

La Informática y los ordenadores.

La Informática y los ordenadores

1. Historia reciente del cálculo personal.....	2
1.1 Representación y cálculo.....	2
1.2 Los algoritmos de la aritmética elemental.....	2
1.3 las tablas matemáticas.....	2
1.4 Los logaritmos y la regla de cálculo.....	2
1.5 El cálculo electrónico digital.....	3
1.6 La calculadora electrónica.....	3
2. La génesis del ordenador moderno.....	3
2.1 Representación y cálculo.....	3
2.2 Charles babbage y ada byron: los precursores.....	4
2.3 Los primeros ordenadores electrónicos.....	4
3. Evolución histórica de los lenguajes de programación.....	4
3.1 ¿Qué es un lenguaje de programación?.....	5
3.2 La librería estándar.....	5
3.3 Niveles de los lenguajes de programación.....	5
3.4 Las generaciones de los lenguajes.....	6
3.5 Paradigmas de programación.....	6
3.6 Programación imperativa.....	6
4. Sistemas operativos: del panel de control a las interfaces gráficas.....	6
4.1 Trabajar sin sistema operativo.....	7
4.2 Sistemas por lotes.....	7
4.3 Los sistemas multiprogramados.....	7
4.4 Sistemas de tiempo compartido.....	7
4.5 Sistemas operativos de libre distribución.....	8

1. Historia reciente del cálculo personal

La informática es una disciplina muy ligada al cálculo aritmético. De hecho, los primeros ordenadores deben mucho al diseño de los aparatos de cálculo, principalmente a las calculadoras mecánicas y electromecánicas. En este trabajo dirigimos nuestra mirada hacia las tres principales herramientas de cálculo con las que han tenido que trabajar los estudiantes y profesionales de perfil técnico durante el siglo xx, hasta que el computador las sustituyó: las tablas matemáticas, la regla de cálculo y las calculadoras electrónicas. Cada una de estas herramientas tiene su historia.

1.1 Representación y cálculo

La representación numérica que practicamos los humanos tiene dos objetivos, la anotación y el cálculo. El sistema de representación más extendido actualmente entre los humanos, el sistema posicional de base decimal, debe su éxito a las facilidades que da al cálculo elemental que practicamos todos los días. Un sistema de representación un mérica como el romano, que no facilita los cálculos, acaba restringido a usos ornamentales o bien desaparece.

Nuestros procesos de cálculo abstraen los objetos que los motivan. Si hemos de sumar distancias, convertimos cada distancia en un símbolo abstracto, el numeral, que la representa. Si hemos de sumar precios de la compra, lo hacemos también abstrayendo el precio de cada producto en un numeral. Los numerales que utilizamos son independientes de los objetos representados y por eso sirven tanto para distancias como para precios o cualquier otra necesidad. Igualmente, el cálculo humano es un proceso que manipula símbolos y es ciego a los objetos concretos representados por los numerales.

A los procedimientos de cálculo abstracto solemos llamarlos algoritmos.

1.2 Los algoritmos de la aritmética elemental

Los algoritmos de cálculo que aprendemos en la escuela se limitan a las cuatro operaciones básicas de suma, resta, multiplicación y división. En nuestro entorno, el uso de los ábacos no se ha difundido mucho, y hemos insistido fundamentalmente en la representación gráfica del cálculo con lápiz y papel. De hecho, hay diferentes estilos gráficos de construcción del resultado según los países y sus tradiciones escolares.

1.3 las tablas matemáticas

Si bien los algoritmos manuales de suma, resta, multiplicación y división están al alcance de cualquiera, la obtención de las funciones transcendentales (por ejemplo, coseno, tangente, logaritmo, etc.) plantea dificultades mucho más grandes. La ayuda más antigua para calcular valores de una función matemática es la difusión de tablas copiadas de una tabla original. A partir del siglo xv, la imprenta permitió copiar las tablas a gran escala, incluyendo en ellas los errores cometidos por los creadores.

1.4 Los logaritmos y la regla de cálculo

Durante el año 2014 se cumplió el cuarto centenario de la publicación de *Mirifici Logarithmorum Canonis Descriptio* (1614), el texto donde John Napier of Merchiston enunció la teoría de los logaritmos y su utilidad y publicó las primeras tablas. La poca relevancia de esta efeméride se debe a que la calculadora electrónica de bolsillo le quitó la utilidad. Pero en los estudios de bachillerato,

hasta los años 70 del siglo xx, los alumnos aprendían el uso de las tablas de logaritmos y los estudiantes y profesionales de ciencias e ingeniería tenían que ser diestros con la regla de cálculo, un dispositivo manual que aprovecha sus propiedades. Las escuelas disponían de modelos a gran escala de las reglas de cálculo para que los instructores enseñaran su uso.

1.5 El cálculo electrónico digital

La electrónica es una parte de la ingeniería que se ha desarrollado a lo largo del siglo XX. El primer componente claramente electrónico, el triodo, fue inventado en 1906. Los ingenieros se interesaron por la velocidad y falta de inercia propias de los circuitos

electrónicos y pronto los aprovecharon para representar y transformar información.

Pero obviaremos las realizaciones de los precursores conocidas como “analógicas” e iremos directamente a los diseños “digitales”, donde los circuitos representan y manipulan bits. Y un bit solo puede tener dos valores posibles, el 0 y el 1.

1.6 La calculadora electrónica

La historia de las calculadoras electrónicas, es decir, de los dispositivos electrónicos especializados en cálculos matemáticos de forma interactiva, comienza con los años 60. En aquel momento los primeros sistemas operativos interactivos (MULTICS, UNIX) todavía estaban por desarrollar, de manera que los computadores operaban por lotes (en batch) y no se adaptaban bien al uso personal, sin contar su precio, el consumo de energía, el volumen y la dificultad de operación. Un computador, para trabajar, necesita un programa que especifique la serie de operaciones y unos datos con los que comenzar.

2. La génesis del ordenador moderno

La omnipresencia de la informática en la sociedad actual es tal que, paradójicamente, la convierte en invisible. Sin embargo, vale la pena recordar que el ordenador, representante por antonomasia de esta informática ubicua y en todos sus formatos computador personal, tableta, teléfono inteligente, no siempre ha estado ahí. Su historia, al menos su devenir en lo que hoy entendemos por ordenador, comienza poco antes de la mitad del siglo XX. Esta contribución pretende contar, de una manera sintética, las circunstancias que propiciaron el advenimiento de los primeros ordenadores electrónicos y la aparición del concepto de ordenador moderno.

2.1 Representación y cálculo

La representación numérica que practicamos los humanos tiene dos objetivos, la anotación y el cálculo. El sistema de representación más extendido actualmente entre los humanos, el sistema posicional de base decimal, debe su éxito a las facilidades que da al cálculo elemental que practicamos todos los días. Un sistema de representación numérica como el romano, que no facilita los cálculos, acaba restringido a usos ornamentales o bien desaparece.

Nuestros procesos de cálculo abstraen los objetos que los motivan. Si hemos de sumar distancias, convertimos cada distancia en un símbolo abstracto, el numeral, que la representa. Si hemos de sumar precios de la compra, lo hacemos también abstrayendo el precio de cada producto en un numeral. Los numerales que utilizamos son indepen-

dientes de los objetos representados y por eso sirven tanto para distancias como para precios o cualquier otra necesidad. Igualmente, el cálculo humano es un proceso que manipula símbolos y es ciego a los objetos concretos representados por los numerales. A los procedimientos de cálculo abstracto solemos llamarlos algoritmos.

2.2 Charles babbage y ada byron: los precursores

La primera concepción de un ordenador universal es obra del matemático británico Charles Babbage, un pionero del cálculo automático atraído por la exactitud y obsesionado con la idea de eliminar los errores de las tablas matemáticas utilizadas para ayudar al cálculo en disciplinas como la estadística, la economía o la navegación. Babbage detentó la cátedra de matemáticas de Cambridge, un puesto que ocupó anteriormente Isaac Newton y, hace poco tiempo, Stephen Hawking. Era bien conocido en los círculos victorianos por su gran espíritu crítico y también por un carácter vehemente e irascible que le llevó a perseguir de manera implacable a los músicos callejeros porque, según él, le molestaban mientras trabajaba en su casa londinense.

La Máquina Analítica (*Analytical Engine*) concebida por Babbage, uno de los proyectos tecnológicos más avanzados del siglo, permitía efectuar las cuatro operaciones aritméticas elementales e incluía la posibilidad de tomar decisiones en base a los resultados intermedios. Esta última característica convertía a este dispositivo en algo más que una mera calculadora al servicio de un calculista humano: se trataba, de hecho, del verdadero antecesor del ordenador moderno.

2.3 Los primeros ordenadores electrónicos

Debemos señalar, antes de nada, que en nuestra opinión es poco relevante, e incluso inútil, intentar señalar, de forma indiscutible, cuál fue el “primer” ordenador electrónico de la historia. Como veremos en seguida, el nacimiento de los ordenadores modernos fue en gran medida fruto del interés y trabajo de muchas personas, en solitario o dentro de un equipo de trabajo, y de distintas instituciones, con intereses divergentes y en localizaciones y países diversos. Por otro lado, podemos afirmar que, en general, el trabajo de Babbage y Ada Byron, poco conocido entre los círculos especializados, tuvo escasa influencia en la concepción del ordenador moderno. En todo caso, lo que sí parece claro es que fue la Segunda Guerra Mundial (1939-1945) el acontecimiento que hizo de catalizador en la combinación de los ingredientes necesarios para la invención del ordenador electrónico.

El gobierno inglés, por ejemplo, invirtió grandes esfuerzos en la construcción de una serie de máquinas electrónicas para descifrar los mensajes que la marina alemana codificaba mediante la conocida máquina Enigma. Interceptar e interpretar correctamente los mensajes que contenían las órdenes de destrucción a los submarinos alemanes desplegados en el océano Atlántico se convirtió en un objetivo crucial. En esta tarea, desarrollada en Bletchley Park, al noroeste de Londres, participaron matemáticos de la talla de Alan Turing i Max Newman.

3. Evolución histórica de los lenguajes de programación

En la presente revisión de la evolución de los lenguajes de programación vamos a evitar un repaso exhaustivo de todos ellos, en parte porque no aportaría una mejor perspectiva pero sobre todo porque la lista es de tal magnitud que resulta imposible de abarcar. Remitimos al lector a otras publicaciones más completas donde podrá ver cómo la mayoría de estudios hablan de como máximo 50 lenguajes de programación de entre los aproximadamente 2.500 que se supone existen o

han existido.

3.1 ¿Qué es un lenguaje de programación?

Existen varias definiciones de lenguaje de programación. Desde un punto de vista práctico se puede definir como el *instrumento que permite a los humanos comunicar instrucciones u órdenes a las computadoras*. Es obvio que no puede admitir ambigüedades como es el caso del lenguaje natural que utilizamos los seres humanos para comunicarnos. El hardware ejecuta exactamente lo que se le dice que ejecute y en el orden exacto que se le ha dicho. Con esta idea en mente se entenderá mejor la siguiente definición de lenguaje de programación: *lenguaje formal definido artificialmente para expresar algoritmos que, instrucción a instrucción, ejecutará una computadora*. Se utiliza el adjetivo “formal” porque un lenguaje de programación está sujeto a una serie de reglas sintácticas que definen sin ambigüedad la semántica de las expresiones que podemos utilizar para indicar las operaciones al ordenador.

3.2 La librería estándar

Es común que los lenguajes de programación vayan acompañados de su librería estándar y, cada vez más, de su entorno de ejecución. Muchos lenguajes de programación llevan asociada una librería de funciones básicas como pueden ser las funciones matemáticas más habituales o las de entrada/salida. También se incluyen algunas estructuras de datos como soporte para utilizar algunas funciones de la librería.

Este conjunto de funciones se conoce como librería estándar. Es difícil determinar si dichas funciones forman o no parte del lenguaje como tal, pero sin embargo, los programas necesitan tenerlas disponibles en el momento de su ejecución. Hasta mediados de los años 90 del siglo XX, la librería estándar o parte de ella se incluía en el fichero ejecutable resultado del proceso de compilación. La evolución de los sistemas operativos ha traído consigo que no sea necesario incluir dentro del fichero ejecutable el código compilado correspondiente a la librería estándar, por ello, desde hace unos años se habla de entorno de ejecución para un lenguaje de programación.

Este entorno puede ser un conjunto de librerías dinámicas que el sistema operativo cargará en memoria cuando el programa lo necesite durante su ejecución, o una máquina virtual como es el caso de Java.

3.3 Niveles de los lenguajes de programación

Algunos intentos de clasificar los lenguajes de programación han seguido diversos criterios. Antes de pasar a la siguiente sección que versa sobre los paradigmas de programación, vamos a ver otros dos criterios utilizados a lo largo de la historia de los lenguajes de programación para clasificarlos: por niveles y por generaciones.

El nivel más bajo de programación es programar directamente utilizando el lenguaje máquina, codificando las instrucciones y los datos mediante ceros y unos. Esta práctica comenzó nada más aparecer los primeros ordenadores en las décadas de los 40 y los 50 del siglo XX. Nos podemos imaginar la cantidad de horas necesaria para escribir un simple programa y la cantidad de errores que se cometían. Era una labor altamente compleja y poco gratificante. Era la época de las tarjetas perforadas, aunque éstas comenzaron a utilizarse mucho antes para otros propósitos. Las instrucciones del código máquina se grababan en las tarjetas perforadas línea a línea, y cada línea contenía la correspondiente combinación de ceros y unos.

3.4 Las generaciones de los lenguajes

Aprovechando la división anterior en dos niveles, los lenguajes de programación también se clasifican por generaciones. El código máquina se considera representante de los lenguajes de primera generación. Cada diseño de procesador tiene su propio código máquina. Los ensambladores son considerados como lenguajes de segunda generación.

Esta diferenciación comenzó a utilizarse cuando se acuñó el término tercera generación para identificar los lenguajes de programación de alto nivel que son independientes del hardware. Los lenguajes más representativos de la tercera generación son C, Ada, C++, C# y Java.

3.5 Paradigmas de programación

Hemos visto que los lenguajes de programación pueden clasificarse por niveles y por generaciones. No existe una manera única de clasificarlos, varios de ellos comparten características que se consideran relevantes en distintos tipos o paradigmas de programación.

Otro detalle que complica cualquier intento de agrupar los lenguajes de programación en base a un criterio concreto es el hecho de que la mayoría de ellos nacieron a partir de ideas y características ya existentes en otros lenguajes más antiguos.

Como puede observarse en los diagramas de Levenez (2015) y la editorial O'Reilly (2015), salvo los primeros lenguajes y algún caso particular, el resto tienen varios antecesores, es decir, se definen partiendo de características presentes en lenguajes ya existentes.

3.6 Programación imperativa

La programación imperativa es un paradigma de programación basado en instrucciones que debe ejecutar el ordenador para cambiar el estado del programa. Recuerda el estilo imperativo del lenguaje natural, en el que una persona da órdenes que otros ejecutan.

Un programa escrito en un lenguaje que sigue este paradigma es visto como una secuencia de instrucciones que la computadora ejecutará en el mismo orden. Los lenguajes ensambladores son de este estilo: utilizan códigos nemotécnicos para expresar el lenguaje máquina que un procesador hardware es capaz de ejecutar. Los primeros lenguajes que aparecieron también seguían este paradigma, por ejemplo COBOL, FORTRAN, ALGOL o PL/1.

Los paradigmas de programación siguen cierta organización jerárquica fruto de la evolución a lo largo del tiempo. El que vamos a ver a continuación, programación estructurada o procedural, es un derivado de la programación imperativa. Incluso la programación orientada a objetos incluye muchas de las características de la programación imperativa, ya que el código que compone los cuerpos de los métodos sigue este paradigma.

4. Sistemas operativos: del panel de control a las interfaces gráficas

A lo largo de la historia los sistemas operativos han sido imprescindibles para facilitar el trabajo con el computador y conseguir generalizar su uso a toda la población.

En sus inicios únicamente realizaban funciones básicas y para conseguirlo necesitaban el apoyo de un operador humano. Los sistemas actuales tienen una gran responsabilidad y multitud de funciones. Además, suelen estar acompañados de interfaces gráficas que permiten un uso muy intuitivo del computador. Este artículo pretende ser un resumen divulgativo de los hechos, modos de trabajo y técnicas más remarcables que forman parte de la historia y evolución de los sistemas operativos.

4.1 Trabajar sin sistema operativo

Antes de la Segunda Guerra Mundial (1939-1945) los humanos eran los encargados de realizar los cálculos, es decir, los verdaderos computadores (human computer). Durante casi doscientos años, estos grupos de computadores humanos fueron organizados de forma jerárquica ideando procedimientos de cálculo estándar y aprendiendo a dividir cálculos complejos, como la órbita de un cometa o la trayectoria de un misil antiaéreo, en millones de pequeñas operaciones para abordarlos. Algunos grupos formaron verdaderas oficinas de cálculo que preparaban tablas matemáticas para los científicos de la Bell Company o procesaban datos para el gobierno de los Estados Unidos. Pues bien, el proceso de organización jerárquica y auto chequeo llevado a cabo por estos grupos se puede relacionar directamente con las funciones de los actuales sistemas operativos. El ENIAC (Electronic Numerical Integrator and Computer), considerado como el primer ordenador electrónico programable, se acabó de construir a finales de la Segunda Guerra Mundial (Figura 4.2). En esta máquina los datos y programas se introducen a través de la interconexión de tableros con interruptores (plug boards) y se trabajaba de manera directa con el hardware. Su tecnología se basaba en tubos de vacío, que venían a realizar la misma función que los transistores. Los lenguajes de programación eran desconocidos y, por supuesto, resultaría inaudito hablar de SO.

4.2 Sistemas por lotes

A principios de la década de 1950, una vez establecido en la comunidad científica el concepto de programa almacenado, se crea una nueva dinámica de trabajo con el computador. Se sigue trabajando con tarjetas perforadas pero también se incorporan las cintas magnéticas para almacenar la información. Las tarjetas perforadas, una tecnología de almacenamiento usada desde finales del siglo XIX, son cartulinas con 80 columnas y 12 filas (Figura 4.3), cuyas posiciones pueden estar perforadas o no, representando así un código binario. En esta época también aparece una clara distinción entre diseñadores, constructores, operadores, programadores, usuarios y personal de mantenimiento del computador.

4.3 Los sistemas multiprogramados

Para entender el concepto de multiprogramación pensemos en un estudiante que debe realizar dos experimentos, uno de química y otro de biología. El estudiante prepara el experimento de biología hasta que su compuesto necesita reposo, mientras está reposando prepara el de química, mientras el de química reposa atiende al de biología y así sucesivamente. Al final tendremos la impresión de que el estudiante ha estado trabajando en dos experimentos a la vez.

Los sistemas multiprogramados pueden ir alternando la ejecución de instrucciones de diferentes tareas que están almacenadas en memoria. Los usuarios tienen la percepción de que la máquina puede ejecutar varias tareas a la vez. La ejecución simultánea de tareas exige el reparto, entre ellas, de los componentes del computador. Así la máquina dedica tiempo de procesador (CPU) de forma alterna a diferentes trabajos (Figura 4.4).

4.4 Sistemas de tiempo compartido

El concepto de tiempo compartido es una idea novedosa que intenta dar soporte interactivo a los computadores a través de sus periféricos. Los programas pueden, por ejemplo, escribir textos en pantalla solicitando al usuario que teclee datos. En este tipo de sistema los procesos intercalan operaciones de entrada y de salida con operaciones de cálculo. Corbató, considerado el padre del tiempo compartido, en 1962 lo define textualmente así: *the basic technique for a time-sharing*

system is to have many persons simultaneously using the computer through typewriter consoles with a time-sharing supervisor program.

Una sucesión de sistemas operativos desarrollaron progresivamente esta capacidad: CTSS (*Compatible Time-Sharing System*, 1961), Multics (*Multiplexed Information and Computing Service*, 1965) y Unix. Además, Multics fue el primer sistema operativo que introdujo un sistema de archivos jerárquico, siendo el precursor de las actuales carpetas y documentos.

4.5 Sistemas operativos de libre distribución

En 1991 se produce un gran revuelo en los entornos informáticos con la aparición del sistema Linux. Linux es el primer sistema operativo de libre distribución compatible con Unix. Linux es una iniciativa abierta en la que cualquier desarrollador puede implementar y aportar código a este sistema y cualquier usuario puede utilizarlo sin pagar por él, es decir, sin adquirir su licencia. Mientras tanto, los sistemas operativos comerciales son desarrollados por empleados pagados por empresas que lo distribuyen bajo pago de licencia. Cada vez que aparece un nuevo tipo de periférico los usuarios de Linux deben esperar a que algún desarrollador generoso implemente el código imprescindible para que funcione. Los sistemas Linux están ampliamente difundidos y son utilizados por todo tipo de usuarios.

<https://museo.inf.upv.es/wp-content/uploads/2021/04/Un-viaje-a-la-historia-de-la-informatica.pdf>