores.

Por otro lado, entre 1.943 y 1.945, financiados por el ejército estadounidense, Presper Eckert y John Mauchly construyeron la máquina conocida como ENIAC (*Electronic Numerical Integrator and Computer*). Se trataba de una máquina decimal y era programable, aunque, para esto, necesitaba que se cablearan sus unidades; es decir, no se programa por medio de una cinta o tarjetas perforadas —como es el caso de las máquinas programables antes vistas— sino que se establecían conexiones físicamente.

La arquitectura de von Neumann

El destacado matemático John von Neumann (húngaro de nacimiento) supo del trabajo que realizaban Eckert y Mauchly. Junto con ellos hizo parte del equipo que trabajó en el desarrollo del sucesor de ENIAC: EDVAC (*Electronic Discrete Variable Automatic Calculator*). En un reporte titulado “*First Draft of a Report on the EDVAC*”, presentó las características de la máquina, las cuales incluían los elementos típicos de los computadores modernos: unidad de control, unidad aritmética-lógica, memoria y dispositivos de entrada-salida. Además de lo anterior, se proponía una máquina binaria y con programa almacenado en memoria.

Este último se considera un concepto fundamental para los computadores modernos; permite varias cosas, entre ellas la manipulación de programas por medio de otros programas.

Se ha dado en llamar “Arquitectura de von

En sus propios términos

“En la medida en que el dispositivo terminado será una máquina calculadora de propósito general, debe contener ciertos órganos principales relacionados con la aritmética, el almacenamiento en memoria, el control y la conexión con el operador humano [...].

“Es evidente que la máquina debe ser capaz de almacenar, de alguna manera, no solo la información digital necesaria en un cierto cálculo, [...] sino también las instrucciones que dirigen la rutina precisa que se realizará con los datos numéricos. En una máquina de propósito específico, dichas instrucciones son parte integral del dispositivo [...]. Para una máquina de propósito general, debe ser posible darle instrucciones al dispositivo para llevar a cabo cualquier cálculo que se pueda formular en términos numéricos. En consecuencia, debe haber algún órgano capaz de almacenar estas órdenes del programa. Debe haber, además, una unidad que pueda entender estas instrucciones y ordenar su ejecución”.

Arthur Burks, Herman Goldstine,
John von Neumann

“*Preliminary Discussion of the Logical
Design of an Electronic Computing
Instrument*”

⁶ Los tubos al vacío se pueden usar para diversos propósitos; pero, en el caso particular de la computación, se usaron como conmutadores electrónicos.

¿Qué es ...

... el álgebra de Boole? Un álgebra con dos operadores binarios: $+$ (o-lógico), \bullet (y-lógico), un operador unario: $\bar{}$ (negación) y dos valores: 0 y 1 (cierto y falso), que cumplen con los siguientes axiomas:

- | | | |
|-----------------------|---|--------------------------------------|
| ▪ Conmutatividad: | $a+b = b+a$ | $a\bullet b = b\bullet a$ |
| ▪ Distributividad: | $a\bullet(b+c) = (a\bullet b)+(a\bullet c)$ | $a+(b\bullet c) = (a+b)\bullet(a+c)$ |
| ▪ Elemento identidad: | $a+0 = a$ | $a\bullet 1 = a$ |
| ▪ Elemento inverso: | $a+\bar{a} = 1$ | $a\bullet\bar{a} = 0$ |

Neumann” a la arquitectura descrita en ese reporte, y es la arquitectura que siguen los computadores actuales.

Curiosamente, EDVAC no fue el primer computador de ese tipo. En efecto, el proyecto se demoró bastante en terminar (1.952) y, entretanto, en Inglaterra se desarrollaron otras máquinas; en particular, EDSAC (*Electronic Delay Storage Automatic Computer*), diseñada por Maurice Wilkes y terminada en 1.949.

La teoría de Shannon

Hace ya más de dos milenios, en el año 384 a.C., nació el filósofo griego Aristóteles. Destacable por muchas cosas, en particular, por su concepción de la lógica, expresada en un conjunto de tratados conocido como “*Organon*” (“el instrumento”).

En Aristóteles encontramos el origen de la lógica formal, es decir, basada en la forma de los argumentos: se opera sobre los símbolos basándose en la forma que tienen y no en su significado. La importancia del concepto radica en que abre la posibilidad de una manipulación mecánica de dichos símbolos, puesto que la forma es una característica objetiva que puede ser identificada y transformada por una máquina.

La idea del formalismo fue retomada, tiempo después, por otras personas; entre ellas, el filósofo alemán Gottfried Wilhelm Leibniz y el matemático inglés George Boole. Este último buscaba expresar la lógica de manera simbólica, para lo cual creó un álgebra conocida hoy día como Álgebra de Boole. La hizo pública en un opúsculo titulado: “*The Mathematical Analysis of Logic*”, posteriormente la amplió en un tratado titulado “*An Investigation of the Laws of Thought, on which Are Founded the Mathematical Theories of Logic and Probabilities*”.

El álgebra desarrollada por Boole tenía dos valores —0 y 1; falso y cierto—, y un conjunto de axiomas que expresaban sus propiedades, y permitían su manipulación simbólica.

Claude Shannon, ingeniero y matemático estadounidense, tiene en su haber varios méritos: por un lado, creó la teoría de la información y la aplicó a las comunicaciones en un artículo titulado “*A Mathematical Theory of*

*Communication*⁷. Por otro lado, en su tesis de maestría (*A Symbolic Analysis of Relay and Switching Circuits*), propuso el uso del álgebra de Boole para el análisis de circuitos de conmutadores. En este trabajo, él identificó los valores de verdad (“cierto” y “falso”) con el 0 y 1 binarios, y con el “prendido” y “apagado” de un conmutador; basado en esto mostró cómo se podían construir circuitos para calcular, basados en conmutadores, por medio de la lógica.

Shannon no construyó un computador, pero la teoría que desarrolló es fundamental en el proceso de diseño de los computadores, puesto que permite sistematizar el análisis, diseño y síntesis de los circuitos en los cuales se basan.

Los conmutadores

Los computadores, por lo menos hasta el momento, están basados en conmutadores; un computador es, en el fondo, un inmenso conjunto de conmutadores interconectados. Siendo simplistas, la historia de los computadores se puede reducir a la búsqueda del conmutador ideal: desde los mecánicos, pasando por los electromecánicos, hasta los electrónicos. Es sorprendente que una de las máquinas más complejas, como lo es el computador, esté completamente basada en el humilde conmutador. El conmutador es una herramienta muy sencilla, pero ya encierra el poder de lo computable.

Un conmutador es un dispositivo en el cual una señal de control permite o impide la circulación de otra. Un ejemplo son los conmutadores de luz: al oprimirlos con el dedo —señal de control— permitimos, o impedimos, el paso de la corriente que enciende al bombillo.

Una primera forma de conmutador es el relé, el cual fue inventado en el siglo XIX por el científico estadounidense Joseph Henry. Su funcionamiento es sencillo: la señal de control activa un electroimán, el cual atrae una placa que, a su vez, cierra un circuito que permite el paso de la señal controlada. Es solo un conmutador que, en lugar de activarse por la aplicación de una fuerza mecánica, lo hace por una atracción magnética.

El relé tenía algunos inconvenientes: relativamente lento, con posibilidades de atascarse, aparatoso y con bastante consumo de energía. Por lo cual, al cabo del tiempo, dio paso al tubo al vacío; el cual presenta la ventaja de ser completamente electrónico.

Estos tubos, inventados por el estadounidense Lee de Forest⁸, permiten que una corriente regule el paso de otra, pero sin la intervención de dispositivos mecánicos. Los tubos tienen un cátodo —filamento metálico que se calienta para emitir electrones—, un ánodo —electrodo que recibe los electrones— y, en medio de ellos, una rejilla —tercer electrodo; conectado a la señal de control— que permite modular la corriente que fluye del cátodo al ánodo.

⁷ Como dato curioso, en ese trabajo él acuñó la palabra “bit”.

⁸ Para ser precisos, se trataba del tríodo; hay otras clases de tubos.

¿Qué es ...

... un semiconductor? Un material que puede variar sus características conductivas. Por ejemplo, el silicio se puede volver conductor bombardeándolo con radiación electromagnética o dopándolo.

¿Por qué es semiconductor? El silicio tiene cuatro electrones de valencia. En un cristal de silicio, cada átomo forma cuatro enlaces covalentes con cuatro vecinos. En estas condiciones, los electrones no pueden moverse y, por ende, el silicio no puede conducir corriente eléctrica.

Si se bombardea con radiación electromagnética, los electrones pueden tomar energía suficiente para subir a otro orbital donde quedan libres y pueden conducir corriente.

También, si se le agrega un material de valencia cinco, cuatro de los electrones formarán enlaces covalentes con los vecinos, pero uno quedará libre para conducción eléctrica. Similarmente, se puede lograr conducción agregando un material de valencia tres.

Los tubos hacían un mejor trabajo que los relés, pero consumían bastante energía, se calentaban y no eran lo suficientemente confiables para su uso en circuitos con gran cantidad de conmutadores. Por este motivo, en los laboratorios Bell, se tomó la decisión de crear un mejor dispositivo; decidieron explorar los semiconductores. Este trabajo condujo al desarrollo del primer transistor, en 1.947, por John Bardeen, Walter Brattain y William Shockley, descubrimiento por el cual recibieron el premio Nobel en 1.956.

Los transistores se elaboran con materiales semiconductores, como el silicio. El silicio se puede volver conductor dopándolo con impurezas de materiales con cinco electrones de valencia —impurezas tipo n, como el fósforo—, o con materiales con tres electrones de valencia —impurezas tipo p, como el boro—.

Si se une silicio tipo n con tipo silicio tipo p, en la unión (juntura pn), se producen efectos especiales que permiten la creación de diodos; estos son dispositivos asimétricos que permiten el flujo de corriente en una dirección pero no en la otra. Si se ponen tres capas (pnp o npn), se crea un transistor; su funcionamiento es más

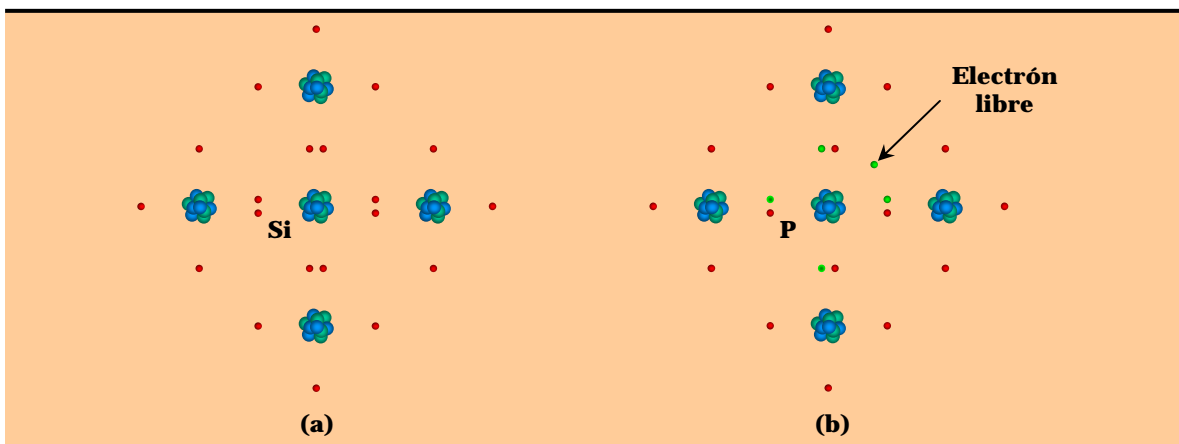


Fig. 1.3. (a) Cristal de silicio. (b) Cristal de silicio con un átomo de fósforo

curioso, puesto que permite o impide el flujo de corriente dependiendo de la señal de control —que se conecta a la capa de la mitad—⁹.

Un paso importante en la evolución de los computadores es el uso de transistores para implementar los conmutadores. En efecto, debido a su pequeño tamaño y bajo consumo de energía, permiten el desarrollo de computadores con una mayor cantidad de conmutadores. Esto redundaba en máquinas más poderosas y con mayor capacidad, sin aumentar por ello el costo ni el tamaño.

Los circuitos integrados

Aunque el transistor era una gran mejora con respecto al tubo, en la medida en que los circuitos se volvían más complejos, también se volvía inmanejable la gran cantidad de transistores que intervenían. Los circuitos integrados, inventados independientemente y de manera casi simultánea por Jack Kilby y Robert Noyce, aparecieron como una solución a este problema.

El circuito integrado está basado en el uso de transistores, solo que, en lugar de fabricarlos de manera independiente e individual, se crean simultáneamente todos los transistores de un circuito sobre una pequeña placa de silicio. El proceso de creación de un circuito integrado es largo y complejo; está basado en diversas técnicas ópticas y químicas. Se usan las técnicas antes descritas para el transistor —dopaje del semiconductor con impurezas— solo que aplicadas de manera masiva y en miniatura.

El circuito integrado permite la miniaturización de circuitos con millones de transistores, con el consiguiente ahorro en espacio, tiempo, costo y consumo de energía. Permite la creación de computadores muy complejos y con gran capacidad, pero compactos y de costo razonable.

Los microprocesadores

En 1969, Intel recibió el encargo de diseñar un conjunto de chips para Busicom, un fabricante japonés de calculadoras. Un ingeniero de Intel, Ted Hoff, pensó que el diseño sería más económico si varios de estos chips se reducían a un solo chip programable; Intel estuvo de acuerdo con el planteamiento, y Hoff, junto con Federico Fagin y Stanley Mazor, procedió a realizar el diseño.

El resultado fue el Intel 4004: un procesador de 4 bits, construido con 2.300 transistores. Intel vio el potencial que tenía el invento, y decidió quedarse con el producto en lugar de vendérselo a Busicom.

Es de resaltar que en ese momento no se estaba pensando en el microcomputador; los microprocesadores eran útiles como controladores programables para diversos artefactos electrónicos.

⁹ Los transistores, como los triodos, son más que conmutadores: pueden amplificar las señales. Sin embargo, en computación, su uso principal es como conmutadores.

Intel diseñó otro procesador en esta línea: el Intel 8008; primer procesador de 8 bits. Con este procesador y, especialmente, con su sucesor —el Intel 8080— se inaugura la época del microcomputador.

El microprocesador es una consecuencia natural de los circuitos integrados: dado que estos permiten integrar una gran cantidad de transistores, es posible pensar en construir un procesador completo en un chip.

Por supuesto, ni la historia ni la evolución de los computadores acaban en este punto; la evolución tecnológica ha continuado, y el resultado ha sido computadores cada vez más veloces, con mayor capacidad y menor tamaño. Han aparecido nuevas ideas, y nuevos dispositivos, algunos de los cuales han originado revoluciones a título propio —baste con pensar en las redes de computadores—.

Sin embargo, sí son estos los elementos fundamentales que conducen al computador tal como lo concebimos actualmente, y son suficientes para nuestros propósitos.