

UN VIAJE A LA HISTORIA DE LA INFORMÀTICA

Escola Tècnica Superior d'Enginyeria Informàtica  etsinf



EDITORIAL
UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Un viaje a la historia de la informática

Coordinador

Xavier Molero Prieto

2016

**EDITORIAL
UNIVERSITAT POLITÈCNICA DE VALÈNCIA**

Edita

Escola Tècnica Superior d'Enginyeria Informàtica

Coordinador

Xavier Molero Prieto

Editorial

Editorial Universitat Politècnica de València

distribución: Telf.: 963 877 012 / www.lalibreria.upv.es / Ref:6309_01_01_01

ISBN: 978-84-9048-491-3 (Versión impresa)

Impreso bajo demanda



Un viaje a la historia de la informática por Escola Tècnica Superior d'Enginyeria Informàtica
Se distribuye bajo una licencia de Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional.

Me es grato comenzar la presentación de esta publicación felicitando al grupo de profesores y profesoras de la Escola Tècnica Superior d'Enginyeria Informàtica de la Universitat Politècnica de València, y a quienes han contribuido a la creación de este libro titulado *Un viaje a la historia de la informática*. A todos ellos, expreso mi sincera enhorabuena y reconocimiento por plasmar de forma divulgativa los orígenes de la informática y la vertiginosa trayectoria de intensidad exponencial que ha ido trazando esta ciencia de ciencias sin límites.

Conocer la historia ayuda a entender el presente y, muchas veces, permite predecir el futuro. En materia de informática, aunque es difícil determinar el origen exacto, se sabe que sus comienzos marcaron el fin de una era. Sin embargo, calcular la repercusión que esta potente disciplina puede generar en el futuro se convierte en una tarea del todo complicada. Incluso, con vista a un futuro inmediato, resulta impredecible determinar los derroteros hacia los que derivará la informática.

Lo que sí somos capaces de apreciar es la infinidad de posibilidades que esta ciencia multidisciplinar por excelencia ofrece hoy por hoy, y que la informática ha logrado posicionarse como una poderosa herramienta indispensable en todas las áreas de trabajo y en la forma en la que la sociedad actual concibe la vida diaria.

Un viaje a la historia de la informática muestra de manera atractiva el papel clave que ha jugado la combinación de ingenio y tecnología en el acelerado desarrollo de la informática. De lectura didáctica y agradable, el libro permite entender el concepto de la informática, desde la perspectiva más amplia de la palabra; explica su capacidad de nutrirse de conocimientos básicos, de las leyes físicas, de datos matemáticos, de fórmulas químicas, para realizar análisis, interpretaciones, hallar respuestas y, en definitiva, solucionar problemas.

El contenido de este libro, la acertada disposición de su materia y las curiosidades que describe despiertan el interés de quien lo lee, e invitan a conocer más sobre el tema. Una manera distendida y amena de hacerlo, de conocer más sobre la informática, recorriendo el pasado, echando un vistazo al presente y asomándose a la ventana del futuro, es sin duda, a través del Museo de Informática de la Escola Tècnica Superior d'Enginyeria Informàtica de la Universitat Politècnica de València. Les animo a que lo visiten y deseo que disfruten de su completa e interesante muestra.

Francisco José Mora Mas
Rector
Universitat Politècnica de València

Desde que en 2001 se inauguró el Museo de Informática de la Escola Tècnica Superior d'Enginyeria Informàtica en la Universitat Politècnica de València, son muchas las iniciativas que se han llevado a cabo en torno a la difusión y divulgación de la informática. Hay que agradecer al profesor Tomás Pérez su dedicación y empeño en la etapa inicial. Una labor que ahora continúa el profesor Xavier Molero como coordinador de un museo reconocido por la Generalitat Valenciana y por el que pasa un gran número de escolares cada año, recorriendo su espacio expositivo en la propia escuela y realizando las actividades que se han diseñado con objeto de dar a conocer la informática entre un público que está llamado a ser el protagonista de este mundo digital que nos rodea.

El Museo de Informática de la ETSINF está empeñado en preservar y recordar lo que la informática es, cual Funes el memorioso, que retenía cada imagen como un recuerdo imborrable. De esta manera, tanto los fondos como las actividades culturales del museo pretenden “recordar” y hacernos ver cómo ha evolucionado la informática en las últimas décadas. Es un empeño posible gracias a las donaciones que conforman el patrimonio de este museo y a la colaboración desinteresada de aquellos y aquellas que durante este tiempo han querido ayudar en esta tarea de fomento y difusión de la informática.

Ahora nos llega este libro que plasma en texto lo que el museo transmite con sus actividades y material expositivo a los visitantes. Un texto que no hubiera sido posible sin las aportaciones de los profesores y profesoras que han colaborado en la redacción de los diferentes capítulos. Contar con su experiencia y conocimiento ha sido esencial para la gestación de este volumen.

Lo que hoy tenemos entre manos es la culminación palpable de una etapa divulgativa. Un volumen que resume lo que es la informática a través de un recorrido que repasa hitos, génesis y evolución de los computadores, los sistemas operativos y los lenguajes de programación, los grandes sistemas y los microprocesadores, las aplicaciones, la vertiente profesional o los desarrollos actuales en el ámbito de la robótica o la fabricación digital. Un periplo que integra el pasado con la actualidad, para acabar vislumbrando lo que el futuro nos depara.

Desde esa ventana al futuro, apreciamos un horizonte excitante en el que la informática es protagonista esencial de nuestra sociedad. Un futuro en que la tecnología se entrelaza de manera natural con nuestra vida. Una vida en la que, como decía Arthur C. Clarke, los avances tecnológicos serán indistinguibles de la magia.

Para ayudar a descifrar la magia de lo cotidiano hay que tener presente el desarrollo de lo pasado. Y ese es el cometido de este libro, ser un referente de lo que ha sido y es la informática.

Disfrútenlo.

Eduardo Vendrell Vidal
Director de la ETS d'Enginyeria Informàtica

PREFACIO

No resulta fácil definir en qué momento comienza la historia de la informática. Encontramos sus raíces en el pasado remoto cuando rastreamos las diversas abstracciones que nuestra cultura ha desarrollado para manipular la información cotidiana. Nos parecerá más reciente si observamos la tecnología electrónica en la que se basan muchos aparatos que ahora son de uso corriente pero que nuestros abuelos no pudieron ni siquiera imaginar. Y nos sabrá a pura actualidad si nos fijamos en los cambios que está induciendo en nuestros hábitos sociales. Si, para no perdernos, tomamos como punto de partida la invención de los primeros ordenadores electrónicos de finales de los años 40 del pasado siglo XX, nos daremos cuenta que estas *nuevas tecnologías* contienen décadas de personajes, de ideas y de realizaciones. O sea, más de medio siglo de historia que conviene estudiar y contar.

Desde su creación en el año 2001, el Museo de Informática está involucrado activamente en la conservación y difusión del patrimonio informático. Una doble misión que, en primer lugar, pretende que la sociedad sea consciente de los aspectos clave de la informática entendida como disciplina científica: orígenes, evolución histórica y ámbitos de desarrollo. Y en segundo lugar, tan importante como la anterior, intenta incentivar la reflexión crítica sobre el papel que la informática ha jugado y juega actualmente en el desarrollo de las distintas sociedades de nuestro planeta.

Ambos objetivos han servido de inspiradores para este viaje colectivo a la historia de la informática. Un viaje fruto de la colaboración desinteresada de un grupo de profesores y profesoras con gran experiencia docente, pertenecientes la mayoría a la Escola Tècnica Superior d'Enginyeria Informàtica de la Universitat Politècnica de València. Un trabajo, en definitiva, concebido y desarrollado con un espíritu didáctico y divulgativo que pretende llegar al máximo de personas posible.

Como punto de partida de este viaje particular y diverso a la historia de la informática hemos elegido el desarrollo reciente de los dispositivos de cálculo personal, desde las reglas de cálculo hasta el advenimiento de las primeras calculadoras electrónicas.

En este viaje no podían faltar ni los primeros ordenadores ni los primeros lenguajes de programación. Si conocer las circunstancias que envolvieron el diseño y la construcción de los primeros ordenadores electrónicos puede resultar fascinante, igual de interesante resultará saber cómo se desarrollaron y hacia dónde evolucionaron los más de dos mil quinientos lenguajes de programación que, según algunas estimaciones, se supone que han existido hasta la fecha. Y no podrían faltar en este recorrido histórico los sistemas operativos, desde sus versiones más elementales gestionadas por personal especializado hasta las interfaces gráficas actuales que hacen del ordenador un sistema accesible a cualquier persona.

Tampoco nos hemos olvidado de la parte lúdica que ofrece la informática. De igual forma que existen evidencias de que el juego es una actividad propia de las primeras

civilizaciones humanas, también sabemos que, tan pronto como existieron los primeros ordenadores, se empezaron a desarrollar programas con los que dar rienda suelta a nuestro afán innato de diversión y entretenimiento. No en vano el mercado actual de videojuegos representa una de las industrias más importantes de nuestra sociedad.

El viaje que hemos planificado también recorre otros ámbitos como, por ejemplo, el origen y desarrollo de Internet hasta convertirse en la red de comunicación entre ordenadores a escala mundial que está cambiando los modos de hacer y de pensar de nuestra sociedad del siglo XXI.

También se recorren campos como la aparición de la profesión informática en la sociedad así como la relación existente entre ética e informática. En particular, y centrándonos en nuestro propio país, se examinan los cambios producidos en el ámbito de la profesionalización de la informática en las últimas décadas del siglo XX y su inclusión dentro del espacio docente universitario.

Otra temática que hemos visitado es el diseño e implementación de los sistemas informáticos. Por un lado se presentan las distintas alternativas empleadas a lo largo de la historia para implementar los superordenadores utilizados en la resolución de problemas complejos como, por ejemplo, los modelos matemáticos usados en la predicción meteorológica. Por otro, damos un paseo por la historia de los microcontroladores, esos dispositivos responsables del diseño de ordenadores de pequeño tamaño que, a pesar de estar hoy en día por todas partes —automóviles, videoconsolas, cajeros automáticos, etc.—, resultan invisibles.

Las últimas etapas de nuestro viaje hacia el pasado incluyen sendas visitas a dos campos donde la informática juega un papel esencial. El primero de ellos es la robótica, una disciplina que, desde sus inicios con el diseño de autómatas, ha evolucionado de manera incesante en las últimas décadas gracias a los avances en terrenos informáticos como la inteligencia artificial o la arquitectura de ordenadores. El segundo campo donde nos hemos detenido es la historia de la impresión 3D, una tecnología que está permitiendo avances espectaculares en disciplinas tan importantes como la medicina o la arquitectura, entre otras muchas.

En definitiva, este libro plantea una docena de sólidas y variadas inmersiones en la historia de la informática, un océano plétórico de vistas y detalles que pide ser recorrido con los ojos bien abiertos y los pulmones llenos de aire.

Pero, si en algún momento hace falta salir a la superficie para tomar oxígeno, este viaje multidisciplinar también nos lo permite. Solamente habrá que desplazarse hasta la última etapa. Allí encontraremos una ventana abierta al futuro a través de la cual podremos intuir cómo serán los ordenadores de los próximos años. ¿O quizás son ya los ordenadores del presente?

Xavier Molero Prieto
Responsable del Museo de Informática

ÍNDICE

Prólogo Rector	I
Prólogo Director de la ETS d'Enginyeria Informàtica	III
Prefacio Responsable del Museo de Informática	V
Capítulo 1 Historia reciente del cálculo personal	1
Capítulo 2 La génesis del ordenador moderno	15
Capítulo 3 Evolución histórica de los lenguajes de programación	27
Capítulo 4 Sistemas operativos: del panel de control a las interfaces gráficas	41
Capítulo 5 Del osciloscopio al Smartphone, un recorrido por el videojuego	57
Capítulo 6 Apuntes sobre los orígenes de internet	71
Capítulo 7 La profesionalización de la informática en España	81
Capítulo 8 Ética e informática, un binomio en permanente evolución	93
Capítulo 9 Los grandes sistemas informáticos	103
Capítulo 10 Microcontroladores	113
Capítulo 11 La informática en el desarrollo de la robótica	127
Capítulo 12 La evolución de la impresión 3D	141
Capítulo 13 Una ventana al futuro	155

Álvaro Doménech Pujol
Universitat Politècnica de València

Resumen: 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.

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 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*.

Los aparatos que calculan o que nos ayudan al cálculo trabajan sobre representaciones mecánicas o electrónicas equivalentes a nuestros numerales. Los inventores de los dispositivos, según el caso, tuvieron que escoger un sistema de representación ya existente o inventarse uno nuevo. Y un dispositivo de cálculo aplica un algoritmo.

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.

Todos estos procedimientos funcionan porque partimos de una notación numérica en la que cada posición decimal tiene dos posiciones vecinas, una a la derecha donde los dígitos valen la décima parte y otra a la izquierda donde valen diez veces más. Así cobran sentido los acarreo de la suma, los préstamos de la resta, los desplazamientos hacia la izquierda en la multiplicación y los desplazamientos hacia la derecha en la división.

En la escuela aprendemos dos estilos de representación, como muestra la Figura 1.1. La representación en coma fija, la más habitual, consiste en una cadena de dígitos con una coma que separa la parte entera de la parte fraccionaria. Si falta la coma, entendemos que la parte fraccionaria es nula. Tenemos un vocabulario para nombrar las posiciones: unidades, decenas, centenas, etc. a la izquierda de la coma, décimas, centésimas y demás a su derecha. Esta representación es suficiente en muchos ámbitos, como los ejercicios escolares, la aritmética doméstica o la contabilidad comercial.

La coma flotante, que aprendemos en la escuela secundaria, consiste en representar una mantisa M , un 10 (la base) y un exponente E por separado. Los valores se escriben de la forma $\pm M \cdot 10^{\pm E}$, y hay que adaptar la aritmética común para ellos: por ejemplo, multiplicar dos valores implica multiplicar las mantisas y sumar los exponentes.

$$\begin{array}{r}
 375 \\
 \times 615 \\
 \hline
 1875 \\
 3750 \\
 22500 \\
 \hline
 230625
 \end{array}$$

$$\begin{array}{r}
 3,75 \times 10^2 \\
 \times 6,15 \times 10^5 \\
 \hline
 23,0625 \times 10^7 \\
 \downarrow \\
 2,30625 \times 10^8
 \end{array}$$

Fuente: elaboración propia

Figura 1.1. Ejemplos de multiplicación humana. A la izquierda tenemos una multiplicación en coma fija, donde podemos apreciar como los productos parciales se desplazan progresivamente hacia la izquierda formando la figura en escalera que conocemos. En realidad, el desplazamiento hacia la izquierda equivale a multiplicar por 10, y podemos entender que la multiplicación produce los tres productos parciales 1875, 3750 y 225000 que hay que sumar para obtener el producto final. A la derecha, tenemos una multiplicación en coma flotante. La mantisa del producto se obtiene multiplicando las mantisas, igual que en coma fija, y los exponentes se suman. En este caso, hay normalizar el resultado incrementando el exponente y desplazando la coma. Esta manipulación manifiesta, de nuevo, la relación entre los desplazamientos y los productos y divisiones por 10.

LAS TABLAS MATEMÁTICAS

Si bien los algoritmos manuales de suma, resta, multiplicación y división están al alcance de cualquiera, la obtención de las funciones trascendentes (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.

La construcción de la tabla original la hacía un experto trabajando de forma metódica, partiendo de valores singulares conocidos y aplicando procedimientos algebraicos para deducir otros. Por ejemplo, Tolomeo (s. II) incluyó tablas de funciones trigonométricas en su libro *Almagesto*, calculadas a partir de los ángulos singulares 0, 90°, 180°, etc. y de las identidades que hoy todavía se estudian en la escuela secundaria, como son $\cos(\alpha-\beta) = \cos \alpha \cdot \cos \beta - \sin \alpha \cdot \sin \beta$ o bien $\cos(\alpha/2) = ((1-\cos \alpha)/2)^{1/2}$. En el siglo XVI Henry Briggs construyó una tabla de logaritmos decimales a partir de los valores singulares $\log(1) = 0$ y $\log_{10}(10) = 1$; a continuación, aprovechando propiedades como $\log(\sqrt{x \cdot y}) = \frac{1}{2}(\log(x) + \log(y))$, fue obteniendo los logaritmos de otros valores.

Las bibliotecas de muchas universidades (Figura 1.2) todavía disponen de ediciones de tablas de logaritmos y trigonometría, que aparecen usualmente en dos tipos de textos: bien en monografías dedicadas a las tablas o bien en manuales profesionales de las diversas ramas de arquitectura e ingeniería que dedicaban secciones importantes (a menudo los primeros capítulos) a las tablas más útiles de la especialidad junto con fórmulas e información general útil para el oficio. En cualquiera de los dos casos podemos encontrar tabulados cálculos como x^2 , x^3 , \sqrt{x} , $\sqrt[3]{x}$, $\pi \cdot x$ y $1000/x$, junto a las tablas de logaritmos, de funciones trigonométricas y de muchas fórmulas frecuentes.



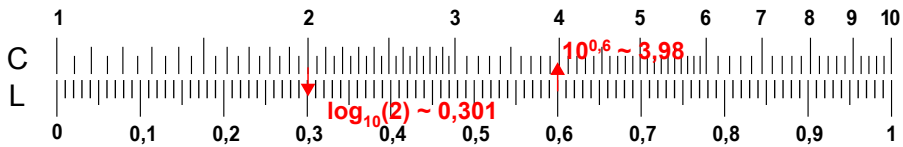
Fuente: elaboración propia

Figura 1.2. A la izquierda, dos manuales dedicados a tablas. A la derecha, manuales de ingeniería en los que las tablas constituyen un capítulo concreto. El manual abierto en primer plano incluye en la parte derecha una nota de corrección de errores añadida durante la encuadernación.

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.

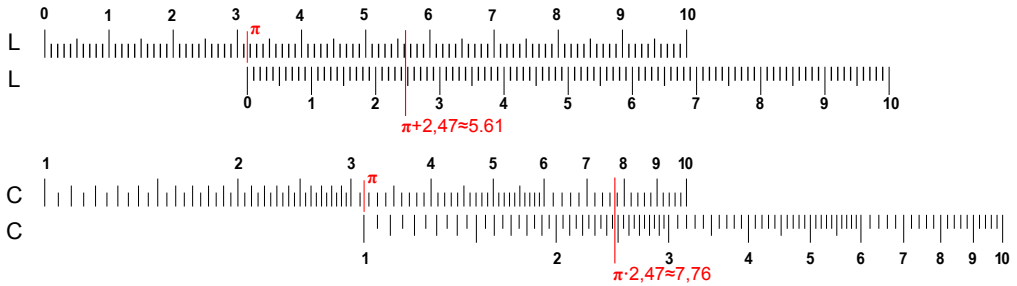
Una escala logarítmica es aquella en la que un valor x está representado en una posición proporcional a $\log(x)$. La Figura 1.3 muestra una escala logarítmica C del 1 al 10 enfrentada a una escala lineal L de 0 a 1, ambas de la misma longitud. Las dos, en paralelo, forman un ábaco gráfico que permite calcular a ojo las funciones $\log_{10}(x)$ y 10^x .



Fuente: elaboración propia

Figura 1.3. Dos escalas, C logarítmica y L lineal, enfrentadas. Una escala logarítmica como la C , que va del 1 al 10, se denomina “década”. Notemos que el 2 de la escala logarítmica enfrenta un punto en la escala lineal que aproximamos como 0,301. De ahí se obtiene que $\log_{10}(2) \approx 0,301$, pero aplicando las propiedades de los logaritmos, también que $\log_{10}(20) = \log_{10}(2 \cdot 10) = \log_{10}(2) + \log_{10}(10) \approx 1,301$, que $\log_{10}(2000) \approx 3,301$ o que $\log_{10}(0,02) \approx -2 + 0,301 = -1,699$. Igualmente, también se obtiene que $10^{0,6} \approx 3,98$, y con la aritmética de los exponentes, que $10^{1,6} \approx 39,8$ y que $10^{-0,4} \approx 0,398$.

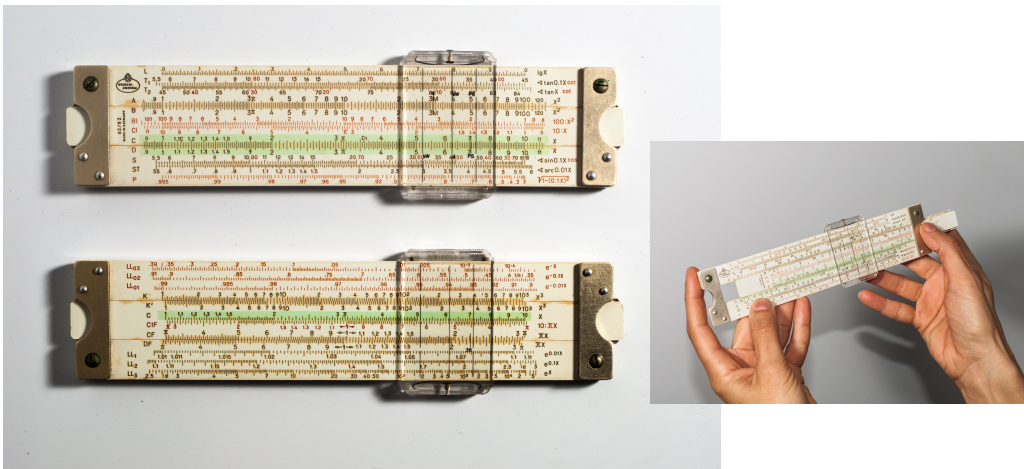
Disponer dos escalas lineales y desplazar una sobre la otra permite sumar y restar, pero la misma operación con escalas logarítmicas significa multiplicar o dividir (Figura 1.4). Desplazar escalas lineales para sumar y restar no parece una gran ventaja si lo comparamos con la aritmética de lápiz y papel, pero desplazar escalas logarítmicas para multiplicar o dividir sí es una ayuda valiosa.



Fuente: elaboración propia

Figura 1.4. Efecto del desplazamiento con escalas lineales y logarítmicas. Es importante destacar que la ubicación de los valores reales se hace a ojo, como es el caso de las constantes π y 2,47 y de los resultados. Los mismos desplazamientos también permiten restar $5,61 - \pi$ para obtener 2,47 o dividir $7,76 / 2,47 = \pi$

La regla de cálculo (Figura 1.5) es un instrumento mecánico que permite hacer operaciones aproximadas con valores reales. En ella encontraremos tres piezas: una fija, formada por dos reglas paralelas, entre las que se desliza la segunda pieza, una regla móvil y, finalmente, un cursor, también móvil, que se desplaza independientemente sobre las reglas para ayudar a alinear los valores de las escalas. Cada regla contiene una o más escalas logarítmicas, acompañadas por escalas de otro tipo que amplían la funcionalidad del aparato.



Fuente: elaboración propia

Figura 1.5. Regla de cálculo de los años 70. A la izquierda se muestran los dos lados. Notad los rótulos de algunas escalas: “A” y “B” para escalas logarítmicas de dos décadas, “C” y “D” para escalas de una década; con la letra “I” aparecen versiones invertidas de las mismas escalas, con “F” versiones donde la década comienza por π . Con otras letras se rotulan ayudas para funciones exponenciales y trigonométricas. La escala “L” es la única lineal que aparece aquí.

Este diseño es el producto de la evolución de una serie de inventos del siglo xvii, con la incorporación de nuevos materiales (policarbonato) y el añadido de escalas adicionales. Desde los inicios del siglo xx, las escalas habituales han ido rotuladas de forma estándar con letras (A, B, C...).

Para operar con la regla hay que normalizar cada operando, mentalmente o por escrito, separando las mantisas y los exponentes y llevando cuentas separadas. Las escalas también hacen papel de ábaco gráfico para algunos cálculos. Desde el punto de vista práctico, la regla de cálculo tiene algunos inconvenientes: está orientada a la notación en coma flotante y solo trata las mantisas, lo que obliga al usuario a gestionar, por escrito o mentalmente, el exponente del resultado. Además, no da buen soporte a la suma y tiene la precisión muy limitada, normalmente por debajo de las cuatro cifras significativas.

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.

Los humanos estamos habituados a trabajar con más de veinte símbolos (dependiendo del alfabeto) para representar textos y con diez dígitos (del 0 al 9) para los números. Los inventores de los dispositivos digitales tuvieron que decidir cómo representar textos, números y cualquier tipo de dato en forma de bits, a la vez que inventaron las técnicas de procesamiento de los datos así representados.

Dentro de los dispositivos digitales tenemos circuitos. Unos sirven para representar datos y otros para transformarlos. Los circuitos dedicados a la representación de los datos, llamados registros, tienen capacidad para un número de dígitos binarios determinado durante el diseño del dispositivo. Para codificar valores numéricos se han utilizado sobre todo dos técnicas, que para simplificar llamaremos “binario” y “BCD” (que viene a ser algo así como “decimal-en-binario”), sin contar con la complicación añadida por los signos positivo y negativo. Como aprenden los alumnos de ingeniería informática, cada una de ellas supone un conjunto de valores representables y unos circuitos de cálculo.

La codificación binaria (Figura 1.6) consiste en trabajar en base 2, es decir, a entender que una posición numérica vale el doble que su lado derecho y la mitad que el izquierdo. En comparación con la representación decimal habitual, solo se ha cambiado el factor 10 que relaciona dos posiciones consecutivas por el factor 2. Ahora, en lugar de

unidades, decenas o centenas, podríamos hablar de parejas, “cuatrenas” u “ochenas”. La conversión entre esta codificación y la humana es trabajosa y exige hacer divisiones por la base (2 o 10) y tomar los residuos.

La codificación BCD consiste en hacer paquetes de cuatro cifras binarias llamados nibbles. Un nibble permite representar un dígito decimal. La conversión entre este formato y el humano es sencilla y solo exige una tabla de diez entradas.

	Coma fija	Coma flotante
Decimal	250	$2,5 \times 10^2$
Interpretación:	$2 \times 10^2 + 5 \times 10^1$	$(2 \times 10^0 + 5 \times 10^{-1}) \times 10^2$
Binario	0000000011111010	111110100000111
Interpretación:	$1 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^1$	$1,111101 \times 2^7$ $(1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} + \dots) \times 2^7$
BCD	0000001001010000	0010010100000010
Interpretación:	$2 \times 10^2 + 5 \times 10^1$	$(2 \times 10^0 + 5 \times 10^{-1}) \times 10^2$

BCD	decimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9

Fuente: elaboración propia

Figura 1.6. Ejemplos de representación del 250 en coma fija y en coma flotante. Las representaciones digitales (binario y BCD) se hacen sobre un registro de 16 bits. En todos los casos, hay una base numérica b implícita que aparece en la interpretación en la forma bn. En notación decimal y en BCD, la base de numeración es b = 10, y solamente en binario la base es b = 2. Los dos formatos digitales de coma flotante que aparecen aquí son inventados e inútiles en la práctica, porque no pueden representar ni mantisas ni exponentes negativos. En los dos ejemplos, de los 16 bits se han dedicado los 12 primeros por la izquierda a la mantisa y los cuatro restantes al exponente, lo que permite representar mantisas de tres dígitos y exponentes de un dígito en BCD.

Los circuitos que transforman valores representados en registros son los operadores. Un operador aritmético es equivalente a un algoritmo, y para las operaciones habituales (suma, resta, multiplicación y división), los operadores se basan en los algoritmos humanos con alguna optimización. Por eso hay operadores distintos para cada operación, con variantes para BCD y para binario, para la coma fija y para la coma flotante.

Todas estas alternativas de codificación y operación se habían aplicado en los dispositivos de cálculo del siglo xx. Entre los pioneros, las máquinas de Torres Quevedo (1913) y el ENIAC (1945) funcionaban en decimal, mientras que los inventos de Zuse (a partir de 1936) lo hacían en binario. Entre los computadores de las primeras generaciones, tenemos el caso de IBM cuando presentó los primeros computadores fabricados en serie en el año 1954: el modelo 650 representaba y calculaba en decimal, y el 704 en binario. Este último podía operar en coma flotante.

Registros y operadores son piezas fundamentales de los computadores y, en general, de los dispositivos digitales. A lo largo del siglo xx, la tecnología de construcción de los circuitos se ha basado en dos tipos de componentes: las válvulas de vacío y los semiconductores. Los fabricantes de válvulas de vacío iban desarrollando diversos productos: triodos, pentodos, heptodos y otros, que permitían diseños progresivamente más fiables y económicos de los circuitos. En cuanto a los semiconductores, la fabricación en serie de los transistores comenzó a mediados de la década del 50 y los primeros circuitos integrados la década siguiente. El incremento del número de transistores por chip ha sido continuo desde entonces comenzando por unas pocas decenas y sobrepasando el millón en la década de los 90. Por lo tanto, un dispositivo electrónico que pesaba toneladas y ocupaba una habitación en el año 30 cabía en una caja de zapatos en el año 60 para acabar ocupando unas pocas micras cuadradas dentro de un chip del año 90. La evolución del consumo eléctrico también es notable, porque ha disminuido de kilovatios a microvatios y, en consecuencia, también ha bajado el calor disipado y ha hecho posible alimentarlo con una pila. Adicionalmente, la miniaturización ha permitido incrementar la velocidad de trabajo.

Los diseños varían bastante, pero podemos hacernos la idea de que, para construir un registro de 16 bits útil para hacer cálculos, habrá que combinar decenas de triodos o de transistores. También las alternativas de diseño de los operadores son muchas, pero para construir un sumador de coma fija para datos de 16 dígitos harían falta varias decenas de transistores. Hay que decir que los circuitos capaces de operar con coma flotante son relativamente costosos. Por ejemplo, hasta el año 1980, cuando Intel anunció el coprocesador 8087 (formado por 45.000 transistores), no se integraron en un único chip el conjunto de operadores que permitían hacer las operaciones elementales de suma, resta, multiplicación, división y raíz cuadrada.

LA CALCULADORA ELECTRÓNICA

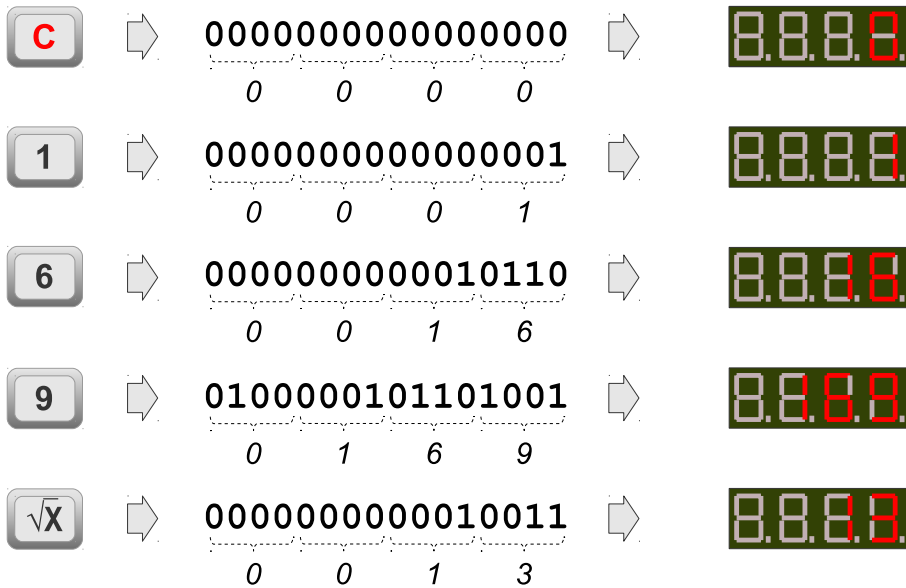
La historia de las calculadoras electrónicas, es decir, de los dispositivos electrónicos especializados en cálculos matemáticos de forma interactiva, comienza con los años 60. En aquel momento los primeros sistemas operativos interactivos (MULTICS, UNIX) todavía estaban por desarrollar, de manera que 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. Para calcular la tangente de un ángulo no era nada práctico tener que “cargar” un programa (es decir, transferir a la memoria del computador, mediante un lector de tarjetas, el programa que ha de leer el ángulo, calcular la tangente y escribir el resultado) junto con los datos (en este caso el ángulo) y obtener el listado correspondiente (la tangente del ángulo, claro) en la impresora.

La calculadora electrónica nació como máquina de propósito concreto que imitaba a los dispositivos mecánicos o electromecánicos que ya existían utilizando la tecnología electrónica emergente. Muchos de los primeros fabricantes de calculadoras electrónicas (Casio, Sharp, Olivetti, etc.) ya estaban establecidos como fabricantes de artículos de oficina, mientras que otros (Texas Instruments, Hewlett Packard) se iniciaban en el sector de los semiconductores y de la instrumentación electrónica. Precisamente, Texas Instruments, uno de los primeros fabricantes en serie de transistores, diodos emisores de luz (LED) y chips, comercializó también la primera calculadora de bolsillo. Es decir, que las industrias de la calculadora y del computador nacieron de forma independiente, aunque compartían tecnología y ciertos aspectos estructurales en sus diseños. Igual que los computadores, las calculadoras se basaron en un procesador con registros y operadores conectados a una memoria que contiene instrucciones y datos. Y también, entre las primeras realizaciones (máquinas de sobremesa para oficinas o para laboratorio), encontramos válvulas de vacío, transistores e, incluso, memorias de núcleos magnéticos. La presentación de datos utilizaba diversas tecnologías, como los teletipos, los tubos de neón o los tubos de rayos catódicos.

Las calculadoras representaban internamente los datos en código BCD para simplificar la comunicación con el usuario. Una calculadora disponía de unos pocos registros para los operandos, todos dimensionados para un número de cifras dado. Si, por ejemplo, la calculadora trabajaba con 12 cifras decimales, los registros tenían 48 bits de largo. Si trabajaba en coma flotante, en cada registro se dedicaban dos de los dígitos para el exponente. En todo caso, se reservaban uno o dos dígitos para la representación de los signos y otros detalles aritméticos. Los pocos operadores electrónicos disponibles hacían operaciones muy sencillas, como sumar dos cifras BCD, cambiar una cifra de lugar, multiplicar o dividir una cifra por dos y poco más. Por lo tanto, sumar el contenido de dos registros necesitaba un programa, el microprograma, que descomponía la suma en operaciones elementales. La multiplicación de dos operandos era mucho más compleja, porque se descomponía en la obtención de los productos parciales (calculados cifra a cifra) más la suma de estos para llegar al resultado final. Las primeras calculadoras solo trabajaban en coma fija porque los algoritmos de la coma flotante eran demasiado complicados y exigían demasiada memoria para los microprogramas.

El conjunto de microprogramas de la calculadora se almacenaba en una memoria permanente durante la fabricación. En los laboratorios de los fabricantes existía la figura del *microprogramador*, especialista que creaba el microprograma apropiado para cada operación y que se preocupaba tanto por su corrección como por el aprovechamiento de la poca memoria disponible. La funcionalidad de una calculadora, a diferencia de un computador, viene de fábrica: si un modelo dado no calcula la raíz cuadrada el usuario no podrá cambiar el comportamiento posteriormente. Por otra parte, variando el surtido de microprogramas contenido en la memoria, los fabricantes creaban una gama de calculadoras parecidas pero especializadas, que se vendían como básicas, científicas, financieras, etc.

Estructuralmente, la calculadora está formada por un teclado, una memoria que contiene tantos microprogramas como teclas (o combinaciones de teclas) y un sistema de presentación de datos. El comportamiento general es cíclico: el dispositivo espera que a que el usuario pulse una tecla y enseguida activa el microprograma asociado a la tecla. El microprograma opera con el contenido de los registros, hace la presentación de datos y vuelve a comenzar el ciclo (Figura 1.7).



Fuente: elaboración propia

Figura 1.7. Ejemplo de funcionamiento de una calculadora electrónica de cuatro dígitos. Cada tecla provoca cambios en el registro de operando y actualiza el valor mostrado. La tecla C (de Clear) activa un microprograma que hace cero todos los bits del registro de operando. Cada tecla numérica activa un microprograma que desplaza el contenido del registro un dígito (es decir, 4 bits) hacia la izquierda y asigna un valor constante (la cifra rotulada en la tecla) en el dígito de la derecha. Finalmente, la tecla de función (la raíz cuadrada) activa el microprograma que hace el cálculo correspondiente. En todos los casos, los microprogramas vuelcan el contenido del registro en la electrónica de presentación de datos, que determina qué segmentos del visualizador se encienden. Para simplificar, este ejemplo ignora la gestión de la coma y supone una presentación de datos sin decimales.

La aparición de los circuitos integrados hizo posible las calculadoras de bolsillo. En cuanto a la presentación de datos, la producción en serie de los diodos LED que emitiesen luz visible (inicialmente solo roja, con el tiempo vendrían otros colores) comenzó a finales de los años 60. En los años 70 se pudo concentrar toda la lógica de una calculadora en dos o tres chips, junto con algunos componentes discretos (diodos, resistencias y condensadores) que se encargaban de la alimentación. En 1972 Casio presentó la calculadora Casio Mini, de las dimensiones de una tarjeta de crédito.

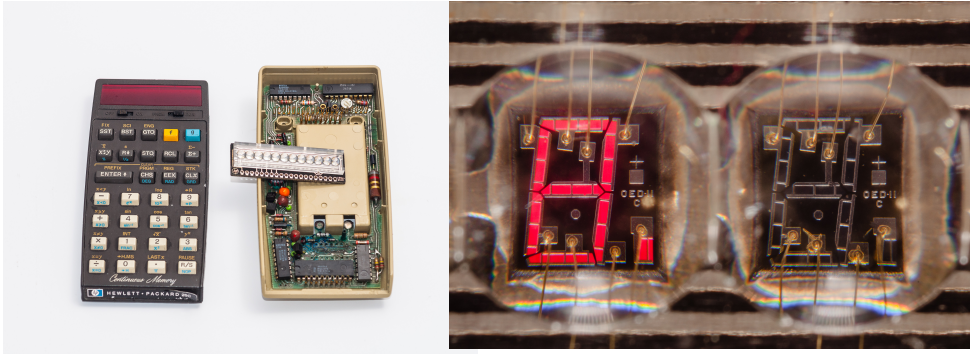
En 1971, Intel comenzó la producción del primer microprocesador, el 4004, por encargo de Busicom, un fabricante japonés de calculadoras de sobremesa. Los microprocesadores eran versiones de poca potencia de los computadores de propósito general integradas en un chip. A partir de entonces, las calculadoras de sobremesa entraron en declive porque el computador personal, basado en microprocesadores, hacía el trabajo de la calculadora con el software apropiado. Pero la calculadora de bolsillo seguía basándose en procesadores diseñados a posta, más sencillos.

LA CALCULADORA CIENTÍFICA DE BOLSILLO

Cuando la cantidad de memoria lo permitió, se hicieron posibles los microprogramas que trabajaban en coma flotante, y los fabricantes etiquetaron el producto resultante como calculadora científica. Pero eso no eliminaba la necesidad de las tablas para calcular funciones trigonométricas o exponenciales.

La obtención de las funciones trascendentes mediante polinomios con un tiempo de cálculo razonable exige operadores de coma flotante imposibles de insertar dentro de una calculadora de los años 70. Sin embargo, había un precedente del mismo problema en el ámbito militar: los sistemas de navegación de los bombarderos de los años 50, cuando los operadores todavía se hacían con válvulas de vacío, necesitaban calcular funciones trigonométricas y los ingenieros desarrollaron un método llamado CORDIC, que resolvía el problema de forma iterativa con operadores de coma fija y una tabla de constantes en ROM. Los ingenieros de Hewlett Packard adaptaron la solución a la representación BCD de coma flotante y la ampliaron para abarcar el surtido de funciones matemáticas habituales. Con esta técnica, el cálculo de una función cualquiera se descompone en sumas, restas y desplazamientos de los dígitos, es decir, en las operaciones elementales disponibles en una calculadora.

Las primeras calculadoras científicas, orientadas al uso en un laboratorio, fueron aparatos de sobremesa. Cuando los chips integraron unos pocos miles de transistores, se hizo posible la primera calculadora científica portátil, el modelo HP-36, que se produjo a partir de 1972. Más adelante, aprovechando la memoria RAM disponible, estas calculadoras se hicieron programables. Los programas eran listas de órdenes de cálculo combinadas con algunas órdenes de control de flujo. Y las calculadoras de alta gama incorporaron un periférico de memoria secundaria para almacenar los programas: la tarjeta magnética. Los modelos posteriores al HP-36, ya programables, tuvieron mucho éxito, y la publicidad indicó que la misión astronáutica Apollo-Soiuz iba equipada con un computador para la navegación espacial y una calculadora HP-65 programable por si acaso fallaba el computador.



Fuente: elaboración propia

Figura 1.8. Calculadora de bolsillo HP-25C fabricada entre los años 1976 y 1978. Se trata de una calculadora científica programable comercializada poco después de la HP-36. Dentro de la calculadora (izquierda) encontramos cinco chips, que se reparten la lógica de cálculo, la memoria, el control del teclado y la electrónica de los visualizadores. Cada uno de los 12 dígitos (derecha) estaba formado por un conjunto de diodos LED alimentados selectivamente a través de los electrodos.

Las calculadoras científicas portátiles impactaron entre los profesionales y los estudiantes. La denominación publicitaria era “Electronic Slide Rule”. El artículo que apareció en el *Hewlett-Packard Journal* sobre la HP-36 medía el tiempo necesario para realizar tres cálculos propuestos con dos herramientas distintas: con una regla de cálculo y con la calculadora. El nombre de las primeras calculadoras científicas de Texas Instruments (SR-50, SR-52, etc.) comenzaba por SR, de *Slide Rule*.

Los fabricantes norteamericanos Hewlett Packard y Texas Instruments lideraron inicialmente el nuevo mercado y el diferente estilo de funcionamiento de sus productos produjo una divertida controversia entre los usuarios, precedente de la de Windows contra Mac.

CONCLUSIÓN

Igual que ha hecho con las telecomunicaciones, la electrónica ha transformado las herramientas de cálculo personal. Como resultado, los aparatos mecánicos que existían fueron sustituidos durante el siglo xx por máquinas electrónicas y han acabado simulados en aplicaciones dentro de los computadores y de los dispositivos portátiles. Las máquinas clásicas han desaparecido o están a punto de hacerlo. Pero los nostálgicos mantienen su recuerdo en multitud de páginas web entusiastas.

BIBLIOGRAFÍA

COCHRAN, D. S. (1972), “*Algorithms and Accuracy in the HP-35*”, Hewlett-Packard Journal, vol 23, nº. 10.

MONNIER R. E.; OSBORNE, T. E.; Cochran, D. S., (1982), *The HP Model 9100A Computing Calculator* en SIEWIOREK, D. P; BELL, C.; GORDON, A., *Computer structures: Readings and examples*, McGraw-Hill.

WILLIAMS, M. R. (1997), *A History of Computing Technology*, IEEE Computer Society Press.

The Oughtred Society, <http://www.oughred.org>

Xavier Molero Prieto
Universitat Politècnica de València

Resumen: 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.

INTRODUCCIÓN

La invención del ordenador representa uno de los hitos fundamentales de la humanidad, quizás tan importante como la aparición de la rueda o la escritura. Como suele acontecer en el caso de inventos tan complejos, su advenimiento no debemos buscarlo románticamente en la idea feliz de una mente genial y solitaria sino, al contrario, en el proceso colaborativo de una comunidad de personas que han seguido distintos caminos buscando inspiración en multitud de fuentes. Una inspiración espoleada, por un lado, por la necesidad de efectuar cálculos sin cometer errores y, por otro, responder a una sociedad en vías de desarrollo que necesitaba tratar de manera automática cantidades muy grandes de información.

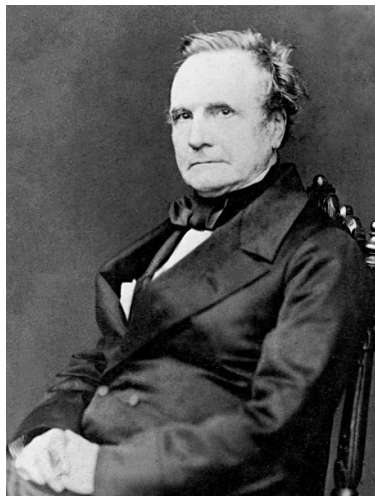
En el campo de la informática, además, se da uno de los fenómenos más sorprendentes y menos conocidos: la mayor parte de los ordenadores diseñados y construidos desde finales de la década de los años cuarenta del siglo XX responden a un mismo principio elemental. Un ordenador es, en su más pura esencia, una máquina automática con una unidad de memoria y una unidad de control que procesa información por medio de las instrucciones de un algoritmo almacenadas en la memoria. Este principio, conocido con el nombre de “programa almacenado”, es el que distingue a un ordenador de una simple calculadora. Su concepción y síntesis se produjo, como veremos, durante la construcción de los primeros ordenadores con tecnología electrónica.

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.

La máquina incorporaba todos los elementos que podemos encontrar hoy en día en un ordenador: una memoria (*store*, almacén) para almacenar hasta unos mil números de cinco cifras, una unidad de procesamiento (*mill*, molino) encargada de hacer las operaciones aritméticas, una unidad de control (*control*) que aseguraba la secuenciación correcta de las operaciones, y finalmente, dispositivos de entrada (tarjetas perforadas para datos e instrucciones) y de salida (tarjetas perforadas, dispositivos de impresión y una campana para llamar la atención de la persona que la estuviese utilizando).



Fuente: Wikipedia

Figura 2.1. Charles Babbage hacia 1860

Sin embargo, desde un punto de vista arquitectónico, la estructura de la Máquina Analítica es equivalente a la de las grandes máquinas de calcular que precedieron la invención del ordenador moderno, en las que los programas se iban introduciendo desde el exterior. Lamentablemente, las dificultades tecnológicas —engranajes, vapor como fuerza motora— y financieras, amén de otros factores como el perfeccionismo desmesurado de Babbage y los continuos retoques del diseño, impidieron que la máquina viera la luz.

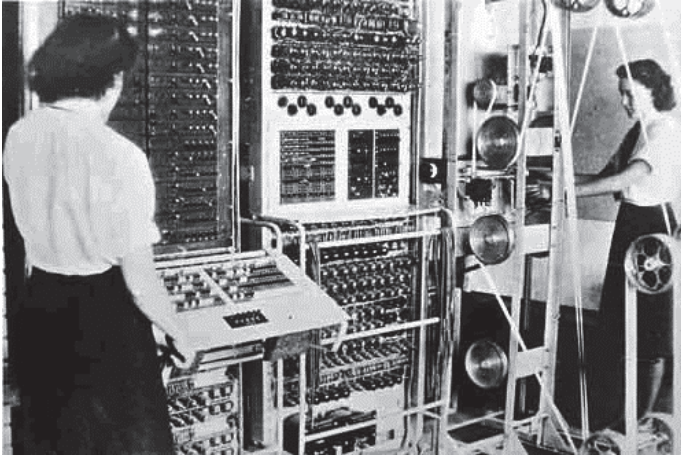
No debemos olvidar, por otro lado, que Ada Byron, condesa de Lovelace, gran amiga y colaboradora de Babbage, conoció con detalle el funcionamiento de la Máquina Analítica y diseñó el primer programa publicado (en 1843) de la historia. Este programa calculaba la sucesión conocida como números de Bernoulli. A pesar de no llegar a construirse, la máquina permitió que Ada lucubrara sobre sus posibilidades y llegara a algunas conclusiones dignas de una gran visionaria. Así, imaginó que el dispositivo, más allá del procesamiento aritmético de números, podría manipular cualquier tipo de información que se expresara mediante símbolos, como ocurre hoy en día con la música, la fotografía, el vídeo, etcétera. Además, dentro de lo que denominamos inteligencia artificial, apuntó que la máquina no podría pensar o razonar del mismo modo que lo hacemos las personas porque era incapaz de producir nada nuevo más allá de aquello para lo que estuviera programada. Desgraciadamente, su muerte prematura impidió que esta ciencia informática en estado embrionario pudiera desarrollar todas sus posibilidades.

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 Betchley Park, al noroeste de Londres, participaron matemáticos de la talla de Alan Turing i Max Newman. Sabemos poco de estas máquinas, denominadas Co-

lossus y construidas bajo la dirección del ingeniero Thomas H. Flowers a partir de 1943, por el secretismo militar ligado a todas estas tareas de espionaje y porque, finalizada la guerra, se dio orden de destruirlas junto con toda la información relevante.

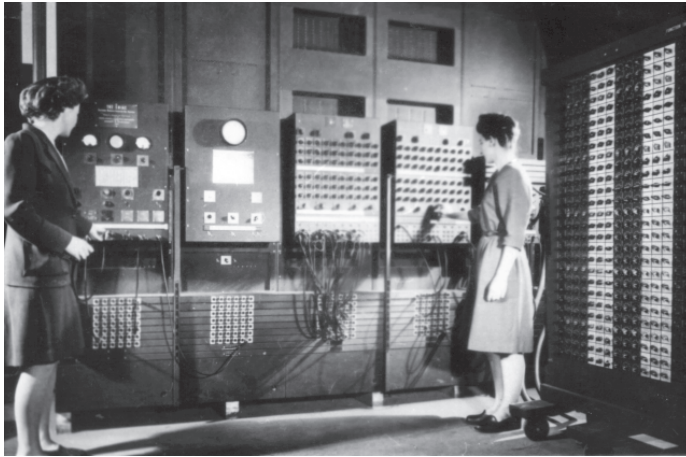


Fuente: Wikipedia

Figura 2.2. Una máquina Colossus manejada por Dorothy Du Boisson (izquierda) y Elsie Booker (derecha)

Por su parte, la creencia en su superioridad tecnológica llevó a los alemanes a dejar de lado las investigaciones en el desarrollo de calculadoras potentes. Así, el trabajo del ingeniero Konrad Zuse, que desde 1936 diseñó varias calculadoras programables usando relés electromecánicos, careció del apoyo necesario por parte de las autoridades militares germanas. Zuse, sin conocer la obra de Babbage y motivado como aquél por las necesidades del cálculo civil, construyó el Z3, una de las primeras calculadoras universales binarias controladas por programa, operativa en una fecha tan temprana como 1941, poco después de comenzar el conflicto bélico.

Fue, sin embargo, en Estados Unidos, donde el esfuerzo bélico llevó a la construcción del ENIAC (*Electronic Numerical Integrator And Computer*), una impresionante máquina electrónica diseñada para calcular trayectorias balísticas y en torno a la cual se establecieron las bases de lo que hoy en día entendemos por un ordenador. La construcción, financiada por el ejército norteamericano pero llevada a cabo en el ámbito académico, fue llevada a cabo en secreto por un equipo dirigido por John Mauchly, físico responsable del diseño conceptual, y John P. Eckert, ingeniero de una gran capacidad responsable del diseño de los circuitos electrónicos.



Fuente: Wikipedia. Ejército de los Estados Unidos

Figura 2.3. Las programadoras Betty Jean Jennings (izquierda) y Fran Bilas (derecha) junto al panel de control principal del ENIAC

El proyecto se llevó a término en muy poco tiempo porque, a pesar de conocer a priori las limitaciones de la máquina (poca memoria, difícil programación), el ejército la necesitaba para proveer de tablas de tiro las nuevas armas del conflicto bélico. Los ingenieros que participaron en la construcción del computador tenían experiencia previa en la tecnología de válvulas de vacío aplicadas al ámbito del radar y consideraron la fiabilidad un criterio clave de diseño. Prespert empleó circuitos sencillos y funcionales, evitando otros más elegantes pero menos fiables, y usar dispositivos electrónicos de bajo coste pero sometidos a unas condiciones de trabajo muy por debajo de sus umbrales máximos de funcionamiento. Además, las válvulas de vacío se organizaron en paneles fácilmente desmontables en caso de fallo.

Sobre el ENIAC se cuentan muchas anécdotas cuya veracidad está lejos de la realidad. Por ejemplo, se dice que, cuando la máquina se ponía en funcionamiento, los 147 kW de potencia que necesitaba provocaba el parpadeo del barrio oeste de la ciudad de Filadelfia; un mito todavía hoy difícil de desterrar. Por otro lado, y a pesar de la creencia contraria, las aproximadamente dieciocho mil válvulas de vacío que lo componían eran menos sensibles al calor que sus setenta mil resistencias (la máquina pesaba alrededor de 30 toneladas). Aunque se suele afirmar que los fallos en las válvulas provocaban a menudo la parada de la máquina, en realidad, la lectora y la perforadora de tarjetas que IBM proveyó para la entrada y salida de datos dieron más problemas que las válvulas. Es cierto que estas últimas fallaban, sobre todo durante el proceso de apagado y encendido, pero también se desarrollaron métodos sistemáticos para detectar y sustituir los paneles donde se producían estos fallos. Los cálculos se solían hacer dos veces para comprobar que se obtenía el mismo resultado y, de forma periódica, se la sometía a una serie de programas de test cuya respuesta era conocida de antemano.

El ENIAC disponía de veinte acumuladores capaces de almacenar, cada uno, un número decimal de diez dígitos; estos acumuladores, que incorporaban la capacidad de sumar y restar, podían combinarse para operar con números de mayor precisión. Otras unidades específicas se encargaban de operaciones más complejas (multiplicación, división, raíz cuadrada), así como del control y secuenciación de las operaciones. La máquina, con una arquitectura distribuida, se programaba interconectando sus unidades entre sí de una forma determinada. En realidad, los programas del ENIAC consistían en diagramas de conexiones y posiciones de conmutadores, y su diseño podía suponer el trabajo de varias semanas; pasar del diagrama de conexiones al cableado físico de la máquina costaba uno o dos días. Consecuentemente, el tiempo de ejecución de los programas era, comparado con los anteriores, muy pequeño. A pesar de ello, las conexiones de los programas específicos de tablas balísticas podían modificarse fácilmente para introducir cambios en los parámetros.

La tecnología electrónica permitió hacer 5.000 sumas por segundo, una velocidad impresionante comparada, por ejemplo, con las dos sumas por segundo que podía conseguir el Harvard Mark 1 de Howard Aiken, una inmensa calculadora construida en 1944 con relés electromecánicos y basada en las ideas de Charles Babbage y su Máquina Analítica.

La construcción del ENIAC planteó de forma implícita, ya desde su inicio, una clara división de género: el diseño del hardware se consideró una tarea masculina, mientras que la programación, una disciplina inexistente en aquel momento, fue equiparada a los trabajos administrativos o de oficina y, consecuentemente, atribuido a la competencia del sexo femenino. En aquella época se creía que las mujeres estaban mejor dotadas que los hombres para hacer cálculos matemáticos y, de hecho, la palabra *computer*, entonces, se usaba para referirse a una persona con buena capacidad de cálculo (hoy utilizaríamos el término calculista). Tanto es así que el nombre inicial propuesto fue simplemente ENI (*Electronic Numerical Integrator*); en última instancia se le añadió, por sugerencia de un coronel del ejército formado en el MIT, la coletilla *And Computer*, con lo que el acrónimo final, ENIAC, resultó más pegadizo y fácil de recordar.

Finalmente, fue un equipo de seis mujeres graduadas en matemáticas y lógica, las *ENIAC girls*, que ya trabajaban como *computers* para el ejército, quienes lidiaron con la programación a partir únicamente de los esquemas de conexión y las aclaraciones proporcionadas por los propios ingenieros. Una tarea intelectual, por otra parte, de una dificultad verdaderamente destacable, a la que hubo que añadir el esfuerzo físico derivado de manipular cerca de tres mil conmutadores y un gran número de cables. Aunque nos pueda sorprender hoy en día, desde sus inicios y hasta bien entrada la década de los años 60, la programación de ordenadores fue considerada una tarea eminentemente femenina.

EL CONCEPTO DE PROGRAMA ALMACENADO

A pesar de que el diseño de ordenadores se ha visto afectado en el tiempo por la incorporación paulatina de diferentes mejoras tecnológicas (por ejemplo, el transistor y los circuitos integrados), el principio básico de funcionamiento quedó definido, precisamente, durante la construcción del ENIAC. Sus diseñadores se apercibieron en seguida de las limitaciones en cuanto a la capacidad de memoria y la versatilidad de su programación, pero dada la urgencia del ejército, la máquina se construyó según los planes previstos inicialmente. En efecto, ya en enero de 1944, mientras el ENIAC se construía, Eckert y Mauchly empezaron a valorar la idea de grabar en la memoria tanto las instrucciones como los datos del programa. Esta nueva concepción exigía el diseño de una segunda máquina totalmente nueva denominada EDVAC (*Electronic Discrete Variable Automatic Computer*), una tarea que comenzaron a desarrollar al mismo tiempo que el ENIAC se materializaba en los sótanos de la Moore School de Filadelfia. Esta máquina, implementada con la misma tecnología que el ENIAC pero con una estructura interna concebida de un modo radicalmente nuevo, abriría el camino a ordenadores con un rendimiento hasta entonces inimaginable.



Fuente: ENIAC Computer History Archives

Figura 2.4. John Eckert (izquierda) y John Mauchly (derecha) en la presentación del ENIAC al público estadounidense

Fue algunos meses más tarde, en septiembre de 1944, cuando John von Neumann (1903-1957), un científico de enorme prestigio involucrado en el proyecto Manhattan, se incorporó como colaborador en el equipo del ENIAC. El proyecto Manhattan intentaba entonces construir las bombas atómicas que se arrojarían sobre la población civil

japonesa en agosto de 1945 y von Neumann necesitaba una máquina capaz de resolver los complicados modelos matemáticos que usaban en su diseño. En muy poco tiempo, von Neumann, dotado de una capacidad matemática excepcional, fue capaz de sintetizar de una forma asombrosamente clara y lógica el principio de programa almacenado en el hoy clásico borrador *First draft of a report on the EDVAC*. Este informe de un centenar de páginas fechado el 30 de junio de 1945 fue distribuido con la única firma de John von Neumann y en él no se menciona en ningún momento a Eckert ni Mauchly, ni tampoco a ningún miembro del equipo de trabajo que, con toda seguridad, debatió ampliamente gran parte de las cuestiones tratadas en el informe.

Con el tiempo, la difusión del borrador contribuyó a atribuir a von Neumann, una persona eminente en el ámbito académico por sus extraordinarias aportaciones en física y matemáticas, la paternidad exclusiva del concepto de programa almacenado, mientras que Eckert y Mauchly, profesores de una universidad de segunda fila y probables inventores de la idea, fueron relegados al olvido. Es plausible que, enfrascados como estaban en el diseño de los circuitos electrónicos, no hubieran descrito y definido la arquitectura del nuevo ordenador EDVAC como lo hizo von Neumann, valorando su esencia y concepción más abstracta, pero esto no justifica el trato que históricamente se les dio. Sea como fuere, el efecto que provocó la publicación así del informe fue una célebre enemistad duradera y recíproca, iniciada en 1947, entre von Neumann, por un lado, y Eckert y Mauchly por otro.



Fuente: Wikipedia

Figura 2.5. John von Neumann

El propio Mauchly, para defender su papel en la creación del nuevo tipo de ordenadores, publicó en enero de 1947 el artículo *Preparation of problems for EDVAC-type machines*. En este trabajo describe cómo abordar y resolver problemas con estas má-

quinas, haciendo hincapié en la necesidad de una gran memoria, de un conjunto básico de instrucciones elementales y en la idea de grabar tanto las instrucciones como los datos en esta memoria.

De alguna manera, ocurrió algo parecido a lo que sucedió con Charles Babbage y Ada Byron. Mientras el primero, queriendo construir una máquina fiable, se centró en los aspectos más tecnológicos ligados al cálculo puramente aritmético, Ada pudo alejarse para situarse en una perspectiva más amplia e imaginar, en abstracto, a la Máquina Analítica manipulando números que, en realidad, podrían ser símbolos de otro tipo de información como música o poesía.

LOS PRIMEROS ORDENADORES MODERNOS

Del borrador firmado por von Neumann se hicieron aproximadamente poco más de una decena de copias. Este informe, que mostraba los principios básicos de un ordenador particularizados en los planos de construcción del EDVAC, fue declarado judicialmente de dominio público (aunque reservado), evitándose así que se patentara la idea y fuese explotada, como querían Eckert y Mauchly, en el ámbito comercial. Entre 1945 y 1951 la difusión de los principios básicos del ordenador moderno dará lugar a la construcción varias máquinas repartidas entre Estados Unidos y Gran Bretaña.

En realidad, la clave de la aparición del tipo de ordenador que venimos refiriendo como moderno, viene de la mano de dos factores principales. Por un lado, era necesario que hubiese un ente dedicado a coordinar y dirigir las operaciones del resto de unidades, evitando así la intervención directa del operador. Y por otro, los algoritmos que resolvían los problemas se tenían que traducir en programas y datos y grabar en la memoria de la máquina. Así, los planos del EDVAC preveían que tanto las instrucciones como los datos se expresasen mediante un código numérico (conjunto de bits); era responsabilidad de la unidad de control saber en cada momento cómo interpretar de forma correcta cada conjunto de bits y discernir si se trata de una instrucción o de un dato. El ENIAC, apenas se construyó, estaba totalmente superado, aunque fue utilizado y mejorado varias veces hasta 1955. En definitiva, la nueva máquina, al contrario que sus antecesores, *procesaba cualquier tipo de información binaria*, lo cual incluía, entre otras muchas posibilidades, efectuar los cálculos aritméticos demandados hasta ese momento a las calculadoras y otros dispositivos de ayuda al cálculo.

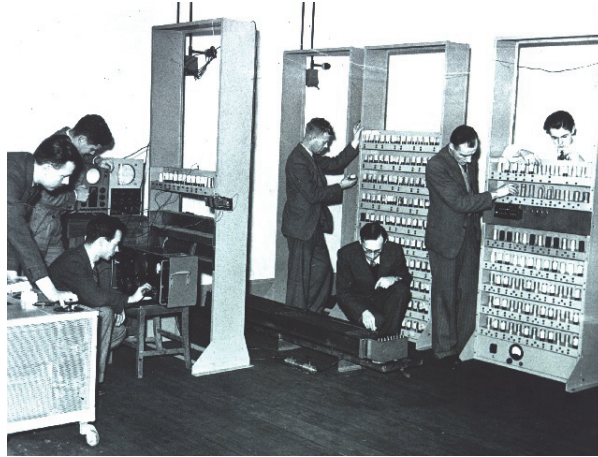
Los primeros diseños efectivos de computadores usando el principio de programa almacenado descrito en el informe se llevaron a cabo en Gran Bretaña. Los dirigentes de este país, tras la guerra, quisieron contar rápidamente con un programa militar nuclear y, para ello, era necesario invertir en tecnología para el desarrollo de grandes calculadoras. En aquel momento en que nacía la Guerra Fría, desde el punto de vista estadounidense, Gran Bretaña era un aliado seguro, por lo que no tuvo inconveniente en compartir información con los investigadores ingleses, matemáticos la mayor parte. Entre otros muchos, fueron los profesores Frederic Williams y Tom Kilburn en la universi-

dad de Manchester, y Maurice Wilkes en la universidad de Cambridge, quienes se pusieron manos a la obra y construyeron los primeros ordenadores modernos.

Williams y Kilburn construyeron el Manchester SSEM (*Small-Scale Experimental Machine*), apodado cariñosamente Baby, que ejecutó su primer programa en junio de 1948. En realidad, se trataba de un prototipo para experimentar con tubos de rayos catódicos (CRT, *Cathode Ray Tube*) como dispositivo de memoria con una capacidad total de 32 posiciones de 32 bits. Esta manera de implementar la memoria permitía ver los bits con los propios ojos, ya que se conservaban en forma de puntos visibles en las pantallas de rayos catódicos. Concebido como la expresión mínima de un computador, solamente podía hacer sumas y negaciones; el resto de operaciones las hacía en base a combinaciones de estas dos.

Este prototipo sirvió de base para diseñar el Manchester Mark 1, operativo en abril de 1949, un computador más completo cuya fiabilidad se demostró al ejecutar durante más de nueve horas y sin errores un algoritmo diseñado por Max Newman para calcular los números primos de Mersenne. La prensa británica usó las palabras “cerebro electrónico” para describir la máquina a sus lectores, algo similar a lo que hizo la prensa estadounidense cuando en febrero de 1946 el Departamento de Guerra desveló a la opinión pública la existencia del ENIAC. A su vez, este ordenador dio pie a la construcción del Ferranti Mark 1, considerado el primer ordenador comercial de la historia (febrero de 1951), adelantándose por muy poco al ordenador UNIVAC I (marzo de 1951), construido por Eckert y Mauchly para la Oficina del Censo de los Estados Unidos.

Maurice Wilkes y su equipo construyeron el EDSAC (*Electronic Delay Storage Automatic Calculator*), operativo en junio de 1949 y no concebido como un prototipo, sino como un ordenador completo dispuesto para su explotación. Se cuenta que a Wilkes le facilitaron en Cambridge el borrador escrito por von Neumann, pero hubo de leerlo en una sola noche porque debía devolver el original al día siguiente y en esa época no había fotocopiadoras. Wilkes participó más tarde en las conocidas coloquialmente como *Moore School Lectures*, el primer curso sobre construcción de ordenadores impartido en el verano de 1946 y auspiciado por el Pentágono. En el curso, denominado *The theory and techniques for design of digital computers*, participaron, entre otros, von Neumann, Eckert y Mauchly, y sirvió para diseminar los conceptos relacionados con el diseño del EDVAC, precisamente en construcción en aquellos momentos.



Fuente: Wikipedia, CC 2.0, Computer Laboratory University of Cambridge
Figura 2.6. Construcción del EDSAC. En el centro, Maurice Wilkes agachado junto a la memoria construida con líneas de retardo

Como todos los ordenadores en aquel tiempo, el EDSAC estaba construido con tubos de vacío. Utilizaba el sistema binario e incorporaba una memoria construida mediante las denominadas líneas de retardo (*delay lines*), inventadas por Eckert, una especie de largos cilindros rellenos de mercurio donde los bits eran almacenados en forma de pulsos acústicos. La capacidad inicial fue de 512 posiciones de 18 bits. Se empleaba una cinta de papel perforada para introducir los programas y los datos en la memoria, y había un teletipo para escribir los resultados. Para demostrar su funcionamiento el computador calculó una tabla de cuadrados y otra de números primos. En aquel momento histórico en que estaba casi todo por descubrir, la escritura correcta de programas se desveló una tarea harto complicada; el propio Wilkes recuerda que dedicó muchísimo tiempo a pasearse entre su despacho y la sala donde se encontraba el EDSAC para resolver todo tipo de errores. También fue necesario diseñar una especie de sistema operativo primitivo, que denominaron *initial orders*, para facilitar tanto el diseño como la introducción y ejecución de los programas en la máquina.

En Estados Unidos, después de la publicación y difusión del borrador de von Neumann y con la abrupta marcha de Eckert y Mauchly de la universidad para fundar su propia compañía de ordenadores, no quedó en la More School ninguno de los diseñadores del EDVAC para acabarlo de construir. De hecho, este ordenador no estaría operativo hasta 1951, y se mantuvo en servicio hasta 1962.

Como decimos, Eckert y Mauchly siempre intuyeron el enorme campo de aplicación de los ordenadores y se propusieron hacer negocio comercializando un ordenador comercial, el UNIVAC I. Como paso previo y debido también a necesidades económicas, construyeron el BINAC (*Binary Automatic Calculator*), una máquina varias veces más rápida que la inglesa EDSAC y bastante más pequeña que el ENIAC, cuyo objetivo era

controlar el vuelo de un misil que acababa de desarrollar la armada estadounidense. Por lo tanto, esta máquina debía ser muy fiable y efectiva, e incluso se planteó que pudiese operar dentro de un avión, una posibilidad absurda a todas luces dado el tamaño, la fragilidad y el consumo eléctrico desmesurado de estas primeras máquinas.

La memoria del BINAC se construyó usando líneas de retardo de mercurio y, para aumentar su fiabilidad, integraba en una especie de tándem dos unidades de procesamiento idénticas que efectuaban los mismos cálculos y la unicidad del resultado se comprobaba de forma constante. Fue la primera máquina en usar cintas magnéticas como almacenamiento secundario. Presentada en 1949, llegó a operar durante 44 horas sin ningún contratiempo, aunque las malas lenguas afirman que solamente funcionaba bien cuando hacía buen tiempo.

Jon Ander Gómez Adrián
Universitat Politècnica de València

Resumen: 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.

El propósito de nuestra aportación consiste en analizar la evolución de los lenguajes de programación a tenor de las necesidades de la industria y en algunos casos del mundo académico, como es el caso del Pascal, diseñado por el profesor Niklaus Wirth para enseñar a programar de manera estructurada a sus alumnos. Pero antes vamos a ver algunas definiciones, seguidas de una taxonomía de los lenguajes de programación y qué características definen cada paradigma de programación. Conocer previamente cómo pueden clasificarse los lenguajes de programación servirá de ayuda para entender mejor su evolución histórica.

¿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. En base al carácter formal de los lenguajes de programación aportamos otra definición complementaria a las anteriores: *conjunto de reglas, símbolos, operadores y palabras reservadas que utilizaremos para escribir programas de ordenador*. Los símbolos, los operadores y las palabras reservadas tienen un significado asociado, por ejemplo el operador '+' se utiliza para indicar al ordenador que realice la suma de dos valores. Por último, otra definición más completa es *notación, conjunto de reglas y definiciones que determinan tanto lo*

que puede escribirse en un programa, y que el procesador puede interpretar, como el resultado de la ejecución de dicho programa por el procesador.

Algoritmo y expresión (en el contexto de la programación de ordenadores) son dos conceptos muy relacionados que servirán de apoyo para entender mejor qué es un lenguaje de programación y su utilidad. De entre las posibles definiciones de algoritmo hemos seleccionado la de Donald E. Knuth (1968), pues nos parece la más apropiada: *secuencia finita de instrucciones, reglas o pasos que describen de manera precisa las operaciones que un ordenador ha de ejecutar para llevar a cabo una tarea en un tiempo finito*. Una expresión es una *combinación de variables, constantes, operadores y nombres de función que tiene la propiedad de ser evaluada*. La definición de algoritmo está muy relacionada con las definiciones de lenguaje de programación vistas más arriba. En esta definición se insiste en la finitud de la secuencia de instrucciones y de llevar a cabo una tarea. No tendría sentido escribir un programa cuya secuencia de instrucciones fuese infinita, ni serviría de nada un algoritmo que no finalizase su ejecución para darnos la solución a un problema. Otro aspecto señalado es describir de manera precisa las operaciones a realizar, es decir, sin ambigüedad, pues el procesador no puede “interpretar” qué quiere decir el programador. La definición de expresión también ayuda a ver que las instrucciones debemos expresarlas de manera precisa combinando distintos elementos y siguiendo un conjunto de reglas y una notación específica, de manera que el resultado obtenido tras evaluar una expresión esté bien definido una vez conocidos los valores de las variables y las constantes que aparezcan en la expresión, así como las funciones que se invoquen. El resultado siempre será el mismo para los mismos valores de entrada.

En contraposición a lo estricto que resulta un lenguaje de programación, las personas utilizamos en muchas ocasiones lo que se conoce como pseudocódigo para compartir esbozos de algoritmos en las primeras etapas del diseño de aplicaciones. El pseudocódigo también resulta de mucha utilidad para explicar algoritmos en libros o en clase. Se caracteriza porque permite incluir lenguaje natural, dado que nunca será compilado o interpretado para ser ejecutado.

Los lenguajes de programación nos proporcionan mecanismos de abstracción para poder definir estructuras de datos y las operaciones o transformaciones que pueden realizarse sobre dichos datos. También incluyen mecanismos para el control del flujo de los programas que nos permiten establecer, con cierto nivel de abstracción, qué bloques de operaciones se ejecutarán repetidamente, o qué secuencias de instrucciones se ejecutarán en distintos momentos de la ejecución del programa habiéndolas escrito una sola vez. Los programadores utilizan abstracciones para representar conceptos y definir los cálculos que se realizarán en base a dichos conceptos. Los conceptos se representan en base a una serie de elementos básicos como pueden ser los datos numéricos y las operaciones que sobre ellos se pueden realizar. Programar, por tanto, consiste en combinar adecuadamente los elementos básicos que forman parte del lenguaje de programación para crear estructuras y cálculos más complejos, y más abstractos por

tanto, para poder obtener soluciones a problemas de cierta dificultad. Utilizando un símil de otro contexto, un escultor crea una obra a partir de ciertos materiales y utilizando sus herramientas; dependiendo de sus habilidades el resultado de su trabajo será más o menos valorado. Podemos decir que un lenguaje de programación proporciona las herramientas básicas para crear programas y, dependiendo del programador, el programa o aplicación resultante será de mejor o peor calidad.

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. Las librerías incluyen cada vez más funcionalidad, a beneficio de los programadores que ya no necesitan desarrollar ciertas funciones hoy consideradas como básicas. Durante los años 80 y 90 del siglo XX era habitual comprar ciertas librerías de manera adicional al compilador.

Volviendo a si la librería estándar puede considerarse parte o no del lenguaje, algo que siempre queda difuso, vamos a ver como ejemplo las cadenas de caracteres en Java. En realidad son objetos de la clase *String*, pero el compilador les da un tratamiento especial como si fuesen tipos de datos elementales. Utilizando las comillas dobles como delimitador se pueden manipular constantes de cadenas de caracteres que son, a todos los efectos, objetos de la clase *String*. De hecho, se permite el uso de operadores para estos objetos, algo que no es posible para objetos de otras clases salvo las clases que son una extensión de los tipos de datos nativos. En las últimas versiones de Java se permite que las constantes de tipo *String* se utilicen en la estructura de selección *switch* para el control de flujo de los programas. Formalmente, la clase *String* es una clase más de las predefinidas en la librería estándar de Java, pero no se concibe que un programa en Java prescindiera de la clase *String*.

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.

Con la aparición de los ensambladores la labor de programar resultó más eficiente aunque casi igual de compleja. Un lenguaje ensamblador es dependiente del hardware, de hecho, consiste en utilizar códigos nemotécnicos en lugar de unos y ceros para indicar las instrucciones y referenciar los registros. Cada modelo de procesador o microprocesador tiene su propio código máquina y por tanto su propio ensamblador.

Programar en ensamblador o en código máquina es lo que se conoce como **programar a bajo nivel**. El programador debe conocer muy bien la arquitectura del procesador para el cual programa, es decir, conocer qué registros tiene el procesador, qué propósito tiene cada registro, qué niveles de indirección se permiten para acceder a la memoria, cómo pueden ejecutarse subrutinas pasándoles argumentos y de qué mecanismos dispone el procesador para cambiar de contexto, es decir, para facilitar la multitarea.

A finales de la década de 1950 aparecieron los primeros lenguajes de programación que ya no eran dependientes del hardware sobre el cual se ejecutarían los programas. Los más representativos son COBOL, FORTRAN y ALGOL, aunque debemos señalar que hubo algunos diseños previos a partir de los cuales se concretaron estos tres, y que los tres sufrieron importantes cambios durante sus primeros años, de hecho el ALGOL 68 tenía importantes mejoras respecto de su antecesor el ALGOL 60. A estos lenguajes independientes del hardware se les conoce como **lenguajes de alto nivel** en contraposición a los lenguajes de bajo nivel. El concepto de compilador aparece ligado a los lenguajes de alto nivel. Un **compilador** es un programa que traduce los programas escritos en un lenguaje de programación de alto nivel, es decir independiente del procesador, al código máquina que sí ejecutará directamente el procesador. Aunque al principio los programadores debían conocer muchos detalles de la arquitectura sobre la que se ejecutarían sus programas, los lenguajes de alto nivel facilitaron, progresivamente, que los programadores pudiesen abstraerse de las características del hardware para concentrarse en los problemas a resolver.

Los lenguajes de alto nivel son mayoritariamente compilados, es decir, el código fuente se compila una sola vez para generar el código máquina. Después, el código máquina podrá ejecutarse tantas veces como sea necesario. Solo deberá compilarse de nuevo si se realizan modificaciones en el código fuente. Hay algunos lenguajes de programación que son interpretados, es decir, el programa fuente se va procesando y ejecutando línea a línea. Obviamente este proceso es más lento que ejecutar directamente el código máquina. Existe otra vía intermedia, puesta de moda gracias a Java, donde el código fuente se compila a un código intermedio independiente del hardware, conocido como *bytecode*, que una máquina virtual se encarga de ejecutar. Si el código intermedio se traduce a código máquina antes de ejecutarse o se ejecuta mediante una máquina virtual es algo que depende de la implementación.

En resumen, los lenguajes de programación pueden clasificarse en dos niveles, alto y bajo, según su dependencia del hardware. Para según qué tareas es bastante habitual combinar partes escritas en ensamblador dentro de un programa escrito en un lenguaje de alto nivel, por ejemplo para conseguir la versión más eficiente posible de algunas funciones que se utilizan repetidamente en un programa cuando la eficiencia es relevante.

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.

La propuesta de lenguajes de **cuarta generación** se desarrolló a lo largo de tres décadas, desde principios de los 70 a finales de los 90 del siglo XX. Nació con el propósito de diseñar nuevos lenguajes de programación que fuesen un refinamiento y una evolución de los lenguajes de tercera generación. Se buscaba que cada nueva generación de lenguajes de programación aumentase el nivel de abstracción con respecto a los detalles del hardware, es decir, que los programadores no necesitasen conocer detalles del sistema sobre el cual se ejecutarían sus programas. A cada nueva generación los lenguajes serían más cercanos al lenguaje natural de manera que la tarea de programar fuese más amigable, productiva y versátil.

La definición de lenguajes de cuarta generación fue evolucionando hasta quedar en desuso. El rasgo más relevante de esta generación es el de incluir soporte en la propia definición de los lenguajes para el acceso a bases de datos, generación de informes, funciones matemáticas más avanzadas, gestión del interfaz gráfico con el usuario o

desarrollo web. Todas estas funcionalidades tenían un denominador común: permitir manejar grandes colecciones de datos de manera sencilla para el programador, que le resultase transparente el tener que traer a la memoria del ordenador los datos con los que estaba trabajando. Por ejemplo, en el caso de acceder a una fuente de datos externa cuyo tamaño excede la capacidad de memoria del ordenador, en un lenguaje de cuarta generación sería transparente para el programador si en memoria solo hay disponible una porción de toda una lista de registros, se accedería como si todo estuviese en memoria y serían las funciones de la librería del propio lenguaje las que se encargarían de ir actualizando los datos en ambos sentidos (de la memoria a la fuente de datos externa y viceversa).

Lenguajes como Python, Ruby y Perl incluyen muchas características de los lenguajes de cuarta generación aunque en realidad son lenguajes de tercera generación algo más avanzados. Otros lenguajes como los orientados a objetos C++, C# y Java, incluyen una librería de funciones cada vez más completa que permite al programador trabajar con todas las funcionalidades previstas para los lenguajes de cuarta generación. Por ello, la frontera entre ambas generaciones de lenguajes ha quedado muy diluida. A pesar de ello, debemos recordar que hubo una explosión de nuevos lenguajes siguiendo la definición de lenguaje de cuarta generación, cada uno dedicado a un ámbito de aplicación. Pueden destacarse ABAP de SAP, Informix 4GL, Ingres 4GL, SQL, FORTH, Clipper, Data Flex, etc. En Wikipedia 4GL (2015) se encuentra una buena clasificación de los lenguajes de cuarta generación según ámbitos de aplicación, muchos de ellos en desuso actualmente.

Los lenguajes de **quinta generación** se definen como los lenguajes en los que programar consiste en definir los requisitos para resolver problemas en lugar de que un programador escriba un algoritmo. Muchos de los lenguajes pertenecientes al paradigma de programación declarativa son lenguajes de quinta generación, por ejemplo Prolog, OPS5 y Mercury. Los lenguajes de quinta generación están diseñados para que la computadora resuelva los problemas sin ayuda de programadores. De esta manera, el programador solo debe preocuparse de qué problemas pueden solucionarse y qué condiciones son necesarias que se cumplan para encontrar la solución a un determinado problema, sin necesidad de implementar algoritmos concretos que resuelvan los problemas.

La idea de quinta generación de lenguajes de programación se puso de moda en la década de 1980 considerándose como el futuro de la informática. Hubo quien se atrevió a vaticinar que reemplazarían a los lenguajes existentes incluso para el desarrollo de los sistemas operativos. Es destacable el hecho de que entre 1982 y 1993 en Japón se dedicaron muchas inversiones a la investigación en esta generación de lenguajes de programación con el objetivo de poner en funcionamiento una gran red de ordenadores utilizando esta tecnología. Pronto se vio que el paso que va desde definir los requisitos para desarrollar la solución a un problema hasta la implementación de los algoritmos era un salto cualitativo muy importante, labor que de momento sigue estando a cargo

de seres humanos. No es fácil generar automáticamente los algoritmos que resuelven problemas a partir de sus requisitos; no en vano es uno de los retos más importantes de la inteligencia artificial.

En resumen, el intento de clasificar los lenguajes de programación por generaciones quedó obsoleto. Por un lado, la diferencia entre tercera y cuarta generación quedó diluida a medida que los lenguajes de programación evolucionaron. Por otro, el desarrollo de los lenguajes de quinta generación ha quedado en algunos intentos interesantes, pero todavía distan mucho de lo que se perseguía en los objetivos de su definición original.

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.

La mejor manera hasta el momento de clasificar los lenguajes es según el paradigma de programación al que pertenecen. Paradigma de programación se define como el estilo fundamental de programación que establece la manera en que se estructurarán los programas en código fuente. Las capacidades, las características y el estilo de cada lenguaje de programación se definen por el paradigma o paradigmas que sigue el propio lenguaje (Tabla 3.1).

Vamos a ver que no existen fronteras bien definidas entre los distintos paradigmas de programación y muchos lenguajes reúnen características propias de varios. De hecho, un porcentaje no despreciable de lenguajes se diseñaron adoptando propiedades definidas en distintos paradigmas con el propósito de cubrir el mayor abanico posible de necesidades por parte de los programadores. Utilizando un símil biológico podríamos decir que los lenguajes de programación experimentan una evolución darwiniana. Para ello nos basamos en el hecho de que se han diseñado muchos, han dejado de utilizarse la mayoría, y los más utilizados sufren una constante evolución para adaptarse al medio, es decir, a las necesidades de los ingenieros del software que desarrollan programas. Las necesidades a cubrir por los lenguajes de programación son muy distintas dependiendo del entorno de aplicación. Por ejemplo, para desarrollar aplicaciones de gestión (contabilidad, facturación o presupuestos) se requieren lenguajes que faciliten al programador implementar eficazmente la parte dedicada a la interacción con el usua-

rio y la parte de acceso a las bases de datos. En el caso del desarrollo de sistemas operativos se requieren lenguajes que permitan interactuar con el hardware, sin restricciones para acceder a posiciones de memoria concretas o a puertos de entrada/salida, a fin de programar componentes de los sistemas operativos como son los manejadores de dispositivos, que nunca interactúan con el usuario.

Tabla 3.1. Algunos de los paradigmas de programación más representativos

Paradigma	Lenguajes más representativos
Propósito general	C
Programación imperativa	C, Pascal, COBOL, FORTRAN, ADA
Programación estructurada o procedural	C, Pascal, Modula-2, C++, Java
Programación declarativa	Prolog, LISP, Haskell
Programación funcional	LISP, OCaml, Erlang, Haskell
Programación lógica	Prolog
Programación orientada a objetos	Simula, Smalltalk, C++, Objective-C, Java
Programación de sistemas	C, ADA
Programación simbólica	LISP

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/I.

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.

Programación estructurada o procedural

Este paradigma de programación aparece en la historia de los lenguajes de programación como un refinamiento para facilitar la legibilidad de los programas y el desarrollo de aplicaciones de mayor envergadura con un menor esfuerzo humano. Su principal característica es la de estructurar el código fuente en bloques y procedimientos (subrutinas o funciones), tal que cada procedimiento se encarga de realizar una tarea concreta y puede ser invocado desde otras partes del programa tantas veces como sea necesario, evitando así tener que escribir varias veces el mismo código. También incluye mecanismos de control del flujo de los programas como los bucles *for* y *while*. El principal objetivo de este paradigma es eliminar o reducir el uso de la instrucción GOTO, ampliamente utilizada en los primeros lenguajes de programación. El uso de dicha instrucción de salto a una línea concreta del código permitía desarrollar programas sin procedimientos ni bloques. En consecuencia, el programador era incapaz de entender su propio código pasados unos meses después de su desarrollo.

La programación estructurada o procedural comienza a desarrollarse como paradigma de programación en la década de 1960 y se consolida durante la siguiente. Es la primera evolución de los paradigmas de programación de alto nivel y representa un salto cualitativo muy importante en la historia de los lenguajes de programación. Los primeros lenguajes diseñados bajo este paradigma fueron ALGOL, PL/1, Pascal y ADA. COBOL y FORTRAN no lo eran en principio pero posteriormente evolucionaron hacia la programación estructurada, ya que el abuso de la instrucción GOTO hacía prácticamente ilegibles muchos programas escritos en estos dos lenguajes.

Programación modular

Una variante importante de la programación estructurada es la programación modular, cuyo principio de diseño consiste en desarrollar los programas en módulos independientes e intercambiables. Un módulo incluye como mínimo un procedimiento o función y las estructuras de datos asociadas. La idea de interfaz de aplicación o API (*Application Programming Interface*) nace de la programación modular. Definir una API para un módulo permite a los programadores abstraerse de cómo es la implementación interna de dicho módulo. Así, un programador que utilice módulos implementados por otros programadores únicamente necesita saber cómo invocar a las funciones de dicho módulo, asume que la funcionalidad proporcionada es correcta y concentra su esfuerzo en solucionar problemas a otro nivel.

El principal objetivo de la programación modular es facilitar el desarrollo de grandes aplicaciones software que deben ser diseñadas como un conjunto de módulos bien definidos, de manera que el desarrollo de cada módulo puede ser responsabilidad de un programador o de un equipo de programadores. Simplemente deben asegurarse de que el módulo ofrece la funcionalidad definida durante el diseño global y sigue la interfaz (API) acordada. La programación modular podría verse como un punto intermedio

entre la programación estructurada y su natural evolución hacia la programación orientada a objetos.

Programación orientada a objetos

La programación orientada a objetos (*Object Oriented Programming*, OOP) puede entenderse como una evolución natural de la programación modular que ya era, a su vez, una mejora de la estructurada. La idea subyacente de la OOP es intentar modelar los objetos de la realidad de manera abstracta dentro de los programas. Es por ello que el programador se encarga de diseñar e implementar clases a partir de las cuales se crearán objetos durante la ejecución del programa. Un objeto es una instancia de una clase. La ejecución de un programa desde la óptica OOP es un conjunto de objetos que interaccionan entre sí.

La tarea de programar mediante lenguajes OOP consiste en diseñar clases de objetos. En el diseño de cada clase se definen los atributos que contendrán los objetos de la clase, es decir, se define la estructura de datos que tendrán los objetos de dicha clase, y también se implementan los métodos que podrán ejecutarse con respecto a los objetos, métodos que tienen acceso a los atributos para utilizar su valor en operaciones o modificarlos. En cada clase se programan los métodos que gestionarán la información almacenada en los atributos, o lo que es lo mismo, es como definir el comportamiento de los objetos. Llegados a este punto, podemos definir los objetos como contenedores de información que saben qué hacer con ella.

Pero la OOP es algo más complejo y completo, supone un importante salto cualitativo en programación porque facilita el desarrollo de grandes aplicaciones software por equipos de programadores. La OOP reúne las ventajas de la programación modular y sus principales características son:

- **Herencia:** permite diseñar clases como derivadas de otras clases con la ventaja añadida de reutilizar código ya programado. Las clases derivadas o sub-clases son especializaciones de la super-clase.
- **Polimorfismo:** objetos de una clase derivada pueden actuar como objetos de la super-clase, según el contexto en el que se utilicen los objetos dentro del programa.
- **Enlace dinámico** (*Dynamic Dispatch*): el polimorfismo permite trabajar con objetos de la super-clase de manera que sobre cada objeto en particular se ejecutarán los métodos de la sub-clase correspondiente, aunque sea invocado como si fuese un objeto de la super-clase. Es en tiempo de ejecución cuando se resuelve qué método se ejecutará, porque en tiempo de compilación no se sabe de qué sub-clase será cada objeto de la super-clase.
- **Sobrecarga de métodos** (*Method Overloading*): pueden existir distintos métodos con el mismo nombre porque tienen la misma funcionalidad, pero cada uno con un perfil distinto, es decir, con distinto número y/o tipo de argumentos.

- **Abstracción:** los atributos de los objetos pueden definirse como privados, en cuyo caso quedan ocultos para cualquier otra clase y solo son accesibles desde los métodos de la propia clase. El programador que utilice en su código objetos de clases desarrolladas por otras personas no necesita conocer los detalles internos de su implementación, sino únicamente los métodos que puede invocar para interactuar con objetos de dichas clases desde su propio código.

En OOP se propician otras buenas prácticas de la programación como la encapsulación y la reusabilidad. Estas buenas prácticas son más fáciles de llevar a cabo gracias a las características de la OOP enumeradas anteriormente. Los lenguajes más significativos de la OOP son: Simula, Smalltalk, C++, Objective-C, Delphy, Java, C#, Perl, Python, Ruby y PHP.

Programación declarativa

La programación declarativa es una manera de programar que puede entenderse como opuesta a la programación imperativa, es un estilo de construir la lógica de los programas que busca expresar la lógica de los cálculos a realizar para obtener la solución a un problema sin describir los algoritmos mediante instrucciones de control de flujo. Otra manera de describir este paradigma es que el programador debe definir qué condiciones deben cumplirse para obtener la solución a un problema en lugar de describir la secuencia de instrucciones que lo resuelve. Este planteamiento es el mismo que el de los lenguajes de quinta generación. A partir de este paradigma se crearon varios refinamientos; los dos subparadigmas más destacados son la programación lógica y la funcional.

Por un lado, la **programación lógica** se basa en la lógica formal. Los programas son conjuntos de construcciones lógicas que siguen la sintaxis de la lógica formal para expresar hechos y reglas sobre el problema a resolver. Dentro de la programación lógica existen a su vez otros subparadigmas que no citaremos aquí. Los lenguajes de programación que siguen este paradigma son Prolog, ASP (*Answer Set Programming*) y Datalog.

Por otro lado, desde la óptica de la **programación funcional** las soluciones a problemas se definen en términos de evaluar funciones matemáticas, sin tener en cuenta el cambio de estado del programa ni variables que cambian de valor. Este paradigma tiene su origen en el cálculo Lambda, un sistema formal desarrollado en los años 30 del siglo XX para investigar en el campo de la computabilidad. Los lenguajes de programación más relevantes de este paradigma son Common Lisp, Scheme, Clojure, Racket, Erlang, OCaml, Haskell y F#. Este paradigma se usa en otros dominios de aplicación como el lenguaje R para estadística, el Wolfram dentro de Mathematica, el SQL para bases de datos y también en utilidades como Lex y Yacc.

OTROS LENGUAJES

Además de los lenguajes de programación vistos, ampliamente utilizados para desarrollar programas de una cierta envergadura, existen otros lenguajes con otros propósitos que enumeramos sin detallar.

- **AWK** es un lenguaje de programación diseñado para procesar datos basados en texto, ya sean ficheros o flujos de datos. Su nombre deriva de las iniciales de los apellidos de sus autores: Alfred Aho, Peter Weinberger y Brian Kernighan. Existe un comando en los sistemas basados en Unix con el nombre `awk`.
- **Programación de SHELL**. Los sistemas operativos, incluido Windows, disponen de un programa conocido como intérprete de comandos o *shell*. Originalmente era el interfaz de usuario de los sistemas operativos y, por tanto, cada uno tenía el suyo. El usuario debe conocer los comandos y teclearlos para ir ejecutando instrucciones o lanzando programas. La evolución de los intérpretes de comandos de Unix trajo consigo que también pudieran crearse *shell-scripts* y ejecutarse automáticamente. Un *shell-script* es un fichero de texto cuyo contenido es un programa en *shell* que ejecuta comandos del sistema operativo o programas de usuario. Disponen de mecanismos de control de flujo al estilo de los lenguajes de programación estructurados y permiten definir funciones para invocarlas posteriormente desde cualquier otro punto del código *shell*. Los lenguajes de programación en *shell* más conocidos son Bourne Shell, C Shell, Korn Shell y Bourne Again Shell.
- **PHP**. Es un lenguaje puramente orientado a ejecutarse en servidores web para crear de manera dinámica contenidos Web. Al navegador (parte cliente) solo le llega código HTML o de cualquier lenguaje de *script*, como Javascript, pero no le llega el código PHP. Cumple el paradigma de orientación a objetos y dispone de una extensa librería de funciones para trabajar con textos y acceder a bases de datos, de manera que podríamos decir que es un lenguaje de cuarta generación.
- **SQL**. *Structured Query Language*, también es un lenguaje de los conocidos como de cuarta generación y es específico de bases de datos. Se pueden programar módulos para ejecutarse en el motor de las bases de datos y también para interactuar con ellas desde las aplicaciones cliente, es decir, invocando instrucciones SQL dentro de programas escritos en otros lenguajes de programación más convencionales.
- **Postscript**. Es un lenguaje específico para impresión que interpretan la mayoría de impresoras láser. No es un lenguaje para programar como entendemos la programación, sino para describir las instrucciones de impresión combinando texto e imágenes. También permite dibujar gráficos mediante instrucciones que permiten trazar líneas y otras formas geométricas.

COMENTARIOS FINALES

La finalidad de esta revisión histórica es que el lector tenga una idea de cuál ha sido la evolución de los lenguajes de programación, qué características se definen en cada paradigma de programación, y que disponga de referencias bibliográficas para profundizar en el tema si le resulta de interés. Pero sobre todo, que tenga dos ideas claras sobre la programación: que aprender a programar no depende del lenguaje ni del paradigma, y que, según el ámbito en el que se vaya a trabajar, sí es importante utilizar un lenguaje de programación u otro, por lo que esperamos que este texto le pueda ayudar a tener los criterios suficientes para decidirse por el más adecuado a sus necesidades.

BIBLIOGRAFÍA

O'REILLY (2015), *Poster of Programming Languages*,

http://cdn.oreillystatic.com/news/graphics/prog_lang_poster.pdf

LEVENEZ, E. (2015), *Computer Languages History*, on-line resource:

<http://www.levenez.com/lang/>

KNUTH, D. E. (1968), *The Art of Computer Programming*, Volume 1: Fundamental Algorithms, Reading: Addison-Wesley.

GROGONO, P. (2002), *The Evolution of Programming Languages*, Course Notes for COMP 348 and COMP 6411, Department of Computer Science, Concordia University, Montreal, Quebec.

Gabriela Andreu García
Silvia Terrasa Barrena
Álvaro Doménech Pujol
Universitat Politècnica de València

Resumen: 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 remarquables que forman parte de la historia y evolución de los sistemas operativos.

INTRODUCCIÓN

Los sistemas operativos fueron ideados para administrar todos los componentes del computador: desde el procesador, la memoria y los periféricos (impresora, teclado, discos, etc.) hasta el software (programas) y los datos. Un sistema operativo está formado por un conjunto de programas que proporcionan un entorno eficiente y seguro para la ejecución de tareas. Las aplicaciones informáticas actuales requieren que se realicen simultáneamente muchas operaciones de cálculo, comunicación y transferencias de datos. Este funcionamiento exige un alto grado de coordinación entre los componentes del computador, coordinación que es llevada a cabo por el software del sistema o sistema operativo (en adelante SO).

La evolución de los SO siempre ha ido unida a las innovaciones tecnológicas que conforman la arquitectura del computador, a los sistemas de comunicación y a los sistemas de almacenamiento de información. Así, con el paso de los años, los SO han conseguido adaptarse a la naturaleza, cantidad y calidad de los elementos que los avances tecnológicos permitían incorporar al computador, abstrayendo sus complejidades y proporcionando sus servicios de una manera amigable. Con todo ello, los SO han experimentado un fuerte incremento de sus responsabilidades, adaptándose a las nuevas exigencias de los usuarios que requerían un aumento de prestaciones, nuevas funcionalidades y mucha seguridad (Tabla 4.1).

Tabla 4.1. Aumento de las funciones de los sistemas operativos a lo largo del tiempo

Década	Funciones del SO	Tipo de SO
1950	Secuencia automática de trabajos Procesamiento por lotes Acceso a dispositivos de E/S	Sistemas por lotes simples (<i>batch processing</i>)
1960	Planificación de trabajos y multiprogramación Gestión y protección de memoria Gestión de disco	Sistemas por lotes multiprogramados
1970	Interacción usuario/máquina Multiprogramación y sincronización de procesos Gestión y protección de archivos Gestión de memoria virtual	Sistemas de tiempo compartido
1980	Interfaz gráfica con ventanas y ratón (GUI) Capacidad multimedia y <i>plug-and-play</i> Acceso a red	Sistemas para computadores personales
1990	Multiprocesador y <i>multicore</i> Tolerancia a fallos Comunicación entre nodos Compartición de carga y recursos	Sistemas paralelos Sistemas de tiempo real Sistemas distribuidos Computación en <i>grid</i>
2000	Almacenamiento y computación remota Trabajo con aplicaciones web	Nube (<i>cloud</i>) Web OS

En las dos últimas décadas el uso del computador ha proliferado de una forma expansiva en toda la población. En la actualidad es usual que haya al menos un computador en cada vivienda, ya sea en forma de portátil, computador personal (PC, *Personal Computer*) o tableta. También es habitual escuchar a personas no expertas en materia de computadores hablar de sus experiencias con algún sistema operativo como Windows, Linux, FreeBSD o MacOS X e incluso Android o IOS. Las interfaces gráficas (GUI, *Graphical User Interface*) son las actuales encargadas de mostrar las utilidades del sistema operativo en forma de iconos (Figura 4.1).



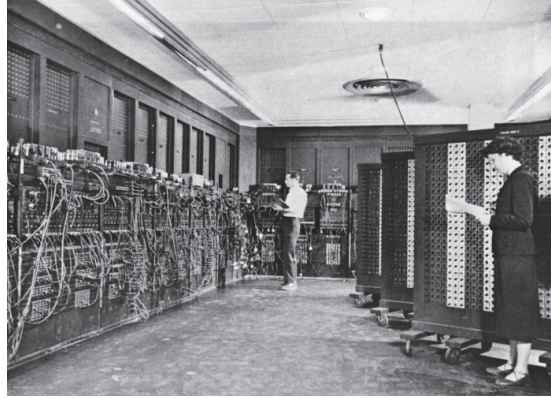
Figura 4.1. Aspecto de algunas GUI que se proporcionan con sistemas operativos muy populares en la actualidad

Sin embargo, como veremos enseguida, hubo una época en la que no existían los SO, no había lenguajes de programación, ni ventanas, ni ratón, y el trabajo con máquinas de cómputo resultaba tedioso e ineficiente.

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.



Fuente: Wikipedia. Ejército de los Estados Unidos

Figura 4.2. Paneles cableados y de control del ENIAC

El ENIAC disponía de veinte acumuladores para almacenar datos (no tenía una memoria como los computadores actuales) y la ejecución de tareas resultaba tediosa. Las operadoras, personas expertas en la máquina, se pasaban días ajustando interruptores y enchufando cables. La comprobación de errores necesitaba una gran cantidad de mano de obra. Tampoco disponía de teclado para introducir datos ni de pantalla para visualizar información. Parte de la entrada/salida se llevaba a cabo a través de una lectora y una perforadora de tarjetas. Los acumuladores mostraban los números a través del encendido de pequeñas bombillas. Tras la experiencia del ENIAC se llegó a la conclusión de que debía existir una manera más fácil de introducir la información en el computador y trabajar con él.

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.

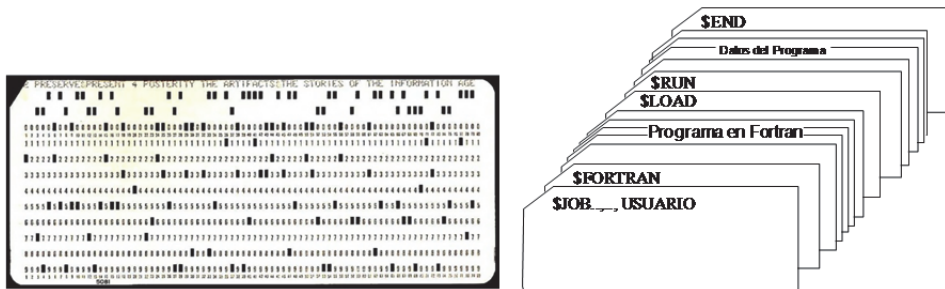


Figura 4.3. Tarjeta perforada y representación de un programa con tarjetas de control

Los usuarios de un sistema operativo por lotes no interactúan directamente con el ordenador, sino que preparan sus trabajos perforando sus tarjetas para proporcionarlas al operador del computador que es el encargado de introducir las en la máquina. En definitiva, en esta época un programa comienza con la lectura de datos de un dispositivo de entrada (perforadora, cinta) para continuar con la ejecución de sus instrucciones o cálculos y finalizar con la transmisión de resultados a una impresora. Los usuarios no tienen la posibilidad de interactuar con su trabajo, siendo impensable que el programa solicite un dato al usuario mientras se ejecuta. Si mientras se ejecuta un programa ocurre un error, las tarjetas de este son devueltas al usuario para que haga las modificaciones necesarias.

El resultado de un programa ejecutado con éxito es recogido por el operador a la salida de la impresora y entregado al usuario. Seguidamente el operador toma un nuevo conjunto de tarjetas para introducir las en la máquina y comienza otra vez el ciclo. Debido a esta transición manual de trabajos, una máquina muy costosa quedaba inactiva durante demasiado tiempo.

A mediados de la década de 1950 ordenadores de éxito como el IBM 701 y IBM 704 incorporan un sistema automático de transición de trabajos mediante el uso de tarjetas de control (Figura 4.3), lo que permitió reducir los tiempos perdidos entre trabajos.

Con la incorporación del transistor, la velocidad de cálculo de los computadores aumentó considerablemente. Sin embargo, y a pesar de la secuencia automática de trabajos, el procesador (CPU) continúa permaneciendo ocioso con demasiada frecuencia. Los dispositivos de entrada/salida (E/S) son lentos comparados con el procesador y hasta que no finaliza una tarea no se comienza a trabajar con la siguiente. Así, mientras se imprime el resultado de un trabajo, la máquina simplemente espera que la impresora finalizase (Figura 4.4A).

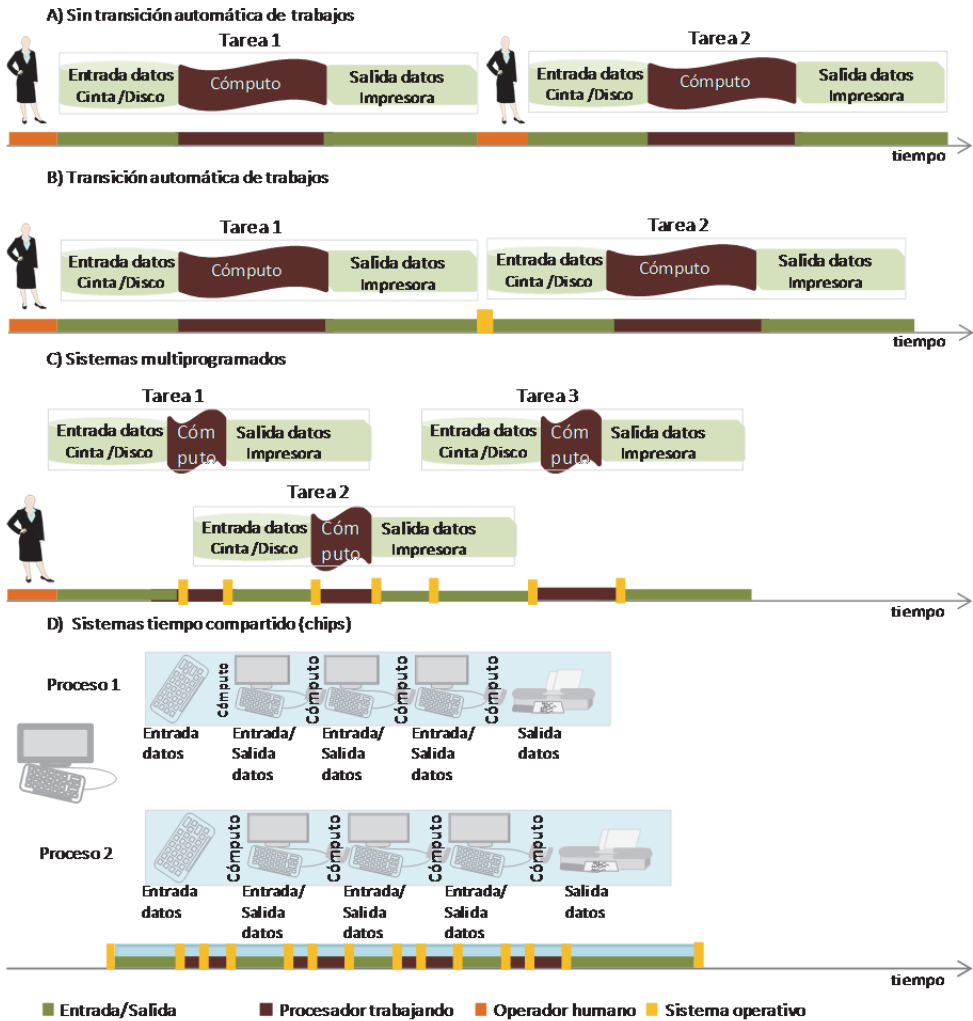


Figura 4.4. Representa la ejecución de procesos en los diferentes modos de trabajo y la proporción de tiempo invertida por el operador, el procesador, la E/S, y el sistema operativo en cada uno de ellos

En la década de los años 60, con las cintas magnéticas y los discos como medios de almacenamiento de información rápidos, aparece un nuevo modo de trabajo: el *spooling* (*Simultaneous Peripheral Operation On-line*). Este método consigue obtener un mayor rendimiento del computador ya que permite trabajar a los dispositivos de E/S con cierta autonomía haciendo posible que la unidad central de proceso de una máquina esté ocupada en cálculos mientras ciertos dispositivos como la impresora están activos.

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).

Una tarea en ejecución es lo que los alumnos del actual grado en Ingeniería Informática entienden por proceso. Un proceso tiene asignado sus propios valores y su propia zona de memoria que deben ser respetados durante los intervalos de ejecución de otros procesos. El concepto de proceso es dinámico ya que a lo largo de su vida, es decir, durante su ejecución, sus necesidades cambian: ahora el proceso solicita usar una pantalla, después requiere un poco más de memoria, a continuación una impresora, etc. Según lo expresa Corbató, *one can mean using different parts of the hardware at the same time for different tasks*. La multiprogramación obtiene un mayor rendimiento del sistema de cómputo ya que, mientras un proceso trabaja con el procesador, otros procesos pueden estar utilizando los periféricos.

La convivencia de varios procesos en una única máquina exigió nuevas funciones y responsabilidades al SO:

- Gestionar y proteger la memoria del computador: el SO debe decidir la zona de memoria donde ubicar cada proceso. Las zonas de memoria de los procesos deben ser independientes entre sí y estar protegidas para evitar interferencias indebidas entre ellos.
- Planificar la alternancia de procesos en la CPU: el SO debe decidir en cada momento qué proceso puede ejecutar instrucciones, es decir, disponer de la CPU.
- Gestionar los dispositivos de E/S: el SO debe evitar conflictos en el uso de los periféricos.

Hasta este periodo los computadores eran diseñados según su dedicación final. Así, había marcadas diferencias técnicas entre los computadores destinados a cálculo científico y los considerados de propósito general o comercial. Estas diferencias implicaban diseñar un SO diferente para cada modelo de ordenador. El desarrollo y mantenimiento de SO totalmente distintos era bastante costoso para los fabricantes.

IBM intentó dar respuesta a este problema introduciendo lo que se denominó familia de computadores. La familia IBM 360 (1964) comprendía desde el modelo minicomputador (con memoria de 8 KBytes) al modelo superordenador (4 MBytes) construido para el sistema de defensa antimisiles norteamericano. La idea era diseñar una arquitectura del hardware y un único SO compatible para todas ellas. A pesar de las diferencias entre modelos, toda la familia IBM 360 tenía el mismo conjunto de instrucciones y en su construcción ya se utilizaron circuitos integrados a pequeña escala.

IBM pretendía que tanto el SO como el software desarrollado para los IBM 360 funcionara en todos los modelos de la familia y fuese útil tanto para usos comerciales como científicos. El resultado fue el OS/360, un sistema operativo enorme y de gran complejidad con millones de líneas escritas y con miles de errores. Cada nueva versión del OS/360 intentaba corregir los errores detectados en la versión anterior pero inevitablemente introducía otros. Esta experiencia provocó que años después Frederik P. Brooks, uno de sus diseñadores, escribiese un libro sobre qué no hacer a la hora de diseñar un sistema operativo. En la portada del libro aparece una imagen con varios dinosaurios, animales de gran peso y envergadura, intentado avanzar en un terreno pantanoso (Figura 4.5). Esta imagen fue elegida por el autor para reflejar las sensaciones y el estado de ánimo soportado durante el diseño de la familia OS/360.

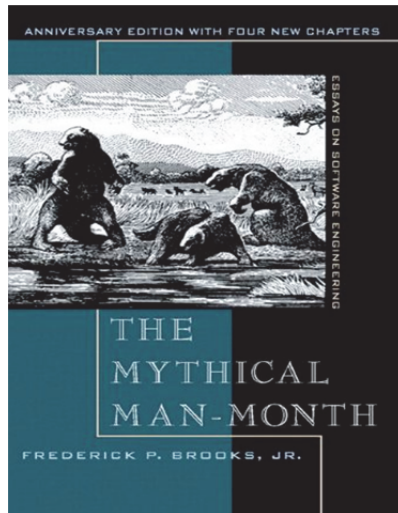


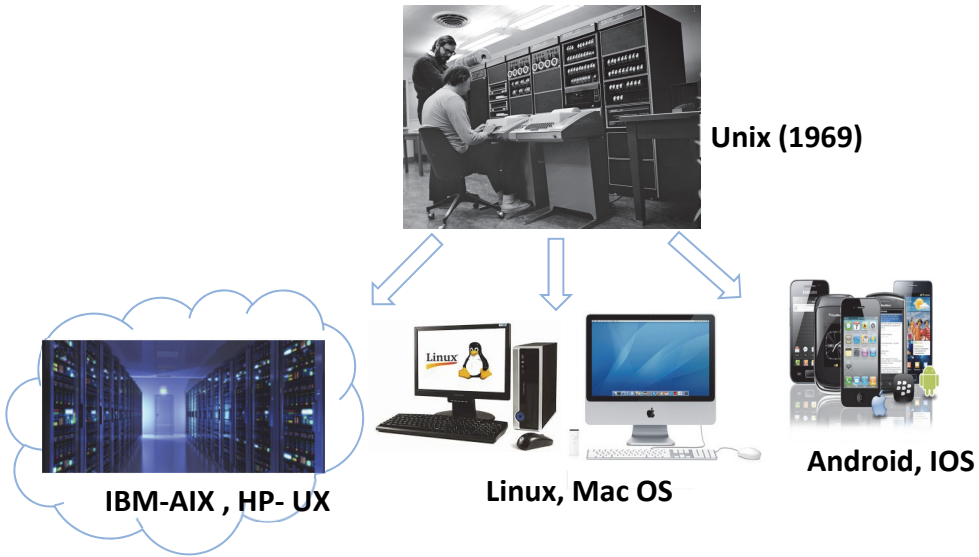
Figura 4.5. Portada del libro de Frederik P. Brooks, *The Mythical Man-Month*, donde la lenta marcha de los dinosaurios representa el esfuerzo realizado en el desarrollo del OS/360

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.

El tiempo compartido permite que las aplicaciones proporcionen sus resultados de forma rápida en un entorno de trabajo más interactivo que los precedentes. En los años 60 ese entorno consistía en un teletipo que combinaba un teclado para entrada de datos y una impresora (Figura 4.6) para salida de datos. Ambos periféricos funcionaban en modo texto, sin gráficos ni colores. Posteriormente la impresora fue reemplazada por una pantalla de texto (no gráfica). Únicamente faltaba dotarle de capacidad gráfica y color para, junto con el teclado y el ratón, dar lugar al entorno de trabajo más habitual de nuestros días.



Fuente: Wikipedia

Figura 4.6. En la parte superior los creadores de Unix Ken Thompson (sentado) y Dennis Ritchie trabajando con un DEC PDP-11. En la parte inferior, sistemas operativos de uso actual basados en Unix

La clave de los sistemas de tiempo compartido es mantener diversos procesos concurrentes de manera que cada uno evolucione en función de los requerimientos del momento. Cualquier usuario moderno tiene en su computador una docena de aplicaciones abiertas (el navegador, el procesador de textos, la calculadora, un juego de buscaminas, etc.) y el sistema operativo detiene a la mayoría de ellas porque esperan alguna interacción. Cuando el usuario focaliza su actividad en una aplicación esta continúa a partir del punto en que se quedó detenida. Igualmente, el navegador también se queda esperando a que llegue la respuesta del servidor tras una solicitud y la aplicación de la impresora mostrará una notificación cuando el documento haya sido impreso.

Por todo ello los computadores actuales, sean servidores web o teléfonos móviles, tienen un sistema operativo de tiempo compartido. Los sistemas operativos de tiempo compartido, junto con la tecnología de comunicaciones y redes que se creó a partir de los años 60, son la base de Internet. En este contexto es imprescindible mencionar a Unix y sus creadores.

El sistema operativo Unix fue creado por Dennis Ritchie y Ken Thompson en los Laboratorios Bell (1969). Unix creció y evolucionó hasta nuestros días mediante múltiples contribuciones de empresas y universidades. El éxito de Unix recae en ser un sistema operativo potente e interactivo que se ha diversificado para funcionar con diferentes apariencias sobre máquinas con hardware muy distintos. Unix nació para trabajar con computadores clásicos con multitud de usuarios en línea y ha conseguido permanecer en los actuales dispositivos inteligentes por diminutos que estos sean. So-

bre la base de Unix (Figura 4.6) han aparecido muchos sistemas operativos de gran éxito como Linux, Android, Mac OS, IBM-AIX, HP-UX. Unix fue especialmente relevante en el desarrollo de la arquitectura de Internet y hoy por hoy se encuentra como soporte en dos de cada tres servidores web. Unix es el sistema operativo más estudiado en los grados universitarios y titulaciones superiores durante las cuatro últimas décadas.



Fuente: Wikipedia. www.mac-history.net

Figura 4.7. Computadora Personal IBM PC modelo 5150 (1981) con una pantalla de tipo texto.
Computadora Personal Macintosh Lisa 2 con interfaz gráfica (1984)

SISTEMAS PARA COMPUTADORES PERSONALES (1975-ACTUALIDAD)

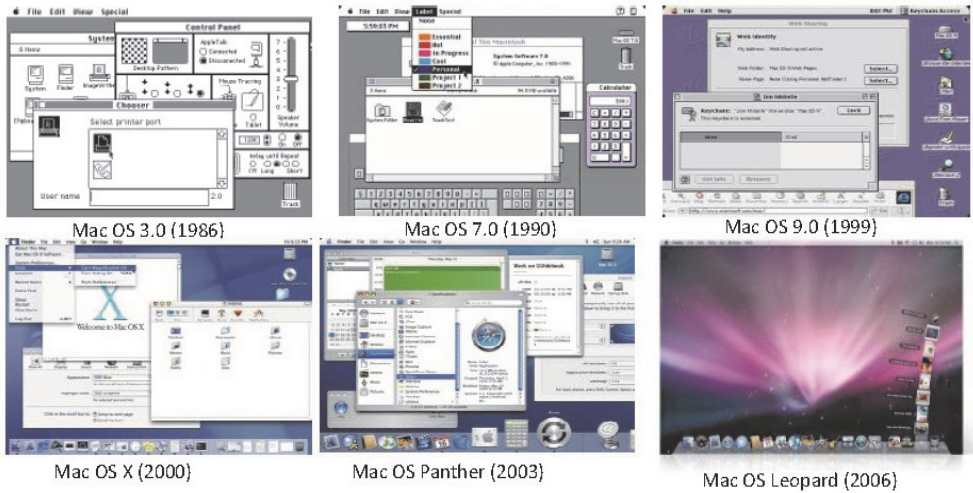
El tamaño y precio de los computadores se reduce considerablemente con la llegada del microprocesador dando lugar a la comercialización de los computadores personales (PC, *personal computer*) (Figura 4.7). Surgen múltiples sistemas operativos para PC que desde sus inicios fueron sistemas de tiempo compartido. Al principio solo podían ejecutar un programa pero rápidamente se eliminó esta restricción pudiendo ejecutar varios a la vez. La Tabla 4.2 resume las principales innovaciones de los sistemas con mayor trascendencia.

Tabla 4.2. Características de los sistemas operativos para PC de mayor trascendencia

Nombre	Innovación más destacable	Versión actual
MS-DOS (1981)	Diseñado para arrancar desde disco flexible (DOS, <i>Disk Operating System</i>)	Windows 8
Apple Macintosh (1984)	Diseñado desde el principio para trabajar a través GUI. Se situó a la cabeza en el sector de autoedición y gráficos.	MAC OS X
Linux (1991)	Sistema de libre distribución. Sus archivos se encuentran en muchos sitios de internet Creado por Linus Torvalds	Multitud de distribuciones: Redhat, Ubuntu Fedora, Debian Suse, Opensuse, etc.

Los primeros sistemas operativos para PC arrancaban insertando un disco flexible en su disquetera y apretando una tecla. Ahora poner a trabajar la máquina es algo sencillo. Comienza una era en la que prácticamente todos los hogares disponen de un PC y en la que todos los miembros de la familia quieren utilizarlo. Surgen dos nuevos términos en las conversaciones entre amigos: las palabras software y hardware. Y aparece un nuevo perfil de usuario no experto dotado de más entusiasmo que de conocimientos técnicos.

Para estos nuevos usuarios, los SO han de ofrecer interfaces mucho más amigables y cómodas que oculten los detalles técnicos del computador. Las interfaces gráficas acercan la complejidad del computador a usuarios no expertos (Figura 4.8) y facilitan el trabajo interactivo con múltiples aplicaciones simultáneas. Pero, además, menús, ventanas y ratón son la manera natural de interactuar con las nuevas aplicaciones del PC como las herramientas de oficina, los juegos, tratamiento de sonido e imagen, etc. Este mundo multimedia, que ahora nos es cotidiano, era inimaginable antes de la creación del PC. Nadie podía concebir el gasto de un millón de dólares en una máquina que no cabía en un salón-comedor para simplemente jugar y reproducir ficheros MP3.



Fuente: Wikipedia

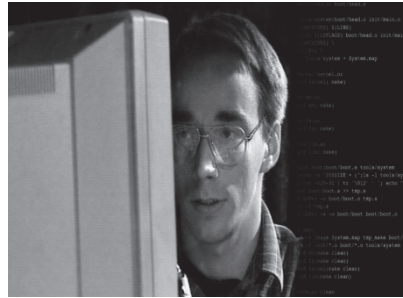
Figura 4.8. Aspecto de la interfaz gráfica de diferentes versiones del sistema Mac OS a lo largo de su historia

El Mac OS (*Macintosh Operating System*, 1984) de Apple se considera el sistema operativo líder en el uso de entornos gráficos. Pero, en realidad, ésta no fue la primera interfaz gráfica aunque sí fue la primera con gran éxito, probablemente por su precio. Los entornos gráficos requieren potencia de cálculo, capacidad de memoria y pantallas con capacidades gráficas, lo que encarecía el computador enormemente en aquella época. La Apple Lisa tenía un precio de 10.000 dólares frente a la Xerox Alto con un costo de 32.000 dólares y la Xerox Star de 16.600 dólares. Por su parte Microsoft lanzó en 1985 su primer entorno gráfico: Windows 1.0.

En los años 1990, con tecnología apropiada y muy asequible, las corporaciones Apple y Microsoft inician una pugna por ser los primeros en proporcionar a los usuarios lo más novedoso. Ambas empresas comienzan a desarrollar sistemas operativos para PC de manera continua. Cada vez que una nueva versión de sistema operativo llega al mercado inmediatamente comienzan el desarrollo de la siguiente versión (Tabla 4.3). Microsoft alcanzó un gran éxito comercial con la versión de Windows 3.0 y pasó a ser la empresa líder de programas informáticos con unas ventas anuales de más de mil millones de dólares. Apple llegó a presentar una demanda (que fracasó) contra Microsoft por imitar su interfaz con ventanas. La escalada de Apple empezó en 2001 con la presentación del Mac OS X y el iPod, pero su gran auge llegó en 2007 con el iPhone y el Apple TV.

Tabla 4.3. Sucesión de versiones de sistemas operativos para PC producidas por Microsoft y Apple.

Año	Versión Microsoft para PC	Versión Apple para PC
1984, 1985	MS-DOS, Windows 1.0	Mac OS 1.0, Mac OS 2.0
1986, 1987	–	Mac OS 3.0, Mac OS 4.0
1988, 1990	Windows 2.0, Windows 3.0	Mac OS 5.0, Mac OS 7.0
1992, 1993	Windows 3.1, Windows NT	–
1995	Windows 95	–
1997, 1998	–, Windows 98	Mac OS 8, –
1999, 2000	–, Windows 2000	Mac OS 9, Mac OS X
2001	Windows XP	Mac OS Cheetah, Puma
2002, 2003, 2004	–	Jaguar, Panther, Tiger
2007, 2009	Windows Vista, Windows 7.0	Leopard, Snow Leopard
2010, 2012	–	Lion, Mountain Lion
2013, 2014	–, Windows 8	Mavericks, Yosemite



Fuente: Athanasios Kasampalis en flickr. com

Fuente: Gerard van Schip en flickr.com

Figura 4.9. A la izquierda Steve Jobs y Steve Wozniak, fundadores de Apple. Y a la derecha Linus Torvalds creador del sistema operativo Linux

A partir de 1990 los sistemas operativos para PC (Windows 3.11 trabajo en grupo, Mac OS 7.0) ofrecen herramientas sencillas para manejar las comunicaciones. Los usuarios perciben esta tecnología en dos ámbitos: conectando computadores propios para intercambiar archivos y utilizando los crecientes recursos de Internet. En 1993, aparecen los primeros navegadores web que pronto se distribuyen gratuitamente junto con el sistema operativo. Los usuarios optan entre navegadores como Mozilla Firefox, Internet Explorer, Opera o Google Chrome.

Con la informática de masas, con millones de computadores interconectados conteniendo información sensible (imágenes privadas, cuentas bancarias o datos para administraciones públicas) sobre millones de usuarios, la seguridad y la privacidad son propiedades clave de los SO. Los ataques provenientes de la red han sido desde el principio de Internet la inquietud de millones de usuarios. Un ataque transformaba cualquier computador en origen de nuevos ataques y podía suponer la pérdida de datos y la reinstalación del software. Por ello son usuales las copias de seguridad en discos y cintas.

En los sistemas previos a la era de Internet, como Unix, la clave de acceso estaba limitada a 8 caracteres; posteriormente, en Windows 95 se limita a 14 caracteres, Windows XP a 32 y Windows 7.0 a 127 caracteres. La historia de la seguridad de los sistemas se ve reflejada en la necesidad de permitir claves de acceso con mayor número de caracteres.

La eficiencia y seguridad en la gestión de las comunicaciones de los sistemas operativos son esenciales en el nuevo mundo digital denominado nube en el que cálculo, almacenamiento de datos y otros servicios se proporcionan en la red. Los usuarios confían toda su información a la nube almacenándola y accediendo a ella desde cualquier puesto de trabajo en cualquier lugar del mundo. El software y sus actualizaciones se suelen adquirir por Internet y, de forma habitual, tienen carácter semiautomático.

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.

CONCLUSIÓN

Con el siglo XXI la informática se hace ubicua, estando presente en todos los entornos de ocio, de conocimiento, de desarrollo y trabajo. Prácticamente todos los aparatos electrónicos conocidos utilizan microprocesadores para ofrecer versatilidad y llevan incorporado un sistema operativo (teléfonos móviles, reproductores de DVD, computadoras, radios, enrutadores, etc.). Además, ofrecen interfaces gráficas de usuario para gestionarlos.

No sabemos qué nos puede deparar la informática en las próximas décadas, pero podemos asegurar que detrás de cada avance tecnológico, será necesario un avance en los sistemas operativos. Por último, merece la pena recordar que el gran secreto de Steve Jobs fue no olvidar que las máquinas, por sí mismas, no sirven para mucho si no están dotadas de un gran software y de herramientas versátiles para trabajar en sus diferentes entornos.

BIBLIOGRAFÍA

BROOKS, F. P. JR., (1995), *The Mythical Man-Month: Essays on Software Engineering*, Indiana: Addison Wesley.

CORBATÓ, F. J., MERWIN-DAGGETT, M., DALE, R. C. (1962), “An experimental time-sharing system”, *Spring Joint Computer Conference*, pp. 335-344.

RITCHIE, D. M., THOMPSON, K. (1974), “The Unix Time- Sharing System”, *Communications of the ACM*, vol. 17, nº 7, pp. 365-374.

SILBERSCHATZ, A., GAGNE, G., GALVIN, P. B. (2006), *Fundamentos de Sistemas Operativos*, Madrid: McGraw-Hill

TANENBAUM, A. S. (2009), *Sistemas Operativos Modernos*, México: Pearson Educación.

Jorge González Mollá
Universitat Politècnica de València

Resumen: El entretenimiento se revela como una actividad de carácter intrínseco a la naturaleza humana, inherente a todas las civilizaciones y culturas conocidas. Desde el nacimiento de la informática, y prácticamente a la par, surgió la idea de utilizar una computadora para poder jugar con ella. En este trabajo se hace un recorrido cronológico de los hechos y circunstancias históricas que motivaron la aparición y el desarrollo de los videojuegos como una nueva forma de cultura, poniendo de relieve aspectos tan controvertidos como la falta de consenso sobre cuál fue el primer videojuego, y de cómo su industria ha ido desarrollándose paulatinamente con el tiempo para acabar convertida en uno de los sectores tecnológicos más productivos a nivel mundial, comercializando productos interdisciplinarios que nos permiten sentir experiencias muy diversas.

INTRODUCCIÓN

Mi madre juega al *Candy Crush*. No parece algo que tenga que sorprendernos, casi podría decirse que hoy en día todo el mundo juega al *Candy Crush*, del mismo modo que hace un año todo hijo de vecino jugaba al Apalabrados. Y sin embargo, aunque seguramente hablar de “todo el mundo” sea una exageración, en realidad, cada vez que hiciéramos una afirmación de este tipo, estaríamos exagerando menos. Desde que, hacia finales del siglo XX, los videojuegos aparecieran en la sociedad como una nueva forma de cultura para toda una generación, su número de usuarios ha ido incrementándose a lo largo de los años, consiguiendo abarcar con el tiempo a un mayor porcentaje de la población. La gota que terminó desbordando el vaso fue la popularización masiva de los teléfonos móviles, y especialmente su transformación en teléfonos inteligentes o *smartphones*, lo que los convirtió en ordenadores portátiles a todos los efectos. Era cuestión de tiempo que los videojuegos abandonaran la etiqueta de dirigidos a un público joven para pasar a formar parte de toda la sociedad en su conjunto. Mi madre tiene 60 años y como muchas otras personas de su edad ha sido ahora cuando ha tenido su primer contacto con el mundo del videojuego, pero por las notificaciones en forma de invitación que recibo a diario, no me cabe la menor duda de que su relación va para largo.

Este artículo se centra en el papel que el ocio representa para el conjunto de la sociedad, destacando como una actividad necesaria de carácter intrínseco a la

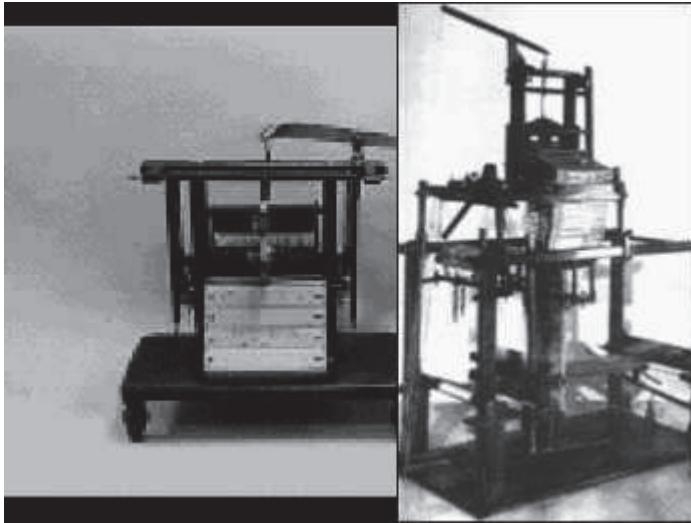
naturaleza del ser humano, y de cómo su encuentro con la informática permitió crear una nueva forma de entretenimiento. Analizaremos, por un lado, los primeros juguetes de la historia de tipo mecánico y/o electrónico, y por otro, hablaremos de los videojuegos con perspectiva histórica, de cómo fue su nacimiento, de su devenir en paralelo al desarrollo de la informática, de cómo se creó una industria al efecto, de la diversidad de plataformas y géneros existentes, y de cualquier otro aspecto sociocultural que pueda poner de manifiesto la trascendencia que tienen los videojuegos en la sociedad actual.

EL HOMBRE JUEGA (Y LA MUJER TAMBIÉN)

El término *Homo Ludens* se introdujo en 1938 como un guiño evolutivo al de *Homo Sapiens*, abanderando que la capacidad de jugar de los seres humanos es un eslabón más en su evolución. Según el autor de este nuevo término antropológico, Johan Huizinga, el juego se define como “[...] una acción que se desarrolla dentro de ciertos límites de lugar, de tiempo y de voluntad, siguiendo ciertas reglas libremente consentidas, y por fuera de lo que podría considerarse como de una utilidad o necesidad inmediata. Durante el juego reina el entusiasmo y la emotividad, ya sea que se trate de una simple fiesta, de un momento de diversión, o de una instancia más orientada a la competencia. La acción por momentos se acompaña de tensión, aunque también conlleva alegría y distensión.”

La arqueología sitúa la existencia de juguetes, objetos usados expresamente con fines lúdicos, en la antigua Mesopotamia, hace más de 5.000 años, y desde entonces al menos hay evidencias de que el juego es una actividad inherente a todas las civilizaciones y culturas conocidas. Introducidos en la niñez, los juguetes permiten el aprendizaje y el acercamiento a la sociedad mediante la imitación o la identificación, fomentando a la vez habilidades y estrategias sociales.

Los orígenes del videojuego se remontan a los primeros juguetes mecánicos y electromecánicos, máquinas con las que el usuario disfruta resolviendo los distintos retos de destreza y habilidad que se le proponen. Su primer precedente lo podemos encontrar a principios del siglo XIX, cuando a partir de la invención de las plantillas perforadas por parte de Joseph Marie Jacquard, se popularizaron diversos artilugios de este tipo tales como pianolas y otros juguetes mecánicos. El invento de Jacquard, cuya aplicación inicial había surgido en el contexto de la industria textil (programar las puntadas del tejido logrando así obtener una diversidad de tramas y figuras) había dado un giro de 180 grados hacia otros campos para los que no se había pensado en un principio.



Fuente: Wikipedia

Figura 5.1. Aplicación original sobre maquinaria textil de la invención del mecanismo de plantillas perforadas

A principios del siglo XX, la electrónica también entra en contacto con el universo del juguete, y así los trenes eléctricos y posteriormente los coches teledirigidos no tardaron en popularizarse, sentando las bases de una de las características esenciales en los videojuegos: el control remoto. Los juegos electromecánicos terminan convirtiéndose, merced a la informática, en videojuegos donde todo es posible, desde la imaginación de mundos complejos a los que poder transportarse a la personificación del usuario en los más variopintos personajes con los que identificarse, logrando así una experiencia de juego lo más entretenida y satisfactoria posible.

AÑOS 1940 Y 1950: PRIMEROS PASOS

Los videojuegos dieron sus primeras señales de vida hacia finales de la década de los 40, concretamente tras la Segunda Guerra Mundial, cuando las principales potencias vencedoras iniciaron una carrera tecnológica para construir las primeras supercomputadoras programables. El 25 de enero de 1947, Thomas T. Goldsmith y Estle Ray Mann patentaron un sistema electrónico llamado “*Cathode-ray tube amusement device*” (Dispositivo de Entretenimiento para Tubos de Rayos Catódicos) basado en los radares utilizados en la Segunda Guerra Mundial, convirtiéndose en el primer “programa” desarrollado para pantallas de tubos de rayos catódicos. Por desgracia, al tratarse solo de un experimento, no queda ninguna constancia real del mismo, aunque gracias a las descripciones y anotaciones de la patente original podemos saber cómo era.

Este dispositivo era un simulador interactivo de misiles en el que debíamos ajustar la curva y la velocidad de los misiles para impactar en unos objetivos virtuales. Sin duda su mecánica jugable lo acerca a lo que sería un videojuego, pero no se le puede considerar como tal porque no mostraba ningún gráfico en pantalla, los objetivos a derribar en realidad estaban dibujados en una lámina semitransparente que debíamos colocar encima de la pantalla y así aparecían sobre impresionados en dicha pantalla. Esto provocaba que tan solo se podía ver cómo el misil pasaba por el objetivo, por lo que en realidad no había ningún tipo de impacto, era el propio jugador el que debía mirar si lo había tocado o no.

Durante la Segunda Guerra Mundial, el prestigioso matemático británico Alan Turing y el experto en computación estadounidense Claude Shannon trabajaron juntos descifrando los códigos secretos del ejército nazi con la máquina Enigma y los codificadores de teletipos FISH. Fruto de esta unión, Turing y Shannon junto a otros matemáticos establecieron las bases de la teoría de la computación, donde señalaban la inteligencia artificial como el campo más importante hacia el que había que dirigir todos los esfuerzos de investigación en un futuro. Gracias a estas investigaciones en el campo de la inteligencia artificial, en marzo de 1950 Claude Shannon publicaba junto al economista y matemático D. G. Champernowne un artículo titulado “Programming a Computer for Playing Chess” en la revista *Philosophical Magazine*, especificando las primeras técnicas y algoritmos necesarios para crear un programa de ajedrez. Como por aquel entonces no había ordenadores lo bastante potentes como para poder ejecutarlo, hubo que esperar hasta finales de 1951 para que una de las primeras computadoras electrónicas comerciales, la *Ferranti Mark I*, permitiera implementarlo por simulación y sin usar gráficos de ningún tipo.

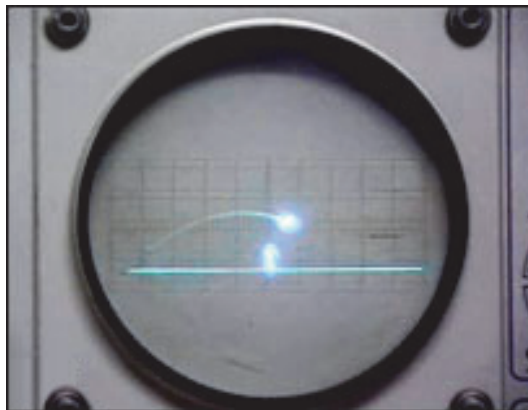
En su día de estreno, la máquina perdió su primera partida frente a un amigo de Alan Turing, pero ganó la segunda frente a la esposa de Champernowne. A casi medio siglo de distancia, Shannon y Champernowne sentaron las bases de los programas de ajedrez actuales, incluso los de las máquinas ajedrecistas de IBM, encabezadas por *Deep Blue* de finales de siglo. Las técnicas empleadas siguen siendo vigentes hoy en día, aunque con la diferencia de que a mediados del siglo XX el ordenador podía tardar entre 15 y 20 minutos para procesar cada movimiento. Al simular las partidas de ajedrez sin gráficos no se le suele considerar un videojuego como tal, aunque también existe mucha controversia al respecto, no en vano se trata de una tecnología relativamente joven, sin un consenso firme aún sobre cuál es el primer videojuego de la historia.

El 5 de mayo de 1951, John Bennett de *Ferranti* presentaba en el *Festival of Britain* la Nimrod, una enorme computadora capaz de jugar al *Nim*, un juego matemático original de la China. Valiéndose de un panel de luces, podíamos jugar contra la máquina usando dos modos de juego. Al carecer de pantalla, se le considera más bien como un juego electrónico y no un videojuego, pero la verdad es que fue la primera computadora diseñada específicamente para jugar un juego. La máquina fue todo un éxito, pero fue desmontada para utilizar sus piezas en otros proyectos, no obstante existe una réplica (a menor escala) en el *Computer Game Museum* de Berlín.

En 1952, Alexander “Sandy” Shafte Douglas presenta su tesis de doctorado en matemáticas en la Universidad de Cambridge basada en la interactividad entre seres humanos y computadoras. Dicho estudio incluía una versión computerizada del tres en raya llamada OXO, que permitía a una persona jugar contra la máquina EDSAC (*Electronic Delay Storage Automatic Calculator*), la primera computadora capaz de almacenar programas electrónicos. OXO no tuvo ningún tipo de popularidad ni repercusión mediática alguna porque aparte de ser una tesis de investigación, solo se podía jugar en la EDSAC, en Cambridge, pues era la única que existía en todo el mundo.

Es el primer juego de computadora en usar una pantalla gráfica digital, de ese modo tiene todos los requisitos para poder considerarse como el primer videojuego de la historia: una máquina (EDSAC), una interfaz (se jugaba mediante un dial de teléfono), una pantalla y un programa. Sin embargo, hay algunas voces discordantes con dicha afirmación porque prefieren considerarlo como un programa gráfico por ordenador porque carece de vídeo en movimiento. Un argumento más que discutible, sobre todo si tenemos en cuenta que hay videojuegos como por ejemplo las aventuras conversacionales sin ningún tipo de movimiento en pantalla y no por ello dejan de ser llamados videojuegos, ¿por qué entonces OXO debería ser una excepción? Quizá su origen académico también contribuyera a que finalmente OXO haya quedado relegado frente al que se considera oficialmente como primer videojuego de la historia, el *Tennis for Two*.

Fue el 18 de octubre del año 1958 cuando William Higinbotham, un físico estadounidense que participó en el *Proyecto Manhattan* (nombre en clave para el desarrollo de la bomba atómica), presentó un programa para jugar al tenis en un osciloscopio. A diferencia de la mayoría de videojuegos de tenis modernos, el *Tennis for Two* usaba una perspectiva lateral a pie de pista, donde se veía una línea horizontal para representar el terreno de juego y una línea vertical y pequeña en el centro del campo que hacía las veces de red (Figura 5.2). Además, ya contemplaba el efecto físico de la gravedad, un recurso muy adelantado a su época.



Fuente: pongmuseum.com

Figura 5.2. Tennis for Two, considerado como el primer videojuego de la historia

A pesar de las colas que se formaron el día de su presentación y las buenas críticas recibidas, *Tennis for Two* fue creado como un simple experimento para entretener a los visitantes del laboratorio en el que trabajaba Higinbotham, por lo que un año más tarde fue desmantelado para poder reutilizar sus piezas en otros proyectos. En el año 2008, con motivo de su 50º aniversario, fue reconstruido usando el diseño original y los componentes de la época. Como detalle curioso, la reconstrucción tardó más de 3 meses en poder completarse por la dificultad de encontrar piezas originales de entonces, mientras que el equipo de 1958 se realizó en menos de 3 semanas.

Al contrario de la creencia popular, el primer videojuego no fue el *Pong* de Atari, videojuego que no saldría hasta 1972, 20 años más tarde que OXO y 14 años más tarde que *Tennis for Two*. De hecho, antes de la aparición del conocido *Pong* todavía quedaban muchos más pasos por dar.

DÉCADA DE 1960: EL VIDEOJUEGO SE ESTABLECE FIRMEMENTE

La década anterior había sido una época de falsos comienzos para los videojuegos. Casi todos los que habían explorado la idea la abandonaron convencidos de que era una pérdida de tiempo. El ajedrez por ordenador era un campo de investigación fructífero pero en cambio permanecía oculto dentro del ámbito académico, lejos aún de su aplicación en el campo del entretenimiento.

En el *Massachusetts Institute of Technology* (MIT) existía un club de estudiantes especializado en el modelismo y la construcción de ferrocarriles a escala, el *Tech Model Railroad Club*. Muchos de sus miembros compartían también su pasión por los ordenadores y la ciencia ficción, lo que motivó que, tras la llegada del primer PDP-1 al MIT, tres de ellos, Wayne Wiatenem, Martin Graetz y Steve Russell, se embarcaran en programarlo con un juego de naves espaciales. *Spacewar!*, así se le llamaría al programa, vio la luz en 1962 y ocupaba 9 kilobytes de memoria. El juego tuvo tanto éxito entre el personal del MIT que acabó distribuyéndose a través de ARPANET para demostrar las capacidades del nuevo PDP-1, donde al final se incluiría de serie.

En *Spacewar!*, dos naves espaciales tratan de destruirse mutuamente mientras evitan la fuerza gravitatoria de una estrella. Cada jugador controla a una de las naves y puede disparar misiles cuya trayectoria se verá afectada también por el campo de fuerza de la estrella (Figura 5.3). A pesar de su éxito, los jóvenes estudiantes no patentaron su trabajo, ni tampoco se plantearon su comercialización, ya que dependía de una máquina, el PDP-1, que costaba 120.000 dólares. Aun así, el juego terminó siendo una de las ideas más copiadas en la historia de los videojuegos.



Fuente: Wikipedia. www.media.mit.edu

Figura 5.3. Spacewar!, dos naves tratando de destruirse mutuamente con una estrella como testigo

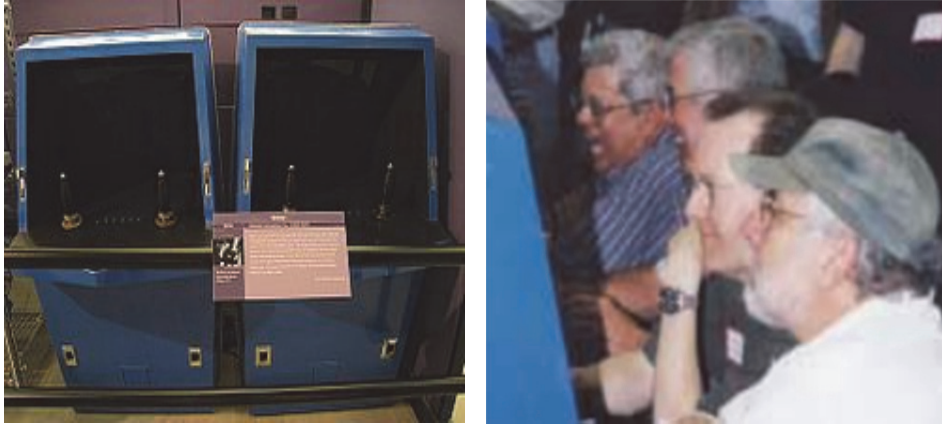
En 1966, Ralph Baer reconsideró una idea que había abandonado unos años antes: un dispositivo que, conectado a un televisor, permitiese convertirlo en una máquina de videojuegos. No en vano ya en 1950 el 90% de los norteamericanos disponían de un televisor en sus hogares, un número que contrastaba fuertemente con el escueto 9% contabilizado en la década anterior. Era lógico pensar que estos electrodomésticos pudieran servir para algo más que para ver la tele. Baer había trabajado anteriormente como técnico de televisión, y por tanto estaba familiarizado con unas herramientas que dibujaban líneas y patrones de colores que se movían por la pantalla.

Baer comenzó a diseñar su máquina y en 1967 ya tenía un primer prototipo que incorporaba una serie de juegos, entre los que estaban el ping-pong y un juego cooperativo de dos jugadores en el que ambos debían acorralar al contrario. Baer también diseñó un periférico *GunStick* (pistola electrónica) que, conectado al dispositivo, permitía disparar a un conjunto de objetivos. Su *Brown Box* (así la llamaba Baer) atrajo la atención de una compañía de televisión por cable, pero tras unas intensas negociaciones que duraron varios meses no se llegó a ningún acuerdo, por lo que el invento de Baer tuvo que esperar unos cuantos años más para finalmente ver la luz.

1970-1975: EL NACIMIENTO DE UNA INDUSTRIA

En 1971, Bill Pits, un estudiante de la Universidad de Stanford, y un amigo suyo, Hugh Tuck, construyeron una versión de *Spacewar!* que funcionaba con monedas para su explotación en los salones recreativos de la época, que hasta entonces solo contenían máquinas electromecánicas. Con cambio de nombre incluido, *Galaxy Game* no resultó rentable a 10 céntimos por partida, por lo que decidieron que debían dar servicio a 8 máquinas a la vez con un mismo computador. La máquina se instaló en junio de 1972

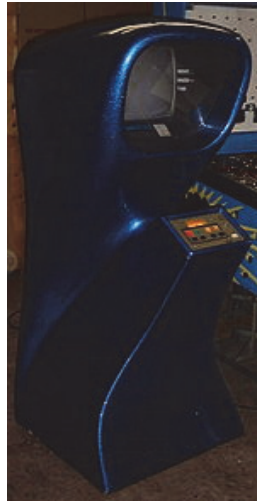
cerca de la Universidad de Stanford, y allí permaneció con bastante éxito hasta 1979, cuando fue desensamblada y almacenada en una vulgar oficina. De 1997 a 2000 se expuso en la Universidad de Stanford, y desde entonces se exhibe en el *Computer Museum History Center* en California (Figura 5.4).



Fuente: Wikipedia, Tom Purves

Figura 5.4. (Izq.) Dos máquinas arcade de Galaxy Game controladas por un mismo computador PDP-11. (Der.) Slug Russell, Bill Pitts, Steve Golson, and Nolan Bushnell (de izquierda a derecha) jugando a Galaxy Game

Nolan Bushnell también había tenido la gran idea de llevar *Spacewar!* a los salones recreativos pero tras el fracaso de su prototipo inicial, había decidido prescindir de los computadores. Junto a Ted Dabney, compañero de trabajo, construyó un nuevo equipo sin CPU, a partir de componentes discretos hechos por ellos mismos y de un televisor en blanco y negro. Ya en el verano de 1971, Bushnell y Dabney se habían asociado bajo lo que acabaría siendo Atari, y en noviembre –dos meses después del primer *Galaxy Game*–, la máquina *Computer Space* (Figura 5.5) se instaló cerca de la Universidad de Stanford con un gran éxito entre los estudiantes. Sin embargo, por culpa de su sistema de control, fue un fracaso entre el público no universitario.



Fuente: Wikipedia

Figura 5.5. Computer Space, primera incursión de Atari en el mundo arcade

Ralph Baer no desistió en sus empeños y finalmente en 1971 llegó a un acuerdo con Magnavox.

Un año después presentaba públicamente su nueva y rebautizada *Magnavox Odyssey* (Figura 5.6). Al mismo tiempo se presentaba el primer accesorio de la máquina, un rifle de plástico *GunStick* y diez juegos adicionales, que se vendían por separado. Nótese que ya en 1972 salía al mercado una consola de videojuegos que, sin entrar en comparaciones, es tal y como las de la actualidad. Varios errores de *marketing* impidieron lo que sin duda habría sido un *hit* en volumen de ventas, pero sí vendieron lo bastante como para atraer la atención de Nolan Bushnell, fundador de Atari.

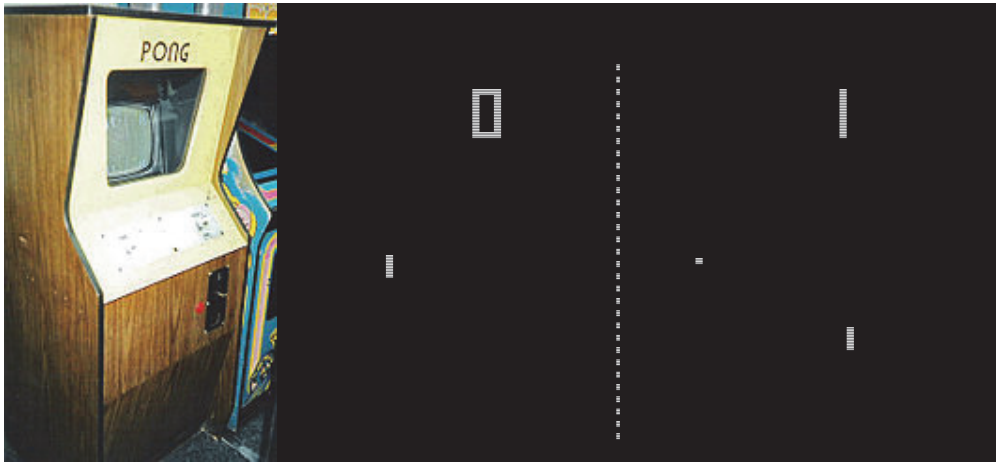


Fuente: Wikipedia, Marty Goldberg

Figura 5.6. Magnavox Odyssey, la primera videoconsola doméstica de la historia

En 1972, Atari tomaba el ping-pong, uno de los juegos de serie de la nueva *Magnavox Odyssey*, y construía una versión arcade del mismo, con algunas mejoras, que recibió el nombre de *Pong*.

La idea resultó ser un éxito rotundo, de repente el país se inundó de máquinas *Pong* (Figura 5.7), así como de copias fabricadas por compañías de la competencia. Taito lanzó al mercado oriental su propia versión del juego, y lo mismo ocurrió en países de Europa tales como Francia o Italia. En 1974 había casi 100.000 *Pongs* en EEUU generando unos 250 millones de dólares anuales. Un año después llegó la primera consola doméstica de Atari, permitiendo jugar al *Pong* en casa.



Fuente: Wikipedia

Figura 5.7. Pong, una máquina arcade que supuso el primer videojuego que lanzó Atari

Mientras Atari comercializaba nuevos títulos que en la práctica inauguraban un nuevo género, sus competidoras, compañías creadas a la luz de su éxito, solo sacaban nuevas copias de *Pong*. El negocio del entretenimiento se reestructuró por completo. Los videojuegos eran más rentables que los juegos electromecánicos de la época, más frágiles y más costosos de mantener. Con Atari había surgido un nuevo modelo de negocio dentro de la industria del entretenimiento. Magnavox ganaría más tarde la demanda que interpuso contra Atari sobre los derechos de *Pong*, por haberse basado descaradamente en el juego Ping-pong incluido con la *Magnavox Odyssey*, pero para entonces Atari ya se había convertido en la referencia a seguir de esta nueva industria.

1975-1980: LOS VIDEOJUEGOS APUESTAN POR EL MICROPROCESADOR

Sería también Atari la que sacaría a la luz *Breakout*, otro gran éxito inspirado en el propio Pong. El juego, lanzado en 1976 con un éxito inmediato, sentaría cátedra como “machacaladrillos”, portándose o imitándose desde entonces a prácticamente todos los sistemas de videojuegos. Curiosamente, fueron los conocidos Steve Jobs y Steve Wozniak quienes se encargaron originalmente de la primera versión de *Breakout*, aunque finalmente Atari se decantaría por otra.

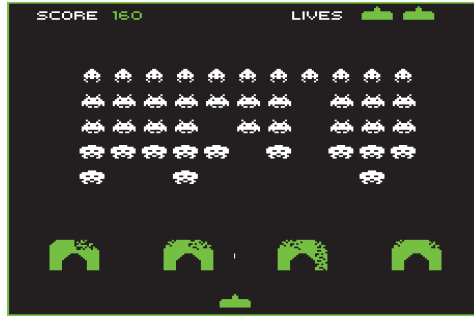
Con *Death Race*, el primero en causar polémica por sus contenidos presuntamente inadecuados, Atari y el resto de compañías recreativas ponían fin a su tecnología basada en circuiterías TTL. La aparición de un nuevo tipo de microchip en el mercado –la familia de procesadores Intel 8080– supondría en la práctica una nueva revolución en la incipiente industria de los videojuegos. El año 1975 vería nacer a *Spirit of 76*, un *pinball* que surgía como abanderado de la tecnología digital, así como a *Gun Fight* de Midway, la primera máquina recreativa basada en un microprocesador.

El éxito de los videojuegos en los salones recreativos estimuló su aparición en los primeros ordenadores personales de la época, pese a que todavía eran un lujo al alcance de muy pocos. Además, esta primera generación de ordenadores personales presentaba muchas limitaciones como plataforma de videojuegos, dejando de lado toda posibilidad de jugar a juegos de acción. Esto provocaría la aparición de los videojuegos conversacionales y los de estrategia por turnos, tales como el *Adventure* de Will Crowther, donde el jugador, usando frases en lenguaje natural, tenía que resolver una serie de acertijos y misterios para salir con vida del interior de una cueva. *Adventure* (1975) no fue el primer juego conversacional de la historia, pero sí el más influyente. Prueba de ello es que sirvió de inspiración al matrimonio formado por Ken y Roberta Williams, futuros fundadores de Sierra Entertainment, en el desarrollo de su *Mystery House* para Apple II. *Tanktics*, en 1977, inauguraría el género de los *wargames*, y *Alakabeth*, de Richard Garriot, haría lo propio con los juegos de rol (*Role Play Games*) en 1979.

Entre 1976 y 1978, las consolas domésticas también se abrirían a la tecnología microprocesador, y así Atari, Magnavox y otras compañías rivales pondrían a la venta sus nuevas videoconsolas con un catálogo de cartuchos de juegos que se compraban separadamente de la unidad principal. Se trata de la Fairchild Channel F, la RCA Studio II, la Atari 2600, y la Magnavox Odyssey II. Paralelamente, Mattel lanzaría en 1976 *Auto Race*, la primera consola portátil de la historia, mientras Ralph Baer revolucionaría de nuevo el mercado en 1978 con su famoso *Simon* de MB.

Por otro lado, Taito se estrena en la tecnología microprocesador con *Space Invaders* (Figura 5.8), un juego que originalmente consistía en disparar contra tanques y aeroplanos, pero que debido al éxito de la película *Star Wars*, terminaría por adoptar su forma definitiva de batalla espacial. El juego obtuvo inmediatamente un éxito de dimensiones

descomunales, fue convertido a todos los formatos importantes de la época y dio lugar a numerosas continuaciones e infinitos clones. No solo inauguró el género de los "matamarcianos", sino que además situó a la industria japonesa en el lugar que le correspondía e impulsó la fiebre por los videojuegos a nivel mundial.



Fuente: Wikipedia

Figura 5.8. Space Invaders, el primer videojuego arcade de Taito basado en tecnología microprocesador

DÉCADAS DE 1980 Y 1990: CRECIMIENTO EXPONENCIAL

Desde la irrupción de *Space Invaders* la fiebre por los videojuegos aumentó considerablemente, los ingresos crecían mes a mes y las máquinas recreativas se habían instalado en todas partes. Las más importantes compañías dedicadas al negocio del entretenimiento habían creado sus divisiones de videojuegos, como sucedía con LucasFilm, Walt Disney o 20th Century Fox, incluso McDonald's se había asociado con Atari. La industria crecía y se veía favorecida además por un rendimiento audiovisual en aumento gracias a las nuevas generaciones de computadoras. El éxito de los "matamarcianos" se combinó con las nuevas tecnologías de gráficos vectoriales en títulos tales como *Galaga* (1981), conocido sucesor de *Galaxian* (1979), o *Defender* (1981), el primer "matamarcianos" horizontal. Pero todos acabarían rindiéndose ante *Pac-Man* (1980), el videojuego más popular de todos los tiempos y el primero en crear su *merchandising* paralelo.

Mientras Atari empezaba a flaquear y otras compañías desarrollaban videojuegos para su 2600, Europa apenas había coqueteado con diversos clones de *Pong* cuando aterrizó el *Sinclair ZX80*. El nuevo ordenador daba paso a la era de los 8 bits, caracterizada por unos precios reducidos que supondrían el espaldarazo definitivo para la penetración de los ordenadores en los hogares. Al *ZX80* le seguirían el *ZX81*, el *ZX Spectrum*, el *Commodore 64*, el *Amstrad CPC*, o los *MSX*. El mercado de videojuegos de la década de 1980 ya era desbordante y se caracterizaba por la atmósfera de creatividad que reinaba en una industria que todavía estaba dando sus primeros pasos: *Football Manager* (1982), *Manic Miner* (1983), *Knight Lore* (1983), o *Tetris* (1984), son solo algunos de los

primeros títulos de lo que acabó siendo un torrente desbordante de imaginación. Nintendo daría un giro trascendental a su empresa con sus “maquinitas” *Game&Watch* de LCD, y sobre todo con su consola *NES* (*Nintendo Entertainment System*, 1983) que arrasó con títulos como *Donkey Kong*, *Super Mario Bros*, *Dragon Quest*, *The Legend of Zelda* o *Final Fantasy*. Estamos ante el apogeo de los videojuegos domésticos sin perjuicio de las máquinas recreativas, que seguían en alza debido a la mayor calidad de sus videojuegos con compañías tales como Capcom (*Commando*, *Ghosts'n Goblins*, *Street Fighter*), Konami (*Contra*, *Combat School*), Irem (*R-Type*), SNK (*Ikari Warriors*, *Metal Slug*), o Sega (*Altered Beast*, *Out Run*, *Golden Axe*).

La década de 1990 se caracterizó por el declive definitivo de los ordenadores de 8 bits y el esplendor de una nueva generación de computadoras más avanzadas con arquitectura de 16 bits. Ordenadores como el *Commodore Amiga*, el *Atari ST* o la familia *IBM PC* y compatibles, abarcarían un mercado que finalmente acapararía el PC y que se mantiene hasta la actualidad. Juegos como *Defender of the Crown* (1986), *Another World* (1991), o aventuras gráficas como *Maniac Mansion* (1987), recogían la influencia de las técnicas cinematográficas en su desarrollo que más adelante explotarían con la aparición del CDROM y los contenidos de tipo multimedia como en *SimCity* (1989), o *Civilization* (1991), redefiniendo el género de estrategia por turnos. En el ámbito de las consolas, Sega surgió como rival de Nintendo con su *Master System* (1985) y sobre todo con su *MegaDrive* (1988), con títulos tan emblemáticos como *Sonic the Hedgehog*, pero Nintendo contrataría con la *Super Nintendo* (1991) y sobre todo con la *GameBoy* (1989), la que fue la consola portátil líder de la época, por encima de las que luego sacaron Atari y Sega.

El período culmina con la llegada de la tecnología 3D a los ordenadores, con juegos como *Alone in The Dark* (1992), *Doom* (1993), o *Tomb Raider* (1996), y la capacidad de jugar en red. Las arquitecturas de 32 y 64 bits van tomando un protagonismo cada vez mayor, y poco a poco los videojuegos domésticos van superando tecnológicamente a la inmensa mayoría de arcades, logrando barrer prácticamente del mercado el modelo de negocio de las máquinas recreativas. Sony y su *PlayStation* (1994) tomarían el relevo de Sega en su particular pugna con Nintendo, eclipsando la salida de la *Sega Saturn* (1994) y su título estrella, *Virtua Fighter*, de lucha en 3D. Nintendo lo intentaría con su *Nintendo 64* (1996), y Sega se apagaría con su *DreamCast* (1998), pero para entonces *PlayStation* ya se había convertido en la consola de mayor éxito del mundo, con ejemplares tan emblemáticos como *Gran Turismo* (1997), *FIFA* (1998), o *Silent Hill* (1999).

SIGLO XXI: PROFESIONALIZACIÓN Y EXPANSIÓN DE LA INDUSTRIA

Si hay algo que caracteriza a la industria del videojuego del siglo XXI es su transformación en una industria multimillonaria de dimensiones descomunales, casi inimaginable pocos años atrás. Empresas como Microsoft, con su *X-BOX* (2001), se

lanzarían al mercado de las videoconsolas, un año después de que la *PlayStation 2* de Sony, de 128 bits, batiera todos los récords de ventas.

Nintendo, Microsoft y Sony continuarían su particular brega por el mercado de las consolas, dotando a sus nuevos modelos, tanto portátiles como de sobremesa, de tecnologías tan innovadoras como pantallas táctiles o sistemas de control por movimiento. De esta época son originales *Los Sims* (2000), *Grand Theft Auto III* (2001), *Call of Duty* (2003), y *SingStar* (2004).

El crecimiento en el sector de la telefonía móvil, sobre todo desde la llegada del *iPhone* (2007), permitió la instalación de videojuegos en los *smartphones* (casi más ordenadores que teléfonos), suponiendo el zarpazo definitivo para seducir al resto de la sociedad hasta entonces impasible. *Angry Birds* (2009) o *Candy Crush Saga* (2012) son algunos ejemplos que hablan por sí solos. Medio siglo después, el concepto de videojuego continúa reinventándose día a día, y lejos de haber alcanzado su madurez creativa, los videojuegos siguen siendo una nueva forma de arte que parece estar dando aún sus primeros pasos.

BIBLIOGRAFÍA

DONOVAN, T. (2010), *Replay, The History of Video Games*, Yellow Ant.

DEMARIA, R.; WILSON, J. L. (2003), *High score!: the illustrated history of electronic games*, McGraw-Hill Osborne Media.

LOGUIDICE, B. (2009), *Vintage Games: An Insider Look at the History of Grand Theft Auto, Super Mario, and the Most Influential Games of All Time*, Focal Press.

EGENFIELD-NIELSEN, S.; SMITH, J. H.; PAJARES, S. (2008), *Understanding Video Games*, Routledge.

WOLF, MARK J. P., PERRON, B. (2008), *The Video Game Theory Reader 2*, Routledge.

Vicente Santonja Gisbert
Universitat Politècnica de València

Resumen: Internet, como ocurre con todas las grandes obras de ingeniería, es un logro colectivo al que han contribuido numerosos científicos, ingenieros y gestores. Es justo destacar las aportaciones más significativas de aquellos que determinaron la trayectoria a seguir en el diseño de esta red de redes global. Pero sería injusto olvidar que los que definieron estas grandes líneas, lo hicieron apoyándose en el trabajo anónimo de otros muchos que allanaron el camino y facilitaron la labor de los primeros. En este artículo recordamos algunos de los hitos más importantes que llevaron al diseño de la Internet actual.

PRECEDENTES

En 1908, en un artículo visionario, Nikola Tesla afirmaba “Pronto será posible que un hombre de negocios en Nueva York dicte instrucciones y que al instante aparezcan en su oficina en Londres o en otro lugar... Un instrumento barato, no más grande que un reloj, permitirá a su portador escuchar en cualquier lugar, en mar o tierra, la música o la canción, la voz de un líder político, la dirección de un eminente hombre de ciencia, o el sermón de un clérigo elocuente, generado en otro lugar, por muy distante que sea. De la misma manera, cualquier imagen, carácter, dibujo o impresión se podrá transferir de un lugar a otro”.

La visión de Nikola Tesla se haría realidad unas decenas de años después, gracias a las contribuciones de muchos investigadores procedentes de todo el mundo, en su mayoría pertenecientes a universidades y centros de investigación. La historia de cómo esta predicción se convirtió en realidad no es otra que la historia de Internet: uno de los sistemas de ingeniería más complejos desarrollados por la humanidad.

El punto de partida de este desarrollo se puede datar el 4 de octubre de 1957, cuando la Unión Soviética consigue poner en órbita el primer satélite artificial: *Sputnik 1*. La reacción de los Estados Unidos no se hizo esperar. En febrero de 1958 el presidente norteamericano Eisenhower anunció la creación de la *Advanced Research Projects Agency* (ARPA). Según el propio documento fundacional, la misión de ARPA era evitar en el futuro la “sorpresa tecnológica” que supuso el lanzamiento del *Sputnik*, que puso en evidencia la superioridad de los soviéticos en la llamada “carrera espacial”. La *Information Processing Techniques Office* (IPTO) formaba parte de ARPA y estaba

especializada en sistemas de computación. El primer director de IPTO, nombrado en octubre de 1962, fue J.C.R. Licklider, que provenía del *Massachusetts Institute of Technology* (MIT). Previamente, en agosto de ese mismo año, Licklider había plasmado sus ideas, avanzadas a su tiempo, sobre lo que él denominaba de forma “intencionadamente grandilocuente” la *Intergalactic Computer Network*. Se trataría de un conjunto de computadores interconectados que permitirían el acceso universal a programas y datos. Fue él quien impulsó la investigación en redes como una de las prioridades de IPTO.

En julio de 1961 Leonard Kleinrock, estudiante de doctorado en el MIT, publicó un artículo en el que demostraba que la conmutación de paquetes (o la conmutación de mensajes, según veremos a continuación) suponía una alternativa ventajosa a la conmutación de circuitos para la transmisión de datos. Hasta ese momento, la red de comunicación más extendida era la de telefonía, que se basaba en la conmutación de circuitos. Esta tecnología exige que, mediante el uso de conmutadores, se cree una línea dedicada entre los dos extremos de la comunicación. Esta línea se mantiene reservada mientras dura la comunicación, incluso en los periodos de silencio. Mediante el uso de la teoría de colas, Kleinrock demostró que la conmutación de paquetes (o mensajes) se adapta mejor al tráfico discontinuo y en ráfagas característico de la transmisión de datos. Para la conmutación de paquetes, el mensaje a transmitir se descompone en paquetes de una longitud fija. A cada paquete se le añade una cabecera con la dirección destino. Los paquetes viajan del emisor al receptor almacenándose y retransmitiéndose en diversos *routers*. En el caso de conmutación de mensajes, el funcionamiento es similar pero los mensajes se transmiten enteros, sin descomponerse en paquetes. En 1963 Kleinrock se traslada a la Universidad de California en Los Ángeles (UCLA) como profesor asistente y en 1964 publica un libro que contiene los fundamentos teóricos de las redes de comunicación de datos. A pesar del esfuerzo realizado por el propio Kleinrock para proponerse como uno de los *padres* de Internet, algunos autores afirman que las aportaciones de Kleinrock tratan principalmente de la conmutación de mensajes y no propiamente de la conmutación de paquetes, sobre la que se basa Internet. Según estos autores, ni la palabra “paquete” ni el concepto de división del mensaje en unidades de longitud fija para su transmisión aparecen explícitamente en los textos de Kleinrock.

A finales de 1966 Larry Roberts, también procedente del MIT, se incorporó al IPTO como gestor de programas. Más tarde, en 1969 sería nombrado director de esta oficina. Desde su llegada marcó como su principal prioridad el desarrollo de una red de computadores basada en la conmutación de paquetes: ARPANET. Roberts conocía los trabajos de Kleinrock, realizados mientras ambos eran estudiantes de doctorado en el MIT. En 1967, Roberts presentó su proyecto ARPANET en una conferencia en la que se exponía otro artículo que trataba sobre redes de conmutación de paquetes. Los autores de este artículo eran los ingleses Donald Davies y Roger Scantlebury del *National Physical Laboratory* (NPL). Al entrevistarse con estos autores, Roberts tuvo conocimiento de los trabajos de Paul Baran del *RAND Institute*, que estaba investigando el uso de la conmutación de paquetes para la transmisión de voz en aplicaciones militares desde

1964. Por tanto, fue Baran el primero en plantear dos ideas cruciales para la conmutación de paquetes: (1) la descentralización de las funciones de retransmisión y encaminamiento y (2) la división de los mensajes en paquetes (aunque la propuesta de la palabra “paquete” se debe a Davies). En definitiva, las aportaciones del RAND, NPL y MIT se produjeron de forma paralela e independiente, sin que unos investigadores conocieran el trabajo de los otros. Conjuntamente, forjaron los fundamentos de la actual red Internet.

El desarrollo del *router* (entonces denominado *Interface Message Processor*, IMP), elemento central en la red ARPANET, le fue encargado a la empresa *Bolt Beranek and Newman* (BBN). Licklider había trabajado en esta firma y en ella trabajaba Bob Kahn, cuyo papel en el desarrollo de los protocolos TCP/IP sería, como pronto veremos, fundamental. El grupo de Kleinrock en UCLA se encargó de realizar las mediciones del funcionamiento de la red. Por este motivo, este fue el lugar elegido como primer nodo de ARPANET y donde BBN instaló el primer IMP en septiembre de 1969. El segundo nodo se instaló en el *Stanford Research Institute* (SRI). A finales de 1969, la ARPANET estaba formada por cuatro nodos, al unirse a los dos citados, los de la Universidad de California en Santa Barbara y la Universidad de Utah.

El 29 de octubre de 1969 se produjo la primera transmisión desde un terminal de UCLA con destino a un computador del SRI en Palo Alto, a millas de distancia. El estudiante Charley Kline, bajo la supervisión de Kleinrock, se proponía transmitir la palabra “LOGIN”. Kline tecleó una “L” y una “O” que fueron correctamente recibidas. Pero cuando tecleó la “G” el sistema se colgó. Se había producido la primera transmisión por ARPANET con el texto “LO”.

29 OCT 69	2100	LOADED OP. PROGRAM FOR BEN BARKER BBN	CSK
	22:30	Talked to SRI Host to Host	CSK
		Left op. program running after sending a host dead message to imp.	CSK

Fuente: Wikipedia. Autor: Charles S Kline, en UCLA Kleinrock Center for Internet Studies
 Figura 6.1. Registro de la primera transmisión a través de ARPANET el 29 de octubre de 1969

En los años siguientes, un gran número de computadores se fueron conectando a ARPANET. En diciembre de 1970, el *Network Working Group* (NWG), dirigido por Steve Crocker, presentó el protocolo de control de la transmisión *Network Control Protocol*

(NCP). Una vez implementado sobre los computadores de ARPANET, comenzó el desarrollo de aplicaciones en red. En octubre de 1972, Robert Kahn, que se había incorporado al IPTO, organizó la primera demostración pública del funcionamiento de ARPANET conectando 20 computadores, dentro del marco de la *International Computer Communication Conference (ICCC)*. Esta demostración fue la prueba de fuego de que ARPANET se había convertido en una realidad.

DE ARPANET A INTERNET

Pero ARPANET aún no era Internet. Era una red de conmutación de paquetes. En esos primeros años de la década de los 70 se diseñaron otras redes basadas en conmutación de paquetes, que aunque eran similares a ARPANET no se podían interconectar con ella. Por ejemplo, en el *Palo Alto Research Center (PARC)* de Xerox, Bob Metcalfe desarrolló la red de área local que denominó “Ethernet” para interconectar las estaciones de trabajo de una oficina mediante una conexión de alta velocidad. Para el diseño de Ethernet, Metcalfe se había basado en la red inalámbrica ALOHAnet desarrollada en la Universidad de Hawái. También en San Francisco se desplegó una red de radio por paquetes conocida como PRNET, y una versión por satélite que se denominó SATNET. Estas redes de conmutación utilizaban mecanismos y formatos de paquetes incompatibles entre sí por lo que no era posible la interconexión.

Kahn decidió desarrollar una nueva versión del protocolo NCP adaptado a redes con *arquitectura abierta*. Esto significa que la tecnología de las redes individuales no está fijada por la arquitectura de red, sino que puede ser definida libremente por sus diseñadores según su ámbito de aplicación y los requisitos de sus usuarios. Una “arquitectura de interconexión de redes” definiría cómo se interconectan estas redes. A principios de 1973 Robert Kahn se propuso abordar el problema de la interconexión de las redes de conmutación de paquetes. Es decir, el diseño de una “internetwork” o simplemente “Internet”. Su propuesta se basaba en cuatro fundamentos básicos: **(i)** Internet está formada por la interconexión de diversas redes. Cada red individual tiene su modo de funcionamiento propio. No se requieren cambios internos en ninguna de estas redes para permitir su conexión a Internet. **(ii)** La interconexión de las redes se hace mediante *routers*. Para simplificar su diseño, los *routers* no almacenan información de estado de los flujos de paquetes que los atraviesan. **(iii)** La comunicación es del tipo *best effort*. No se garantiza la entrega de un paquete a su destino final. Si el paquete no pudiera ser entregado, se producirá una retransmisión desde la fuente. **(iv)** No existe un control global centralizado de funcionamiento de la red. En la primavera de 1973, tras iniciar el proyecto de interconexión de las diversas redes, Bob Kahn solicitó la colaboración de Vint Cerf que había participado en el diseño del NCP.

Cerf y Kahn desarrollaron el *Transmission Control Protocol (TCP)* que se encargaba de los servicios de transporte y de encaminamiento. En esta primera versión de TCP, el protocolo de transporte proporcionaba mecanismos para una entrega secuenciada to-

talmente fiable. Esta implementación resultó ser apropiada para la gran mayoría de aplicaciones que requieren una transmisión fiable, tales como la transferencia de ficheros y las sesiones de terminal remoto. Sin embargo, algunos experimentos iniciales con la transmisión de voz sugerían que, en ese caso, las pérdidas de paquetes no debían ser corregidas por TCP y que el control de la transmisión se podía gestionar mejor desde la propia aplicación. Es decir, la sobrecarga asociada a un transporte fiable no era adecuada para el tráfico de tiempo real. Por ello, la implementación original de TCP se desdobló en dos protocolos, apareciendo así el protocolo IP (*Internet Protocol*), encargado del direccionamiento y el encaminamiento de los paquetes. Es decir, de leer la dirección de cada paquete y hacer que llegue a su destino atravesando una secuencia de *routers*. Por encima de IP, se propuso un nuevo protocolo TCP que definía las funciones de supervisión de la comunicación, tales como el control de errores o el control de flujo. Este protocolo define cómo recomponer el mensaje a partir de los paquetes recibidos, para lo cual se debe restaurar el orden de los paquetes, comprobar si se ha perdido alguno y, en su caso, solicitar las oportunas retransmisiones. Para aquellas aplicaciones que no requieren esta supervisión, o que se encargan directamente de ella desde la propia aplicación, se introdujo un nuevo protocolo denominado *User Datagram Protocol* (UDP), que permite un acceso directo a la funcionalidad de IP. Esta implementación se conoce como *protocolos TCP/IP*. Kahn y Cerf publicaron un artículo titulado “A Protocol for Packet Network Interconnection”. Internet acababa de nacer.



Fuente: Fundación Princesa de Asturias, www.fpa.es

Figura 6.2. Vinton Cerf, Lawrence Roberts, Robert Kahn y Tim Berners-Lee en el acto de recepción del Premio Príncipe de Asturias de Investigación Científica y Técnica 2002

PRIMERAS APLICACIONES SOBRE INTERNET

La compartición de recursos, especialmente de los grandes computadores de tiempo compartido que se estaban instalando en universidades y centros de investigación, fue el motivo principal que impulsó el desarrollo tanto de ARPANET como de Internet. Se

buscaba, por ejemplo, que los usuarios conectados a una red de radio por paquetes pudieran acceder a un sistema de tiempo compartido conectado a ARPANET. Interconectar estas dos redes resultaba mucho más económico que duplicar los equipos. Por ello, las aplicaciones de transferencia de ficheros (basadas en el protocolo FTP) y las de terminal remoto (basadas en Telnet) fueron muy importantes en los primeros tiempos de Internet. Sin embargo, fue el correo electrónico el que experimentó un éxito enorme e introdujo un nuevo paradigma para el tipo de comunicaciones que se iban a producir en las nuevas redes. Se pasó de la comunicación entre máquinas a la comunicación entre personas. Todo un anticipo de los que más tarde sería una revolución social basada en el uso de Internet.

En realidad, el *e-mail* existía antes de Internet. Empezó en 1965 en los computadores de tiempo compartido como una aplicación que permitía que los usuarios del computador pudieran comunicarse entre sí. En marzo de 1972, Ray Tomlinson de BBN adaptó el correo electrónico para funcionar sobre ARPANET y desarrolló el software necesario para ponerlo en funcionamiento. En julio de ese año, Roberts desarrolló un software para escribir, listar, almacenar, leer, retransmitir y responder los mensajes. Esto facilitó la rápida expansión del correo electrónico.

A medida que el número de computadores conectados a Internet aumentaba, surgió la necesidad de organizar los ficheros e información accesible por la red y facilitar la localización de contenidos. Hubo diferentes enfoques para llevar a cabo esta tarea: la *FTP Archive list*, *WAIS (Wide Area Information Servers)* y *Gopher*. *WAIS* era una aplicación de búsqueda en Internet que permitía realizar una búsqueda en varias bases de datos a la vez. Se utilizaban bases de datos especializadas en diversos temas y que operaban en diversos servidores. Un directorio de servidores permitía al usuario determinar qué bases de datos atenderían la búsqueda. *Gopher* se convirtió en la primera interfaz usada en Internet para acceder a la información disponible en un servidor. *Gopher* proponía el uso de un conjunto de menús jerárquicos para clasificar la información contenida en el servidor. Este conjunto de menús llevaban en último término a un fichero de texto. No existían enlaces entre los ficheros. Aunque las opciones del menú *Gopher* eran ejemplos de hipertexto, éstas no fueron comúnmente percibidas de esta manera. Estos proyectos se quedaron cortos en poder alojar todos los tipos de datos existentes y en poder crecer sin cuellos de botella.

Por otro lado, el concepto de *hipertexto* se estaba utilizando con diversos fines. Hubo un primer precedente en la tecnología *Memex* propuesta por Vannevar Bush en 1945. Posteriormente, Ted Nelson en el proyecto *Xanadu* y Douglas Engelbart en el proyecto *NLS (oN-Line System)* desarrollaron la idea. Diversos sistemas de hipertexto, como el *HyperCard* de Apple Computer, estaban ya en funcionamiento cuando se propuso su uso en la *World Wide Web*.

LA APLICACIÓN QUE ABRIÓ INTERNET AL MUNDO: WORLD WIDE WEB

Tim Berners-Lee, con la colaboración de Robert Cailliau, desarrolló la *World Wide Web* (WWW) en los primeros años de la década de 1990. El CERN era el principal nodo de Internet en Europa. Además reunía un gran número de investigadores de diferentes países y una gran variedad de sistemas informáticos interconectados. Pero se echaba en falta un método eficiente para compartir la información entre científicos. Berners-Lee, que trabajaba en el CERN desde 1984, tenía una idea para llevar a cabo esa compartición. Propuso combinar dos tecnologías ya existentes: hipertexto e Internet. Berners-Lee presentó su idea en la *European Conference on Hypertext Technology* en septiembre de 1990. Ninguna empresa se interesó por su propuesta. Así que, a finales de 1990, Berners-Lee había desarrollado todas las piezas necesarias para poner en marcha la WWW: el protocolo *HyperText Transfer Protocol* (HTTP), el lenguaje *HyperText Markup Language* (HTML), el primer navegador Web, denominado *WorldWideWeb* (que era también un editor web) y el primer software servidor HTTP (*CERN httpd*) instalado sobre el primer servidor web físico (<http://info.cern.ch>). Este servidor contenía las primeras páginas web que consistían en una descripción del propio proyecto. El navegador y el servidor se desarrollaron sobre un computador NeXT, la empresa creada por Steve Jobs cuando fue expulsado de Apple. En agosto de 1991, Berners-Lee difundió un resumen del proyecto World Wide Web en el grupo de noticias *alt.hypertext*. Fue el primer anuncio público de la web. Paul Kunz del *Stanford Linear Accelerator Center* (SLAC) visitó el CERN en septiembre de 1991 y se entusiasmó con el proyecto WWW. De vuelta a Stanford adaptó el software de Berners-Lee para hacerlo funcionar sobre el sistema operativo VM/CMS en un *mainframe* IBM. De esta forma instaló el primer servidor web fuera de Europa y el primero de Estados Unidos. Este servidor contenía una base de datos con 300.000 artículos de investigación.



Fuente: www.w3.org Simposio The Future of the Web, 29 de octubre de 2014

Figura 6.3. Tim Berners-Lee y Vint Cerf

El navegador desarrollado por Berners-Lee tenía una interfaz gráfica pero solo funcionaba sobre computadores NeXT, que no estaban muy extendidos. La aparición del navegador *Mosaic* en 1993, desarrollado por Marc Andreessen y Eric Bina del *National Center for Supercomputing Applications* en la Universidad de Illinois en Urbana-Champaign (NCSA-UIUC), supuso un importante avance en la popularidad de la *World Wide Web*. Se trataba de un navegador web con soporte para gráficos y audio. Uno de los motivos del éxito de *Mosaic* fue la disponibilidad de versiones para Unix, Windows y Mac. Además era fácil de instalar y tenía un excelente soporte técnico (ofrecido por el propio Andreessen). Al finalizar sus estudios en la UIUC, Andreessen se unió a James H. Clark, fundador de *Silicon Graphics*, para crear la *Mosaic Communications Corporation* dedicada al desarrollo comercial del navegador *Mosaic*. Para evitar problemas con el NCSA, en abril de 1994, la compañía cambió su nombre por *Netscape* y su navegador pasó a llamarse *Netscape Navigator*. Por su parte, NCSA siguió desarrollando el *NCSA-Mosaic*, que podía descargarse de forma gratuita, y siguió contribuyendo a la elaboración de estándares abiertos para la web. El NCSA también desarrolló el servidor web *httpd*, base del actual servidor *Apache*.

En sus primeros años, 1992 y 1993, la primera expansión del *World Wide Web* tuvo lugar en el mundo académico (universidades y centros de investigación). En enero de 1993 había unos 50 servidores web en todo el mundo. En octubre de 1993 había más de 500. En los años 1994 y 1995 este crecimiento se extendió a todos los ámbitos. Desde la perspectiva histórica actual, al año 1994 se le conoce como el “el año de la Web”, ya que ese año concentró una serie de acontecimientos que dieron un impulso definitivo a la expansión de la web. Como se ha indicado, en el mes de abril se fundó Netscape; en mayo, se celebró en el CERN la primera Conferencia WWW, que tuvo una segunda edición en octubre de 1994 de ese mismo año en Chicago. También en octubre, Tim Berners-Lee, que se había trasladado del CERN al MIT, fundó el *World-Wide-Web Consortium* (W3C) con la participación de *Advanced Research Projects Agency* (ARPA) y la Comisión Europea. El propósito de este consorcio era el desarrollo de estándares para la web. Muchas grandes compañías (Microsoft, IBM, etc.) se unieron rápidamente al Consorcio. Bajo el impulso de Berners-Lee, primero el CERN y a continuación el W3C decidieron que los estándares de la web estuviesen disponibles libremente, sin patentes ni derechos de autor: esto facilitó la rápida expansión de esta tecnología.

A principios de 1993, un grupo de amigos, formado por Toni Bellver, Carles Bellver, Enrique Navarro, Enrique Silvestre y Jordi Adell, que trabajaban en la Universitat Jaume I de Castellón instalaron un servidor web con el objetivo de experimentar con esta nueva tecnología. Los cuatro primeros componentes del grupo trabajaban en el Servei d'Informàtica de la Universitat. Adell era profesor del Departamento de Educación. En septiembre de 1993, después de varios meses de funcionamiento estable, el grupo informó a Berners-Lee que habían instalado un servidor web experimental en la dirección <http://www.uji.es>. Este servidor fue incluido en la lista de servidores que mantenía el servidor del CERN. Fue el primer servidor web español.

BIBLIOGRAFÍA

ISAACSON, W. (2014), *The innovators*. Simon & Schuster.

LICKLIDER, J. C. R. (1963), *Topics for Discussion at the Forthcoming Meeting, Memorandum For Members and Affiliates of the Intergalactic Computer Network*. Washington, D.C. Advanced Research Projects Agency.

LEINER, B. M.; CERF, V; CLARK, D.; KAHN, R.; KLEINROCK, L.; LYNCH, D.; POSTEL, J.; ROBERTS, L.; WOLFF, S. (2009), “A Brief History of the Internet”, *SIGCOMM Computer Communication*, octubre, rev. 39, 5, pp. 22-31.

TESLA, NIKOLA. (1908) “The Future of the Wireless Art”, *Wireless Telegraphy & Telephony*, editado por Walter W. Massie & Charles R. Underhill, pp. 67-71.

Andrés Marzal Varó¹
María José Castro Bleda²
Pablo Aibar Ausina¹

(1) Universitat Jaume I

(2) Universitat Politècnica de València

Resumen: En este trabajo revisamos el avance de la profesión de informático en España desde sus inicios, en los años 60. Los once ordenadores por millón de habitantes que había en 1964 aumentaron a ochenta y dos en 1970, lo que supuso un primer aviso sobre la necesidad de disponer de profesionales especializados en el sector (hasta entonces, provenientes de campos muy heterogéneos). Habría que esperar hasta la década de los 80 para que salieran al mercado los primeros titulados universitarios en informática. Por otra parte, el modelo de uso y explotación de la informática se transformó radicalmente en esos años debido a la expansión de los ordenadores personales: la informatización dejó de ser un recurso para empresas de gran tamaño y se convirtió en una demanda de dimensiones inconcebibles pocos años antes. Podemos hablar de unos años de esplendor para el sector profesional de la informática hasta que en 2001 estalló la burbuja «punto.com», con efectos devastadores sobre la economía y la profesión informática.

EL INICIO: EL FINAL DE LOS AÑOS 50 DEL SIGLO XX

El primer ordenador comercial fabricado en serie y vendido internacionalmente es el UNIVAC I, de Remington Rand (más tarde parte de Sperry, ahora Unisys), cuya primera venta tuvo lugar en marzo de 1951, en Estados Unidos. Ese mismo año entró en funcionamiento un ordenador con fines comerciales en Gran Bretaña, el LEO I, y la francesa Bull lanzó el Gamma 3.

Si obviamos el uso de máquinas tabuladoras en los años 20 del siglo XX, podemos considerar que la industria española introdujo los sistemas informáticos relativamente tarde. Unas pocas empresas y entidades se anticiparon e introdujeron los primeros sistemas informáticos en los años postreros de la década de los 50 del siglo XX. La tardía introducción de computadores en nuestro país tiene origen en los insalvables problemas para la importación que soportaba una España autárquica y aislada. Fue necesaria la firma del Convenio Defensivo, de Mutua Defensa y Ayuda Económica entre Estados Unidos y España, en septiembre de 1953, para que la adquisición e importación de equipos estadounidenses fuese siquiera posible.

El primer ordenador instalado en España es un IBM 650 adquirido para RENFE en 1957 y puesto en explotación en 1958. La máquina existía desde 1953 y se habían vendido cerca un millar en el mundo. El segundo ordenador fue un UNIVAC para cálculo científico de la Junta de Energía Nuclear y se adquirió en 1960. En 1961 se instaló en Valencia un UNIVAC para Hidroeléctrica Española. Ese mismo año, la Empresa Nacional Siderúrgica Sociedad Anónima (ENSIDESA) adquirió un IBM 1401. Un año más tarde, la misma empresa instaló otro equipo para su sede madrileña. En 1962 se instalaron equipos UNIVAC en el Banco Español de Crédito (Zaragoza y Madrid) y en la Compañía Telefónica Nacional de España. En 1962 Sevillana de Electricidad y Galerías Preciados instalaron un IBM 1401, modelo del que instalaron en España una decena de equipos entre 1962 y 1963. General Eléctrica Española adquirió un IBM 650 en 1963. En los años inmediatamente posteriores se implantó un número creciente de máquinas IBM 360. Podemos citar Pegaso y La Seda entre las empresas que se informatizaron con estos equipos en aquellos años. Bull respondió al modelo 360 de IBM con la serie Gamma, de gran éxito. El primer Gamma instalado en España fue adquirido por FENSA en 1963. Más adelante Bull-General Electric compitió con la serie 360 de IBM con su Serie 400. Algunos de los primeros ordenadores con usos docentes se instalaron en las Escuelas Especiales de Ingeniería en los años 60.



Figura 7.1. Años 50

EL INICIO DE UNA PROFESIÓN: LOS AÑOS 60

La primera Feria de Muestras con presentación de material informático en España se celebró en 1961: el Salón Informativo de Material de Oficina (SIMO). El propio nombre de la feria da una idea cabal de la consideración que por aquel entonces tenía el sector. SIMO mantuvo las siglas largo tiempo, pero en 1969 pasó a anunciarse como Feria de Muestras Monográfica Internacional del Equipo de Oficina y de la Informática. Nuevamente, el nombre revela la visión que entonces se tenía de la informática en el entorno de los negocios en España.

En los 60, algunas multinacionales se instalaron en España para ir dotando a las empresas de los equipamientos informáticos que precisarían en los siguientes años. Podemos citar entre estas a la estadounidense International Business Machines (IBM) ya implantada en España desde el año 1949 —aunque su presencia arranca realmente con la filial Máquinas Comerciales Watson, dedicada a la venta de tabuladoras, en 1941—, la francesa Compagnie de Machines Bull (que actuó en España con los sucesivos nombres de S.A. Bull de España (SABE), Bull-General Electric (Bull-GE), Honeywell Bull, Bull-CII y simplemente Bull) o la alemana Nixdorf Computer AG (hoy absorbida por Siemens). Estas empresas se asentaron principalmente en Madrid y Barcelona, aunque crearon algunos polos de interés en Bilbao (con una sede de Bull) o Valencia (con una planta de fabricación de IBM en la Poble de Vallbona). Madrid albergaba el 50% del parque informático nacional, y Barcelona el 30%. El 50% de los equipos daban servicio a la administración y al sector financiero. El primer sistema basado en tecnología española es Factor-P, una facturadora de la empresa catalana Telesincro diseñada por Joan Majó (fundador de la empresa en 1963) y Jordi Vidal. Era un sistema ligero orientado a la contabilidad que competía en el nicho de dispositivos como los de la serie Mercator, de la casa italiana Olivetti.

Se puede comprobar, pues, que el ritmo de implantación de equipos fue lento en los primeros años sesenta: se pasó de 11 ordenadores por millón de habitantes en 1964 a 37 en 1967, y a 82 en 1970. Pese a este número relativamente bajo de equipos, el crecimiento del parque informático fue poniendo de manifiesto una acusada falta de profesionales en el sector. Esta escasez influyó muy positivamente en el prestigio y nivel salarial de los informáticos. Los profesionales que se encargaron de los sistemas informáticos provenían entonces de muy diferentes campos y su formación era muy heterogénea. En el caso de contar con formación universitaria, esta solía ser en Matemáticas, Física (que contaba con especialidades en electrónica), algunas Ingenierías o incluso Química. Independientemente de la formación original, los profesionales acababan asistiendo a cursos de FORTRAN y COBOL, de manejo de los modelos 1401 o 1620 de IBM, de los modelos GE400 y GE10 de Bull, o del UNIVAC y los equipos de NCR, Burroughs o Nixdorf Computer. La necesidad de una formación superior muy especializada se iba haciendo manifiesta, pero en aquel entonces no había estudios universitarios de informática con los que nutrir de especialistas a la profesión.



Fuente: Carteles del SIMO de los años: 191, 1962, 1963, 1966 y 1969. Archivo de UNIVAC, BULL e IBM
Figura 7.2 Años 60

LA EXPANSIÓN DE LA PROFESIÓN Y LA FORMACIÓN ESPECIALIZADA: LOS AÑOS 70

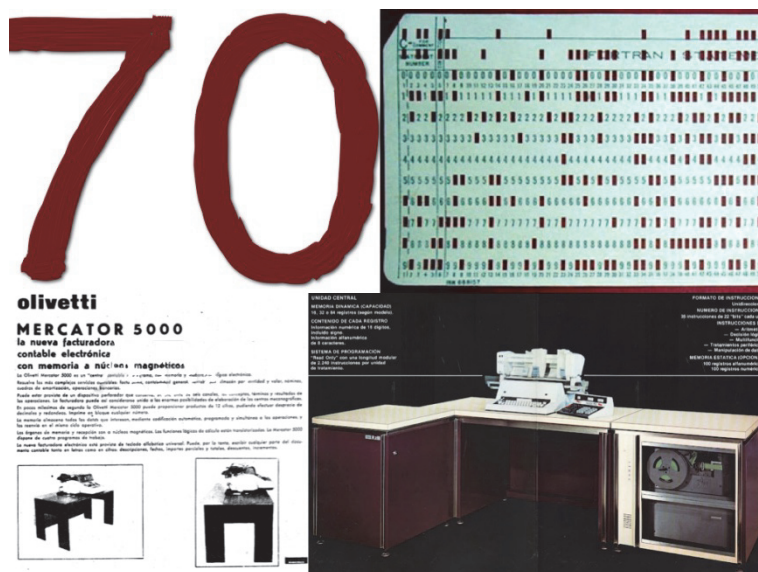
La relativa anarquía en la calificación de puestos de trabajo de entonces encontró cierto orden con la publicación del Convenio Colectivo Sindical para Bull General Eléctrica S.A. en el Boletín Oficial del Estado del 4 de marzo de 1970¹. Se establecía allí una clasificación profesional, con funciones de comercial y de mantenimiento. En la función comercial distinguía entre una escala técnica y otra facultativa, cada una con una serie de puestos:

- en la escala técnica: Operador, Operador principal, Programador, Programador principal, Analista, Analista principal, Técnico de sistemas y Técnico de sistemas principal; y

¹ Resolución de la Dirección General de Trabajo por la que se aprueba el Convenio Colectivo Sindical, de ámbito interprovincial, para la empresa «Bull General Electric S.A.». Boletín Oficial del Estado, <http://www.boe.es/boe/dias/1970/03/04/pdfs/A03518-03526.pdf>, 1970.

- en la escala facultativa: Ingeniero de sistemas, Ingeniero de sistemas principal, Ingeniero comercial (con cinco grados: principiante, 1, 2, 3 y 4), Perforista, Perforista-Verificadora, Monitora y Monitora principal.²

Esta clasificación sirvió de base, más tarde, para determinar perfiles en estudios sistematizados de informática.



Fuente: La Vanguardia (19 abril de 1963)

Figura 7.3. Años 70. Tarjeta perforada en Fortran y Olivetti Mercator 5000

Por no dejar sin conclusión el apunte sobre una industria autóctona capaz de ganar cuota de mercado en la venta de equipos informáticos de diseño propio, comentaremos qué ocurrió finalmente con Telesincro, empresa que llegó a tener el 16% del sector en 1972. Desgraciadamente no resistió la competencia de los equipos de gestión «ligeros», como el Sistema 3 de IBM, y se acabó incorporando a la Sociedad Española de Comunicaciones e Informática S.A. (Secoinsa) en 1976 (Barceló, 2008). Secoinsa, creada con el apoyo del INI y participada por Telefónica, trató de mantener un sector industrial propio en la informática. Esta empresa, llamada a ser una especie de compañía informática «de bandera», acabó siendo adquirida por Fujitsu, que era socio de Secoinsa desde 1975. (Y en 1984 vio la luz el Plan Electrónico e Informático Nacional (PEIN), que modificó la estrategia estatal y retiró al INI de Secoinsa, perdiéndose así toda presencia de capital público o privado nacional en la empresa.)

² Llama la atención el sexismo del convenio, que marca con el sexo femenino los trabajos relacionados con la perforación de fichas.

Los años setenta vieron un notable crecimiento del parque informático para la gestión de empresas. En la Comunidad Valenciana tuvo especial notoriedad el centro de cálculo del edificio Cemeco, inaugurado en 1973 para albergar el equipamiento informático de gestión de la Caja de Ahorros de Valencia. La *Asociación de Técnicos en Informática*, de la que hablaremos más adelante, publicó en 1976 un «Manifiesto Informático» que pretendía dar un toque de atención sobre el carácter estratégico de la informática. Se hablaba en el documento de la existencia de unas 2.000 empresas con equipos informáticos propios y unas 10.000 que hacían uso de la informática para su gestión. El manifiesto ponía énfasis en la importancia estratégica de la informática, con la predicción de un impacto mucho más allá de lo meramente técnico. El documento acababa llamando a la definición de un Plan Informático Nacional. Entre otros asuntos, este Plan debería considerar los aspectos formativos y el asociacionismo profesional en el sector.



Fuente: Archivo de IBM y archivo de Apple

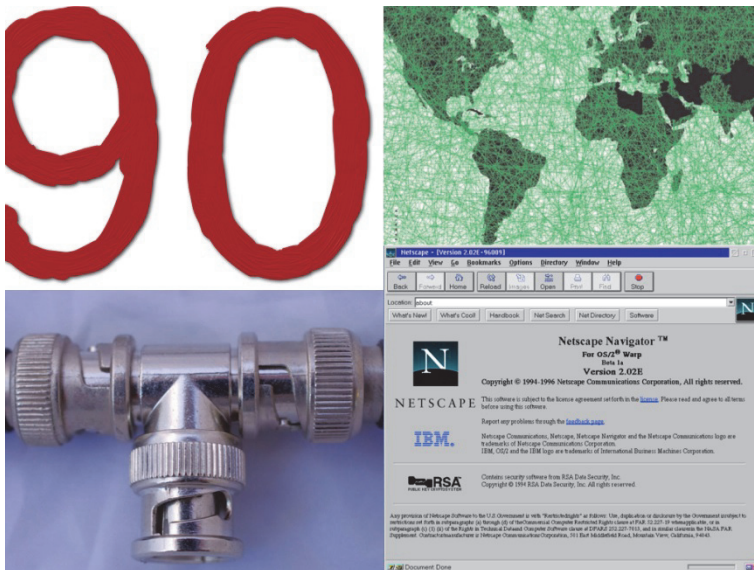
Figura 7.4. Años 80

EL CAMBIO DE PARADIGMA DE LOS ORDENADORES PERSONALES Y LA FORMACIÓN UNIVERSITARIA: LOS AÑOS 80

La situación de los profesionales de la informática cambió en la década de los 80 con la irrupción de titulados universitarios en informática. Se creó entonces una seria fractura en el sector por no haber una propuesta de homologación para los profesionales existentes, que se sentían amenazados por la aparición de unos estudios que podrían llegar a ser exigibles para el ejercicio profesional. Por otra parte, en esos años se dio un fenómeno que transformó radicalmente el modelo de uso y explotación de la informática: la apari-

ción y brutal expansión de los ordenadores personales, capaces de soportar la gestión de pequeñas y medianas empresas con costes asumibles por estas. La informatización dejó de ser un recurso para empresas de gran tamaño y apareció en poco tiempo una demanda de dimensiones inconcebibles pocos años antes. Al mismo tiempo, la administración española iniciaba una profunda transformación modernizadora. Se multiplicaron rapidísimamente las administraciones públicas que requerían de procesos automáticos, dispositivos y profesionales para tratar volúmenes de información crecientes. Podemos hablar de unos años de esplendor para el sector profesional de la informática. Los autores nos licenciábamos en 1990 y era habitual que muchos compañeros trabajasen en el campo de la informática a la vez que terminaban su carrera, hasta el punto de que muchos estudios quedaron en un semiabandono por la dificultad que suponía conjugar la actividad laboral con la superación de ciertas asignaturas o la realización del proyecto final de carrera. Un número muy importante de los titulados de aquel entonces encontraron trabajo en la administración pública, en clara expansión y necesitada de informáticos.

Hay que señalar, en cualquier caso, que el título universitario no se percibía socialmente como una necesidad para el ejercicio de la profesión. El nivel de intrusismo era grande y faltaba una visión clara de qué papel debían jugar los diplomados y licenciados en informática en las organizaciones empresariales. Es más, ni siquiera se percibía una diferencia clara entre el Diplomado y el Licenciado. El número de universidades en España se disparó y el de facultades o escuelas que ofrecieron estos títulos creció sin planificación alguna.



Fuente: Navegador web Netscape Navigator

Figura 7.5. Años 90

LA PROGRESIVA SATURACIÓN DEL MERCADO Y LA IRRUPCIÓN DE INTERNET: LOS AÑOS 90

En 1980 solo había facultades de informática en Madrid, Barcelona, Bellaterra y San Sebastián. En los siguientes veinte años la expansión fue enorme. En 2003, 47 de las 65 universidades españolas ofertaban estudios de Ingeniería Informática. El número acumulado de titulados universitarios en informática hasta ese año se estimaba en unos 70.000.

A finales de los 80 y principios de los 90 se implantaron en España consultoras que contrataron un enorme número de titulados superiores a muy bajo coste por su creciente abundancia en el mercado. Muchas de estas consultoras, a la postre, han acabado jugando un papel similar al de las empresas de trabajo temporal: contratan mano de obra a bajísimo coste y venden su tiempo de trabajo a precios muy elevados, bajo la no siempre observada premisa de que esos trabajadores cuentan con la debida experiencia. Ha cuajado el término «cárnica» para referirse a estas empresas, que se perciben como meras vendedoras de «carne» (recursos humanos). Los bajos sueldos que ofrecen producen una constante rotación de personal y acaban configurando equipos en los que casi todos los integrantes son programadores junior. Naturalmente, así no se puede esperar una calidad elevada en el resultado final y se realimenta un círculo vicioso que genera una pésima imagen de la profesión.

En la segunda mitad de los años 90 asistimos a otro cambio radical con la apertura de Internet al gran público y al comercio, que se hizo accesible a todos gracias, especialmente, a la *World Wide Web*. Lo que hasta entonces era una red que unía a universidades y algunas administraciones, se abrió a la sociedad y la transformó hasta el punto de que hoy cuesta recordar que hubo un tiempo sin Internet. En la última mitad de los años 90 surgieron empresas que pretendían cubrir nichos de intermediación en Internet, o portales de acceso a Internet, o buscadores,... Solían proponerse modelos de negocio que pasaban por acumular capital inversor para despegar, acaparar clientes que generalmente no compraban nada ni pagaban por nada (a lo sumo eran un público objetivo para agencias de publicidad) y vender la empresa por cifras astronómicas cuando su precio en bolsa se disparara sin razón alguna. Internet se percibía como el soporte de una «Nueva Economía» que nadie entendía bien, pero en la que había que posicionarse a cualquier coste. El éxito de algunas operaciones de compraventa o continuas salidas a bolsa de empresas iba alimentado el mito de esa economía de futuro de la que no se podía quedar excluido. Las novísimas empresas acumulaban más y más capital que gastaban en su expansión y en la propia operación del supuesto negocio, sin repartir dividendo alguno. El resultado indefectible era un aumento casi geométrico de la cotización en bolsa. Los parecidos con un esquema de Ponzi o la burbuja inmobiliaria son evidentes.



Figura 7.6. Años 10 del siglo XXI. Logos de empresas tecnológicas, evolución del índice NASDAQ 100 (1986-2012) y aparición de nuevas asignaturas de informática en los colegios de Estados Unidos

LA CRISIS «PUNTOCOM» Y LA PRECARIZACIÓN: LA PRIMERA DÉCADA DEL SIGLO XXI

La burbuja «puntocom» estalló en 2001, tras alcanzarse el pico de valorización bursátil en marzo de 2000, con efectos devastadores sobre la economía y la profesión informática. La cantidad de dinero que se esfumó en menos de un año solo ha sido superada por la que se volatilizó tras el estallido de la siguiente burbuja económica: la financiera de 2008. De paso, la profesión informática perdió algo de su atractivo, en parte construido sobre una fantásica expectativa de enriquecimiento rápido. Una consecuencia fue la drástica contracción de la demanda de estudios universitarios en informática en todo el planeta. España no fue ajena a esta debacle de la demanda y vio como una red de facultades y escuelas universitarias súbitamente sobredimensionada tenía problemas para cubrir las plazas ofertadas, incluso después de recortar seriamente sus *numerus clausus*. Resulta sarcástico que las vocaciones estuviesen vinculadas a un modelo, el «puntocom», que se apoya en la ocupación de un número relativamente bajo de profesionales: la plantilla necesaria para iniciar una empresa «puntocom» (como Twitter, por poner un exitoso y reciente ejemplo) ronda la decena de personas (exactamente doce en el caso de Twitter) y, caso de alcanzar el tan improbable como deseado crecimiento exponencial de usuarios, puede operar con una plantilla de unos pocos centenares de empleados (hoy hay unos seiscientos trabajadores en Twitter, con los que dan servicio a los cinco continentes). Así pues, el atractivo de la profesión parecía basarse en la

oportunidad de formar parte de un grupo de trabajadores o empresarios que es muy reducido. Cuando disminuyó la perspectiva de poder acceder a ese minoritario mercado laboral, se redujo asimismo el número de profesionales con formación superior que producía el sistema educativo.

Asistimos desde hace un lustro a otro proceso de fuerte transformación en el campo: el que propicia la informática móvil. En los últimos años han empezado a aparecer dispositivos móviles, con conexión de red y considerable potencia de cálculo que redefinen el modo en que nos relacionamos con los ordenadores. Está por ver cómo afectará a la profesión esta nueva revolución.

UNA BREVE DIGRESIÓN: EL SECTOR DE LOS VIDEOJUEGOS

No hemos mencionado aún un sector específico que también ha vivido sus altibajos: el de los videojuegos. En los años 80 hubo un sector industrial en España de gran potencia que cubría todas las fases del proceso: diseño, producción, realización y comercialización de software. Empresas como *Dinamic Software*, *Erbe Software* u *Opera Software* eran tremendamente innovadoras en aquel entonces, y capaces de exportar sus productos. Los juegos estaban orientados casi siempre a ordenadores personales de gama baja y gran penetración en el mercado doméstico: Sinclair ZX Spectrum, Commodore 64, Amstrad CPC, etc. El esplendor desapareció en los años 90, y en esa década apenas se cuenta algún caso de éxito aislado (como *Commandos*, de Pyro Studios). En los últimos años hay un renacimiento de la industria en España orientada a la producción de juegos para consolas y aplicaciones para dispositivos móviles, así como la aparición de titulaciones universitarias orientadas a la formación de perfiles profesionales específicos. El sector de los videojuegos y, en general, el del entretenimiento digital, es de los que mayor auge experimentará en los próximos años y convendría que hubiese una oferta de profesionales suficiente para posicionar bien a España en un mercado tan prometedor como competitivo.

UNA REFLEXIÓN FINAL

Acabamos señalando que el sector de la informática, en general, sigue nutriéndose de personas sin la formación adecuada y que los perfiles profesionales no están claramente delineados ni correctamente percibidos por muchos de los demandantes de servicios profesionales. Según el informe sobre el mercado laboral de la informática realizado en 2009 por la Asociación de Ingenieros Técnicos e Ingenieros en Informática³, en el 51% de las ofertas de empleo para dirección de proyectos no se solicita titulación informática. Incluso en la oferta de trabajos para analista de aplicaciones, analista-programador

³ “El mercado laboral en informática 2009”. http://www.ali.es/uploads/miprofesion/75_MERCADO_LABORAL_EN_INFORMATICA_2009.pdf

o programador se exige titulación informática en menos de la mitad de los casos. Hay un rayo de esperanza en el hecho de que la tendencia de los últimos años vaya en la línea de aumentar la exigencia de la cualificación apropiada para cada puesto, pero hay mucho camino por recorrer hasta llegar a una verdadera profesionalización del sector.

BIBLIOGRAFÍA

ASOCIACIÓN DE TÉCNICOS DE INFORMÁTICA (ATI) (1976), “Manifiesto Informático”, *Novática*, nº 7, pp. 2–4.

BARCELÓ, M. (2008), *Una historia de la informática*, Barcelona: Editorial UOC (Universitat Oberta de Catalunya).

Juan V. Oltra Gutiérrez
Universitat Politècnica de València

Resumen: La informática, los ordenadores, han generado muchos sueños y pesadillas no solo a los profesionales que viven su día a día, sino a la sociedad en general. Las relaciones entre la vieja sociedad y la sociedad de la información, en esa aparente lucha entre átomos y bits, necesita de un mediador, y éste es la autorregulación, la ética profesional, que no es más que, de acuerdo con Bowyer, la “relación profesional con clientes, la profesión con la sociedad, relaciones entre profesionales, entre empleador y empleados y detalles técnicos especiales de la profesión”. Pero el tiempo no solo cambia los detalles técnicos de la profesión, sino a la profesión misma e incluso a la sociedad, por lo que cabe preguntarse si ese transcurso del tiempo no ha afectado a la ética profesional en sí misma. Con la vista puesta en el esquema que clasifica las dimensiones morales más comúnmente usado hogaño, el de los profesores Laudon, buscamos el contraste con las visiones que eran comunes antaño, revisando prensa convencional y especializada, y literatura científica de unos años en que la informática no era algo tan asentado en la sociedad como hoy.

INTRODUCCIÓN

A pesar de que la informática, considerada bien como realidad, profesión o ciencia, sea algo que ya no podemos calificar como nuevo –la prueba evidente es la existencia misma del patrocinador de la presente compilación, el Museo de Informática– pues contamos con un gran número de profesionales ya jubilados que atesoran entre sus recuerdos la memoria de nuestro pasado, por no apuntar que podríamos erigir nuestro propio y particular panteón de ilustres, lo cierto es que en comparación con otras facetas del saber humano, como la medicina o el derecho, los informáticos somos unos auténticos recién nacidos. De hecho, hace poco no éramos más que una anécdota en la vida española. En 1970, el parque de ordenadores de Cataluña estaba compuesto por 125 ordenadores de distintas épocas que atendían 3.000 personas; el año siguiente aparecía el procesador Intel 4004, considerado como un artículo de lujo.

Poco a poco, la profesión informática empieza a tomar posiciones. Quizá demasiado poco a poco: leemos de la mano de J. A. Pascual en el no tan lejano 2002 que la profesión en España aún está compuesta una gran parte por “grupos amateurs (...), aunque por lo general aparecen y desaparecen en cortos espacios de tiempo, o se convierten en profesionales”, a pesar de tratarse de “una profesión con vocación de futuro (que) se

remonta a los primeros años de la década de los ochenta, cuando los primeros ordenadores profesionales comenzaban a llegar a las empresas y las universidades”.

Bajo este escenario cambiante a nadie debería llamar la atención que los ejes principales que delimitan el comportamiento de todo profesional que vive su día a día laboral entre ceros y unos, hayan ido transformándose con el tiempo, mientras se conforma la profesión. Hay muchos cambios desde el, permítaseme la expresión, origen de nuestros tiempos, hasta hoy. Pensemos sin ir más lejos en lo que era la informática antes de la popularización de Internet, no solo para el profesional, desde dentro, sino para la sociedad en general. Si consideramos que en disciplinas más clásicas, como las antecitadas derecho o medicina, de alguna manera, con distinta velocidad y frecuencia también han evolucionado su marco ético—la Organización Médica Colegial de España promulgó su primer Código de Ética y Deontología Médica en 1978—, tenemos servida la imagen que nos da paso al presente trabajo.

ÉTICA INFORMÁTICA: ORIGEN Y PRECEDENTES

Aunque se ha definido a la deontología como una ciencia normativa, hay que remarcar, según Vázquez, que el derecho es el fracaso de la ética, algo que se incrementa al considerar que en nuestro caso el conjunto de leyes que nos atañen avanzan mucho más despacio que la realidad tecnológica y social, y en ocasiones con escasa sintonía con las mismas. Fracaso en tanto la autorregulación queda arrumbada en determinadas personas, pero sin perder de vista que el derecho mismo nace de la necesidad de marcar una serie de líneas rojas infranqueables. Es por eso al menos llamativo que nos encontremos que casi simultáneamente a que en la Conferencia de Juristas Nórdicos se aconseja la protección de la vida privada mediante instrumentos específicos y más adecuados a las nuevas formas de injerencia, dando paso en Europa al espíritu de la protección de datos, Miller estudie el mismo tema de forma puramente ética, aunque desde un prisma asentado en la religiosidad de la comunidad.

Pero no nos adelantemos. El padre de la cibernética, Norbert Wiener, abrió la caja de los truenos al decir que existe una voluntad de transferir funciones intelectuales del hombre a la máquina (Figura 8.1). Eso provocó un profundo debate que discurría por el cauce de la ética, o de la moral si se quiere, con todos los matices que distinguen a la una de la otra. Wiener diagnosticó que la sociedad se reorganizaría en torno a ese ser inmaterial que es la información, por lo que los instrumentos de comunicación están llamados a tener cada vez un papel más importante en las sociedades modernas. Ese incremento, pronto comprobado en la evolución de los aparatos estadísticos (el censo), de los flujos de información creciente sobre la circulación monetaria, los intercambios económicos en general unido al origen militar de ese incremento al convertirse la guerra moderna en una guerra de cálculo, dio alas a una idea que aún hoy sobrevuela a todo aquel que dedica su tiempo a esta disciplina: si al estado le resulta imposible regular la red, por su extrema dificultad al cubrir todas las posibilidades, y ser algunas de

éstas capaces de provocar importantes problemas en el seno de la sociedad, entonces ¿qué se puede hacer? La respuesta pasa porque, ya que no se puede regular la conducta en la red y la autorregulación a veces falla, lo que debe regularse es el código, su arquitectura. Esa arquitectura debe así velar por aquellos principios y valores públicos distintos de los que únicamente velan por el interés del comercio u otros distintos al interés general.

seremos más fuertes y mejores. Pero también —¡qué duda cabe!— son grandes y terribles los peligros. Los ha advertido el propio Wiener, fundador de la Cibernética: «El peligro de la máquina para la sociedad no procede de la misma máquina, sino de aquello que el hombre haga con ella... El hombre moderno aceptará las decisiones tomadas por la máquina sin pensar demasiado en los principios y leyes en que se basa... Cualquier aparato construido con la idea de que pueda tomar decisiones, pero sin poseer la capacidad de juicio necesaria, respetará la letra pero no el espíritu. ¡Desgraciados de nosotros si dejamos que guíe nuestra conducta sin haber examinado previamente las leyes de su acción y sin saber si aceptamos plenamente las bases de su comportamiento...!»

Fuente: La Vanguardia, martes 17 febrero de 1970, página 48

Figura 8.1. Los terribles peligros de la cibernética o el miedo en 1970, según J. Tomás Cabot en La Vanguardia

VISIONES Y REVISIONES

Cuando la informática empezaba a dejar de ser sueño de autores de ciencia ficción para transformarse en realidad no dejamos de encontrarnos una reedición de las luchas entre luditas y maquinistas cuando la revolución industrial: una y otra vez nos encontramos el enfrentamiento, real o imaginario, de la máquina contra la persona. Y es que cuando la máquina energética se vio por fin más como progreso que como esclavitud, la máquina informacional planeaba otros problemas. Así, vemos noticias sobre la inquietud obrera frente al robot (ABC, 9 de abril de 1955) e incluso en un año tan avanzado como 1966, el subrayado sobre el peligro de que los robots dominen a los seres humanos (ABC, 21 de noviembre de 1966). Hay un párrafo del artículo "Hacia un mundo automatizado" (ABC, 22 de octubre de 1966) que no me resisto a transcribir:

En el umbral de una nueva era, (...) el único espectro que amenaza a una humanidad automatizada es el del "tiempo libre". ¿En qué utilizará su mente el obrero de una industria provista de servomecanismos, mientras la "retroalimentación" y los "convertidores de optimización" cumplen su obra silenciosa, descargando de esos deberes rutinarios y enojosos al cerebro? Tendremos que ir pensando en rezar a Dios para que esa humanidad ociosa quede libre de malos pensamientos. Porque, desgraciadamente

para ella, los físicos del Instituto Español de Automación y de los organismos similares avanzan más de prisa que los pedagogos.

Pero caeríamos en un error si solo subrayáramos las ideas negativas como las dominantes. Efectivamente, ya en una fecha tan temprana como el 14 de octubre de 1944, con el Harvard Mark 1 como noticia de impacto, ya se puede leer en *The American Weekley*: “El robot de Harvard es un supercerebro capaz de resolver problemas que el hombre no tenía ninguna esperanza en resolver”. Con prácticamente las mismas palabras se saluda a las experiencias pioneras (de las universidades de Manchester, Cambridge, Estocolmo...) desde las páginas de periódicos españoles o franceses: el padre dominico francés Dubarle, en *Le Monde* de 20 de diciembre de 1948, a la sombra de la cibernética, hablaba de las nuevas máquinas de calcular y su potencialidad, subrayando la ineficiencia de la mente para hacer política y especulaba con construir un aparato mundial de toma de decisiones.

Esa idea, tan querida para la ciencia ficción de principios del siglo pasado, de suplir al hombre en toma de decisiones y producción de conocimiento, parecía cada vez más posible. Y es que como indica Breton se percibía la incapacidad flagrante del hombre frente a la imposibilidad de equivocarse de las máquinas. Algo que pronto sería universal. De hecho, este autor llega a indicar que los grandes regímenes comunistas, tras criticar severamente la cibernética y la informática como falsas respuestas técnicas a problemas políticos, se pusieron manos a la obra para alcanzar a sus competidores occidentales.

Y es que conforme los ordenadores dejan de ser una realidad despreciable estadísticamente para quedarse en realidad, más voces se alzan no solo en su defensa, sino apuntando la idea de que la informática pronto será tomada como indicador necesario para medir el desarrollo de los pueblos. Así, leemos en el Boletín de ALI nº 12, de noviembre de 1989, en su editorial “Deontología y honestidad en informática”:

Tanto alrededor de la informática como en su regazo, han ido surgiendo a lo largo de estos años múltiples oportunidades para muchos (...) siendo cada vez más la informática uno de los indiscutibles motores del desarrollo de los pueblos.

Una idea que se repite cada vez más en la prensa especializada; en palabras de J. A. Pascual, en 2002: “De hecho, en un futuro cercano, es probable que el nivel de desarrollo de un país se mida por la diversidad y fortaleza de su sector informático”.

Y LOS ORDENADORES SE CONECTARON UNOS A OTROS

Esos miedos y admiraciones, amor y odio, parecen permanecer dormidos hasta que la red llega a ser algo real. En ese momento aparecen nuevos demonios. Uno de los más persistentes, vivo desde los primeros momentos, es si Internet y su tan manida aldea global nos convertirán en esclavos o, al contrario, nos liberará. Así, podemos leer a algunos autores como E. Dyson que lo importante de la red repercutirá en las personas,

dado que sirve para desarrollarnos intelectual y emocionalmente, pero no altera la esencia de nuestras personalidades, frente a las dudas de Lessig que alertan sobre su incidencia en una obsesión patológica por el trabajo, descuido de la familia y valores humanos, impaciencia e interés desmedido por el dinero y los precios de las acciones.

Y no solo eso. Ya en el año 2000 alertaba sobre un dilema: Internet tiene un sabor genuinamente americano. ¿Hasta qué punto transformará la red a las personas que se unan a ella y hasta qué punto cambiará a la gente en la red? ¿Es el espíritu libre que caracteriza a la red americano o es algo inherente a la red? Puede parecer algo paranoico, pero si leemos la realidad que describe C. Salmon sobre los *call centers* indios, y como sus trabajadores viven hoy en una vida que no es la suya, sumergidos en el sueño americano pero a kilómetros de Calcuta, vemos que no era una duda gratuita, en un planeta sumergido en una globalización creciente, donde la red no pertenece a ningún país o grupo en particular, sino que es un entorno donde muchos pueden encontrar de forma diversa tanto el éxito como el fracaso. Quizá el peor camino sea, como dice Lessig, la imitación: muchos extranjeros visitan Silicon Valley con la esperanza de contagiarse. Se fijan en las características erróneas, como la tecnología, pero no en la cultura de las empresas y la gente. O, por decirlo de otra manera, pretenden crear Apple, Facebook, Google o Intel, sin antes crear la universidad de Stanford.

En realidad todos estos dilemas, y muchos más, hace tiempo que vienen siendo estudiados en un marco que ya es clásico: el de las dimensiones morales de los sistemas de información que clasifica en cinco los principios a tener en cuenta. Vamos a dedicar a éstos y a su visión en la prensa especializada el siguiente punto.

LAS CINCO DIMENSIONES MORALES DE LOS SISTEMAS DE INFORMACIÓN

Si bien la ética profesional cuenta con numerosas definiciones, fuera y dentro de la informática, el problema ha venido dado por la forma de llevar algo, en apariencia etéreo o cuando menos cualitativo, al campo de lo cuantitativo. Es por eso que fueron apareciendo diversos modelos, siendo uno de los más empleados en la actualidad el de los profesores Laudon, que divide en cinco las dimensiones morales a considerar. Marco de uso actual pero desde el que, subidos a su atalaya, si miráramos hacia atrás podríamos caer en el vicio nefando del cronocentrismo. Para no quedarnos con una foto fija y tener una perspectiva del pasado más reciente, analizamos cuatro colecciones completas de publicaciones de prensa especializada, de consumo generalista. Revistas de informática que, a la venta en los quioscos y siendo de gran tirada, nos permiten presumir que la diferencia entre opinión publicada y opinión pública no irá muy desencaminada.

Las publicaciones consultadas son PC-Manía, Computer Hoy, PC-Actual y Netmanía, desde el origen de su aparición en quioscos hasta el cierre de las mismas o, caso de

seguir activas, hasta diciembre de 2014. Ello nos permite tener una visión continua desde 1993, cuando Internet no era más que un bello sueño en España, cuando no algo desconocido absolutamente para el español medio, hasta la actualidad. En puridad, debería haberse introducido un factor corrector que nivelara los distintos meses de los años pues, en alguna ocasión tan solo una cabecera permanecía en los quioscos, mientras se dan otros periodos con tres simultáneas. Pero comoquiera que el propósito es simplemente dar una visión aproximada de una evolución histórica y no la precisión estadística, hemos prescindido de él.

De ellas se han consultado todos aquellos artículos que, de una u otra manera, pudieran considerarse dentro del prisma ético o deontológico. Y a continuación, se han clasificado en esas cinco dimensiones, para poder tener una idea somera de qué preocupaba o dejaba de preocupar en cada momento al segmento de la sociedad mínimamente interesada en la informática, siquiera fuera como pasatiempo.

Quedaría ver cómo ha cambiado, si es que lo ha hecho, el prisma para los profesionales. Para ello recurriremos a una foto fija: sobre el estudio que Vázquez y Barroso hacen sobre los códigos éticos de las empresas e instituciones más relevantes de la época (de 1996), enunciando sus elementos coincidentes, efectuamos atendiendo al esquema de los profesores Laudon una reclasificación, para ver en qué porcentaje era significativa cada dimensión, en ese momento, con respecto a la totalidad. Pero vamos con las dimensiones.

1. Derechos y obligaciones hacia la información. Aquí se plantean cuestiones del orden de ¿qué derechos de información poseen los individuos y las organizaciones con respecto a sí mismos? ¿Qué pueden proteger? Encajan aquí todos esos flecos que la ley de protección de datos pueda dejar pendientes (recordemos que en su artículo 32, la ley prevé la posibilidad de acuerdos sectoriales o decisiones de empresa) o no resueltos en firme, como el control del correo electrónico en las empresas o el uso de la informática para el control de ciudadanos, sean éstos pacíficos o no (prevención de atentados).

2. Derechos y obligaciones de propiedad. En este caso apuntamos a los flecos de la muy recientemente modificada Ley de Propiedad Intelectual y su aplicación, desde su marco tradicional, a una sociedad cada vez más digital que está en una frecuencia distinta que sus legisladores. Se trata no solo de música, cine literatura o software, sino de ingeniería inversa, vacunas elaboradas sin pagar licencias, etc.

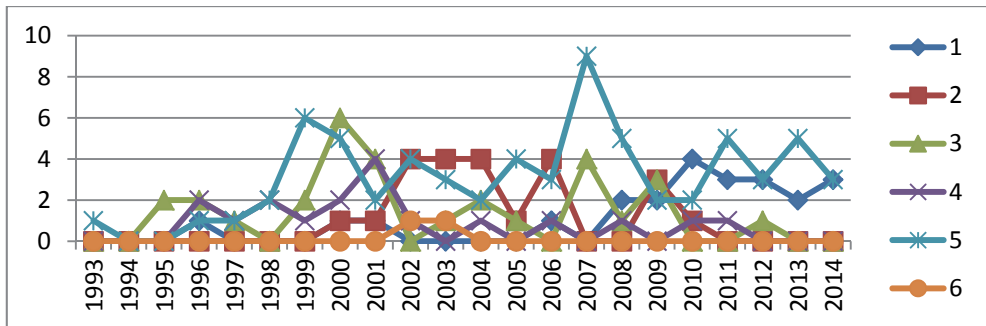
3. Rendición de cuentas y control. ¿Quién es el responsable y debe rendir cuentas si es preciso por el daño hecho a la información individual y colectiva? Su objeto son hechos como la suplantación de la personalidad, falsas identidades o problemas como el *cybersquatting*, *phising*, *pharming* o *web spoofing*. Hay autores que incluyen aquí la libertad de expresión y otros elementos como el *whistle blowing*, algo visto de forma muy negativa en muchos países (una traducción muy generalizada en Hispanoamérica es “soplón”) mientras en otros es protegido por la ley. Más allá del control del correo electrónico, por algunas empresas se establece un control riguroso del comportamiento

de sus empleados, aun fuera de su horario laboral. Es algo sobre lo que reflexiona Lessig cuando dice con acierto que los empresarios que tratan a sus empleados como niños tienen todos los números para acabar con empleados que merecen ese trato. Bowyer habla sobre la política corporativa incluyendo descripción del ambiente de trabajo, como por ejemplo un lugar libre de drogas y alcohol. Vemos en esta larga relación como hay aspectos pre-Internet, que siguen siendo de interés y otros sin razón alguna para la consideración en una sociedad con la red ausente.

4. Calidad del sistema. Se habla de estándares de calidad de los datos y del sistema precisos para proteger los derechos individuales y la seguridad de la sociedad. El estudio de las consecuencias si un sistema falla, la resiliencia.

5. Calidad de vida. Sobre los valores a preservar en una sociedad basada en la información, cómo evitar que se violen los derechos de los ciudadanos y en particular qué valores y prácticas culturales se debe apoyar. Esto último lo ha convertido en un gran cajón de sastre, permitiendo la inclusión de elementos tan diversos como la accesibilidad y usabilidad, las reivindicaciones de género, el cuidado del medioambiente y las ayudas a los más débiles de la sociedad. Don Seller ya hablaba en 1994 de niños y discapacitados. Pero aún se va más allá, estudiando los cambios en las relaciones humanas debidos a los juegos, en concreto de los juegos compartidos, y de la realidad virtual y la simulación.

Del más de medio millar de revistas revisadas (655 ejemplares) en 164 de ellas encontramos referencias a alguna de las dimensiones o, en algún caso, de forma genérica, abarcando a más de una de ellas. Eso nos conforma seis categorías, una por dimensión y otra adicional para éste último caso, repartidas de forma irregular a lo largo del periodo 1993-2014 (Figura 8.2).



Fuente: elaboración propia

Figura 8.2. Número de artículos que tratan sobre alguna dimensión ética en la prensa especializada

Vemos que aunque la tercera y la quinta dimensión son las más reseñadas, la primera, la dedicada a la privacidad, va tomando un incremento que, no puede interpretarse

como casualidad, ya que coincide en su inicio con la implantación de las redes sociales en España.

Revisitando el estudio que Vázquez hizo en 1996, en un momento que parece quedar muy atrás, con 342.000 usuarios de Internet frente a los más de 28 millones de hoy, donde comparaba una completa colección de códigos deontológicos, en el que se enumeran los 40 elementos más repetidos en ellos, podemos tener una visión general que nos indique *grosso modo* cómo han sido los cambios, simplemente reclasificando sus ítems de acuerdo a las dimensiones de los profesores Laudon (Figura 8.3).



Fuente: elaboración propia

Figura 8.3. Análisis retrospectivo de los elementos más repetidos en los códigos deontológicos, clasificándolos según la metodología de Laudon. Fuente: elaboración propia, basada en (Vázquez, 1996)

Vemos que la dimensión 1, relativa a la privacidad, parece muy reducida con respecto a la importancia mediática que hoy se le suele dar (8 impactos de 40). Aunque ya existía normativa (la LORTAD, de 1992) su grado de implantación no era todo lo importante que hubiese sido deseable. En un momento pre-Internet, con evidente ausencia de fenómenos como los foros y no digamos las redes sociales, la privacidad parecía protegerse por sí sola. Vázquez apuntaba, no obstante, que el gran caballo de batalla es la invasión de la intimidad y la vida privada.

La segunda dimensión, relativa a la propiedad intelectual, sí que es a todas luces la que menos atención atraía (un único impacto). A pesar de eso, en el estudio se aludía con preocupación a una noticia de prensa que indicaba que el 90% de los programas de ordenador (1992, noticia de El País) que se emplean en España son copias ilegales. Algo que habría que reenfoque en un mundo con gran uso del protocolo P2P y su facilidad para compartir elementos digitales.

En cuanto a las dimensiones 3, 4 y 5 (con 5, 19 y 7 ocurrencias respectivamente) parecen dominar en conjunto. Tengamos en cuenta que son dimensiones con un apoyo

normativo inferior a las anteriores, por lo que la obligatoriedad debe dar paso a la discrecionalidad en las empresas y organizaciones, por lo que se puede entender su relación abultada. Máxime si consideramos que su sustrato no se ve tan afectado por el uso de la red de redes como las anteriores.

CONCLUSIONES

Wiener, el padre de la computación moderna, dijo que vivir efectivamente es vivir con la información adecuada, pero se oponía a la idea de que la información pudiera considerarse como una mercancía, sometida a las leyes de mercado. Para él, todo almacenamiento de información frenaría la difusión y la libre circulación, y en un mundo cambiante, la expondría a una enorme depreciación. Breton apuntaba que el quid está en la diferencia de criterios. Así, dando un salto espaciotemporal para visualizar el problema desde un mundo ajeno a la informática, hacía un paralelismo: mientras los indios cedían territorios de caza, los sajones compraban propiedades terrenas. La consecuencia fue la masacre de la nación india. No hay que tener grandes luces para colegir quiénes en este caso serían los indios.

Distintas interpretaciones dan paso a una imagen proyectada que se mueve entre el miedo y la admiración. Entre la duda sobre si la informática en realidad ayuda o perjudica al ser humano. Vázquez y Barroso indican que “existe el riesgo que los nuevos tecnólogos se conviertan en instrumentos de despersonalización de la humanidad” y añaden que “es innegable la influencia que conllevan las nuevas tecnologías en niveles de empleo y calidad de vida en la sociedad”. El camino para romper esa dicotomía entre miedo y esperanzas, muchas veces sobrevaloradas, pasa por el tamiz de la ética. Solo con la autorregulación, que siempre va más allá de la normativa, se puede separar lo dañino de lo beneficioso... y hacerlo ver a la sociedad.

BIBLIOGRAFÍA

- BOWYER, K.W. (1996), *Ethics and computing*. Los Alamitos CA, EE.UU.: IEEE Computer Society Press.
- BRETON, P. (1989), *Historia y crítica de la informática*. Madrid: Cátedra.
- LAUDON, K.; LAUDON, J. (2014), *Management Information Systems 13 Ed.* England: Pearson.
- LESSIG, L. (2001), *El código y otras leyes del ciberespacio*. Madrid: Taurus.
- MILLER, S.H. (1968), *Religion in a Technical Age*, Cambridge, Massachusetts: Harvard University Press.

PASCUAL, J. A. (2002), *La programación en España* Madrid: Hobby Press, Pc Mania nº 32-33

SELLER, D. (1994) *ZAP! How computer can hurt you and what you can do about it.* Berkeley, CA. EE.UU.: Peachpit Press.

VÁZQUEZ, J. M.; BARROSO, P. (1996), *Deontología de la informática (esquemas)*. Madrid: Instituto de sociología aplicada.

Pedro López Rodríguez
Universitat Politècnica de València

Resumen: Este trabajo describe someramente las diferentes alternativas que se han utilizado a lo largo del tiempo para implementar los sistemas informáticos. Se comienza repasando las diferentes técnicas que han permitido mejorar las prestaciones de los procesadores. También se presentan los supercomputadores vectoriales, como una alternativa interesante para ejecutar aplicaciones que trabajan con vectores y matrices. En lo relativo a los sistemas multiprocesadores, se describen las configuraciones típicas y sus implicaciones. Finalmente, se presentan los clústeres como la alternativa utilizada actualmente para los grandes sistemas informáticos y supercomputadores.

INTRODUCCIÓN

El mercado de los computadores evoluciona según la tecnología disponible y las necesidades de la sociedad. En cada momento, la industria informática ofrece diferentes dispositivos. En la década de los 70 había dos categorías de computadores: grandes sistemas o *mainframes*, muy caros, instalados en las grandes corporaciones, y miniordenadores, computadores de tamaño medio, instalados principalmente en universidades.

La tecnología de los computadores ha mejorado de una forma notable. Esta mejora procede de la combinación de los avances tecnológicos con las innovaciones en el diseño de los computadores. Al final de la década de los 70 aparece el microprocesador, capaz de integrar un procesador completo en una pastilla de circuito integrado. Con la aparición del microprocesador se introdujeron nuevas categorías de computadores: los computadores personales y las estaciones de trabajo. Inicialmente de escasa potencia, conforme el número de transistores que componen un circuito integrado fue aumentando, la potencia de cálculo de los microprocesadores también lo hizo. Esto, junto con la reducción de costes obtenida por las grandes tiradas, ha motivado que, cada vez más, los sistemas informáticos estén basados en microprocesador. Los minicomputadores fueron sustituidos por servidores basados en microprocesador. Incluso los grandes sistemas y los supercomputadores están basados en la utilización de microprocesadores.

Actualmente podemos clasificar los sistemas informáticos en: dispositivos personales, computadores personales, servidores, grandes servidores y supercomputadores. El ejemplo más representativo de los primeros lo constituyen los teléfonos inteligentes. En estos dispositivos, si bien las prestaciones son importantes, el coste aquilatado y,

sobre todo, un bajo consumo de energía, son igualmente importantes. En lo que respecta a los computadores personales, el abanico de configuraciones es amplísimo. El objetivo aquí es conseguir una buena combinación de prestaciones y precio. Los servidores hacen las veces de los antiguos *mainframes*. Los servidores forman parte de la infraestructura básica de una corporación, ofreciendo diversos servicios (almacenamiento, correo, impresión, etc.) dentro de una red. En estos sistemas prima la elevada productividad y la escalabilidad. Los grandes servidores ofrecen sus servicios a miles de usuarios a través de Internet. Ejemplos de estos servicios son los buscadores, las redes sociales, la compra *on-line* o los servicios de compartición de vídeos. Estos sistemas están basados en clústeres, formados por una colección de computadores conectados en red y que se comportan externamente como un gran computador. Además de la capacidad de procesamiento y el coste de la instalación, es muy importante ofrecer una alta disponibilidad, puesto que un fallo en el sistema puede suponer pérdidas económicas muy elevadas. También es de vital importancia el coste de explotación. Finalmente, los supercomputadores son máquinas diseñadas para obtener unas prestaciones muy elevadas. El coste, aunque siempre importante, es un factor secundario. Estas máquinas ejecutan aplicaciones científicas con poca interacción con el usuario y con un elevado tiempo de ejecución, requiriendo unas elevadas prestaciones en cálculos de coma flotante. Cada vez más frecuentemente, los supercomputadores también están basados en clústeres.

Este trabajo describe las diferentes alternativas que se han utilizado a lo largo del tiempo para implementar los sistemas informáticos. Comenzaremos repasando la forma de medir las prestaciones. En relación a los sistemas compuestos por un único procesador, se enumeran las diferentes técnicas que han permitido mejorar notablemente sus prestaciones. Seguidamente, se describen los supercomputadores vectoriales como una alternativa para ejecutar aplicaciones que trabajan con vectores y matrices. A continuación, se presentan los sistemas multiprocesadores. Finalmente, se presentan los clústeres como la configuración utilizada actualmente para los grandes sistemas informáticos y supercomputadores.

MEDIDA DE LAS PRESTACIONES

Un usuario de un computador personal considerará que una máquina es más rápida que otra si el tiempo de ejecución de un determinado programa es menor. Sin embargo, el administrador de un servidor indicará que la máquina mejor es la que atiende más peticiones o realiza más transacciones por unidad de tiempo. En realidad, ambas medidas están relacionadas, siendo una la inversa de la otra. El problema reside, sin embargo, en decidir cuál es la carga de trabajo a considerar para poder comparar las diferentes alternativas.

Inicialmente, teniendo en cuenta que todas las instrucciones tenían un tiempo de ejecución similar, se utilizaba el tiempo necesario para realizar una operación individual.

Conforme los tiempos de ejecución de las instrucciones se hicieron diferentes, se introdujeron las mezclas de instrucciones, en las que cada instrucción tenía un peso diferente basado en su frecuencia relativa de aparición. La inversa de esta métrica dio lugar a los MIPS (Millones de Instrucciones Por Segundo). Posteriormente, se utilizaron *benchmarks* basados en *kernels* (fragmentos de programas reales) y *benchmarks* sintéticos (programas cortos que se suponen representativos de las aplicaciones reales). Motivado por la aparición de los supercomputadores en las décadas de los 70 y 80 y teniendo en cuenta la importancia de los cálculos en coma flotante, se introdujeron los MFLOPS (Millones de Operaciones en Coma Flotante por Segundo), todavía muy empleados en las aplicaciones científicas. Sin embargo, fue a finales de los 80 cuando se propuso la utilización de conjuntos de programas para medir las prestaciones (por ejemplo *SPEC*, *TPC*). Se trata de programas reales, que se actualizan periódicamente con el objeto de ser representativos de los hábitos de los usuarios en cada momento.

La lista *top500* (www.top500.org) apareció en 1993 como consolidación de la *Manheim supercomputer statistics* que desde 1986 se presentaba anualmente. La lista pretende reflejar las características y ubicación de los 500 supercomputadores más rápidos del mundo. Las máquinas están ordenadas en sentido decreciente de capacidad de cómputo, utilizando para ello los MFLOPS alcanzados ejecutando el *benchmark Linpack*. Este *benchmark* resuelve un sistema de ecuaciones lineales mediante una descomposición LU con pivotamiento parcial. La lista se publica dos veces al año. Resulta muy ilustrativo analizar la evolución de la lista a lo largo del tiempo. En primer lugar, el enorme aumento en las prestaciones: desde los 59,7 GFLOPS (1 GFLOPS= 10^3 MFLOPS) de la CM-5 (1993) hasta los 33,86 PFLOPS (1 PFLOPS= 10^9 MFLOPS) del actual número uno, el Tianhe-2 (noviembre de 2014). Por otra parte, analizando la evolución de los tipos de sistemas, podemos observar que la mayoría de supercomputadores actuales están basados en clúster.

Por otra parte, desde hace algún tiempo, existe una creciente preocupación por el consumo de energía. Durante décadas, los supercomputadores han alcanzado muy altas prestaciones a expensas de un consumo desmedido de energía, no solo de la propia instalación, sino también de los enormes sistemas de refrigeración necesarios para su correcto funcionamiento. La lista *green500* (www.green500.org) nació en abril de 2005 para tomar en consideración no solo la potencia de cálculo, sino también el consumo de energía, en igualdad de importancia. Dicha lista utiliza, como criterio de ordenación, las prestaciones por unidad de potencia (FLOPS por vatio). Un recorrido temporal de la lista muestra una evolución desde los 357,53 MFLOPS/W del BlueGene/P (2007) hasta los 5.271,81 MFLOPS/W del L-CSC, el número uno en noviembre de 2014, que ocupa el puesto 168 de la lista *top500*.

SISTEMAS MONOPROCESADOR

La capacidad de cómputo de los procesadores ha ido aumentando notablemente a lo largo del tiempo. Buena parte de las mejoras se explica por los avances tecnológicos. La Ley de Moore indica que el número de transistores que puede alojar un circuito integrado se duplica cada 18 meses. Dicha ley tiene su base, por una parte, en la reducción del tamaño del transistor, y por otra, en el aumento del tamaño de los circuitos integrados. Una consecuencia de la disminución del tamaño del transistor es la reducción de su retardo, lo que redundará en un aumento de su velocidad. Por otra parte, la disponibilidad de más transistores permite la implementación de ingeniosas innovaciones en el diseño. Ambos efectos combinados, mejoras tecnológicas y arquitectónicas, han permitido mejoras anuales entre el 25% y el 52%.

La segmentación es una de las primeras técnicas utilizadas para mejorar las prestaciones de los procesadores. La idea es conceptualmente bastante sencilla. Segmentar un proceso consiste en descomponerlo en varios subprocesos, así como permitir ejecutar cada uno de ellos de forma independiente y simultánea. Al segmentar un proceso, se obtiene una mejora que puede llegar a ser igual al número de etapas de segmentación. En el caso de los procesadores, se segmenta el ciclo de instrucción, esto es, el proceso de buscar una instrucción, ejecutarla y escribir su resultado. Se considera el IBM 7030 (denominado *Stretch*, 1961) el primer procesador segmentado de propósito general. Sin embargo, la segmentación se topa con varios problemas. El primero de ellos es la existencia de dependencias entre las instrucciones. Es muy frecuente que el operando de una instrucción sea el resultado producido por una instrucción anterior. Al segmentar se solapa la ejecución de las dos instrucciones, de modo que la segunda puede encontrar que su operando todavía no ha sido generado. Si bien el problema tiene solución, es necesario añadir circuitería y/o modificar el compilador, pero no siempre se resuelve sin penalización. Por otra parte, las instrucciones de salto, generadas por la compilación de estructuras de repetición y de selección de los lenguajes de alto nivel, producen una interrupción del flujo de instrucciones en el procesador, penalizando las prestaciones.

Los procesadores RISC, filosofía inaugurada a principio de la década de los 80, propugnan tomar en consideración el efecto de las decisiones de diseño sobre la implementación del procesador. Por ejemplo, utilizan un juego de instrucciones máquina que reduce notablemente la aparición de los problemas derivados de las dependencias y, por tanto, facilitan una segmentación eficiente. Desde 1986, la industria de los computadores ha producido procesadores basados en la filosofía RISC: HP Precision Architecture, IBM RS 6000, Sun SPARC, MIPS, DEC Alpha. Solamente la familia Intel 80x86 sobrevivió aparentemente al debate CISC/RISC. Sin embargo, internamente incorporan un núcleo RISC junto con hardware especializado en la conversión de instrucciones CISC a RISC.

La ejecución fuera de orden es otra de las mejoras introducidas en los procesadores. En un código en el que hay varias instrucciones inmersas en dependencias debe asegurarse

la ejecución secuencial de las mismas. Sin embargo, si hay instrucciones posteriores que son independientes podrían proceder sin necesidad de esperar. La ejecución fuera de orden permite que instrucciones posteriores puedan adelantar su ejecución a instrucciones anteriores en el orden de programa. Aparece así el concepto de ILP (*Instruction Level Parallelism*) o la posibilidad de solapamiento en las secuencias de instrucciones. El IBM 360/91 (1966) introdujo muchos de los conceptos relacionados con la ejecución fuera de orden. El algoritmo de Tomasulo (su diseñador), que permitía la ejecución fuera de orden de las instrucciones fue publicado en 1967. Sin embargo, la gran complejidad de la implementación, excesiva para la época, motivó su escaso éxito. En los años 90, cuando la tecnología de circuitos integrados permitía incorporar esta complejidad en una misma pastilla, muchas de estas ideas se introdujeron con éxito en los procesadores. En aquel momento, otra mejora, las memorias cache, descritas por Maurice Wilkes (1965), se incorporaron por primera vez en una máquina comercial, el IBM 360/85 (1969).

Para evitar la interrupción en el flujo de instrucciones originada por las instrucciones de salto, las técnicas de predicción dinámica de saltos tratan de adivinar cómo va a ser el comportamiento de los saltos a partir de su comportamiento previo. Una vez conocido cuál va a ser el destino de un salto, el paso siguiente es comenzar a ejecutar instrucciones a partir de esa dirección, antes de saber si finalmente se ha tenido éxito en la predicción o no. Estas instrucciones deben ejecutarse de forma especulativa, puesto que la predicción realizada puede fallar. La especulación tiene sus raíces en el IBM 360/91.

Los procesadores superescalares pretenden mejorar aún más las prestaciones buscando y lanzando a ejecución varias instrucciones en el mismo ciclo de reloj. La mayor complicación reside en la decodificación y detección de dependencias entre instrucciones. Sin embargo, algunos programas no exhiben suficiente ILP para aprovechar el potencial del procesador superescalar. Los procesadores SMT (*simultaneous multithreading*) ejecutan varias instrucciones por ciclo de reloj, de uno o varios hilos lanzados a ejecución. Por otra parte, si bien la disponibilidad de más transistores cada vez más rápidos ha permitido la incorporación de sofisticadas técnicas en los procesadores y el funcionamiento a elevadas frecuencias de reloj, los diseños se han enfrentado al problema de la disipación del calor generado. Por este motivo, a partir del año 2004, los fabricantes comenzaron a aprovechar la disponibilidad de transistores introduciendo múltiples *cores* o núcleos de procesamiento en la misma pastilla de circuito integrado, de manera que la mejora de prestaciones ha de buscarse a través de la ejecución de tareas en paralelo.

Todas estas técnicas han ido incorporándose paulatinamente en los procesadores comerciales. Los primeros procesadores superescalares aparecen en la década de los 90. En la segunda mitad de dicha década, los procesadores son superescalares y aplican ejecución fuera de orden con especulación. En la primera mitad de la década del 2000, los diseños alcanzan mayores frecuencias de reloj mediante una segmentación profunda, empiezan a ofrecer *multithreading* y a integrar varios núcleos en la misma pastilla.

COMPUTADORES VECTORIALES

Algunos de los problemas para los que se requiere una gran potencia de cálculo operan sobre vectores y matrices. Estos problemas exhiben unas características que motivaron el diseño e implementación de procesadores vectoriales: repiten el mismo procesamiento a todas las componentes de la estructura y se accede a la estructura completa (el vector) en la memoria. Pues bien, los procesadores vectoriales incorporan en su juego de instrucciones operaciones vectoriales, con las siguientes consecuencias: a) Una instrucción vectorial genera gran cantidad de trabajo, una operación por cada elemento del vector. b) El cálculo de cada elemento del resultado es independiente del resto, por lo que no hay dependencias en el cálculo de los elementos del vector resultado. c) Una instrucción vectorial sustituye a un bucle de instrucciones escalares. d) El patrón de acceso a memoria para buscar el vector completo es conocido con antelación. Todo ello contribuye a la obtención de elevadas potencias de cómputo.

Dos son las alternativas propuestas para ejecutar las instrucciones vectoriales. La primera es la utilización de múltiples unidades aritméticas bajo una única unidad de control. Aparecen así las máquinas SIMD (*Single Instruction Multiple Data*). Estas máquinas utilizan una matriz de unidades de proceso que realizan la misma operación, pero cada una sobre su propio dato, un elemento del vector. Además requieren una red de interconexión para poder intercambiar información entre los elementos de proceso. El ejemplo lo constituye el supercomputador Illiac IV (1972), inicialmente planeado para disponer de 256 elementos de proceso de 64 bits, y que finalmente solo contó con 64.

La otra alternativa la constituyen los computadores vectoriales segmentados. Estas máquinas basan la ejecución de instrucciones vectoriales en la utilización de unidades aritméticas profundamente segmentadas. En efecto, la ausencia de dependencias en el cálculo de las componentes de los vectores permite una segmentación en muchas etapas, permitiendo una notable mejora de prestaciones. El supercomputador Cray-1 (1976) empleaba una arquitectura basada en el uso de registros vectoriales, capaces de almacenar un vector completo y cuyo contenido se trae o lleva de la memoria por medio de instrucciones de carga y almacenamiento, de forma similar a los procesadores escalares.

SISTEMAS MULTIPROCESADOR

Un sistema multiprocesador está compuesto por varios procesadores que cooperan en la ejecución de una tarea. Idealmente, la ejecución de una tarea por parte de varios procesadores debería reducir su tiempo de ejecución linealmente con el número de procesadores. Sin embargo, en la práctica, las subtarefas que se ejecutan en paralelo tienen dependencias entre ellas, que motivan que deban coordinarse y sincronizarse. Para ello, se necesita un mecanismo de comunicación entre los procesadores del sistema. La clasificación tradicional de los multiprocesadores los divide en función de la implementación de dicho

mecanismo en multiprocesadores de memoria compartida y multiprocesadores de memoria distribuida.

En los multiprocesadores de memoria compartida la memoria del sistema está compartida, accesible por todos los procesadores. La comunicación se realiza por medio de variables compartidas, ubicadas en la memoria compartida. Así pues, es necesaria una red de interconexión entre los procesadores y la memoria compartida. El C.mmp (1970) estaba compuesto por 16 PDP-11 conectados a 16 unidades de memoria a través de un *crossbar*. El hecho de compartir la memoria introduce un cuello de botella, tanto mayor cuanto más procesadores haya. Por otra parte, la red de interconexión incrementa el tiempo de acceso a memoria. La utilización de memorias cache en cada uno de los procesadores permite filtrar muchos de los accesos a memoria, reduciendo la latencia y la presión sobre ella. Sin embargo, introduce el problema de la coherencia entre las memorias cache. Una misma variable puede estar ubicada en varias memorias cache; en el momento en que uno de los procesadores la escriba, el resto de copias, incluyendo la memoria central, quedan obsoletas. Los protocolos de coherencia, de invalidación y actualización, resuelven el problema. El Synapse N+1 (1984) fue el primer multiprocesador con una interconexión basada en bus con coherencia cache.

Por otra parte, en los multiprocesadores de memoria distribuida, inicialmente denominados multicomputadores, la memoria del sistema está distribuida entre los procesadores, de manera que cada uno de ellos puede acceder a su memoria local a la máxima velocidad, sin penalización alguna. La comunicación se realiza intercambiando mensajes entre los procesadores. Para ello se requiere una red de interconexión entre procesadores. Las prestaciones del sistema dependen, en gran medida, de las prestaciones de la interconexión. Los multiprocesadores de memoria distribuida son más escalables que los de memoria compartida, habiéndose construido sistemas de gran tamaño, lo que ha motivado que también se les conozca como MPP (*Massively Parallel Processors*). El Cosmic Cube (1985), que incorporaba importantes avances en la red de interconexión, fue uno de los primeros multicomputadores operativos. Muchas de las máquinas que han ocupado el número 1 de la lista *top500* responden a esta arquitectura. La CM-5 (primer puesto en junio de 1993, 59,7 GFLOPS), tenía 1024 procesadores interconectados por una red *fat-tree*. El Intel Paragon (número uno en junio de 1994, 143,4 GFLOPS) tenía 3680 procesadores interconectados por una red de pocas dimensiones. Ejemplos más recientes son el IBM BlueGene/L (número uno durante tres años consecutivos, noviembre de 2004 a noviembre de 2007, hasta 478,2 TFLOPS) con 212.992 procesadores interconectados mediante un toro 3D. Los multiprocesadores de memoria distribuida actuales son híbridos en el sentido en que los nodos son sistemas de memoria compartida compuestos por varios procesadores. De hecho, desde el año 2012 es habitual que cada procesador contenga varios *cores* o núcleos de procesamiento. Por ejemplo, el *Sequoia* IBM BlueGene/Q (primer puesto en junio de 2012, 16,32 PFLOPS) tiene 98.304 procesadores, cada uno con 16 núcleos, ofreciendo un total de 1.572.864 núcleos interconectados mediante un toro 5D.

CLÚSTERES DE COMPUTADORES

Un clúster está compuesto por una colección de computadores, interconectados, que comparten una infraestructura hardware y software común y que se comportan como si fuesen un único computador, proporcionando un conjunto de recursos paralelos a servicios o aplicaciones, para obtener unas prestaciones (HPC, *High Performance Computing*) y/o una disponibilidad (HA, *High Availability*) muy altas.

El diagrama de bloques de un clúster recuerda al de un sistema multiprocesador de memoria distribuida. En efecto, está compuesto por múltiples elementos de procesamiento, cada uno con su memoria (cada uno de los nodos computadores) interconectados mediante una red de interconexión (la red local que interconecta los nodos). La SSI (*Single System Image*) se ocupa de presentar a los usuarios un conjunto de recursos como un recurso único, pero mucho más potente.

Aunque la utilización de clústeres para mejorar la fiabilidad data de la década de los 60, su utilización para la obtención de elevadas prestaciones es más reciente. El proyecto *Beowulf* (1993) pretendía construir un computador capaz de alcanzar 1 GFLOPS por menos de 50.000 dólares. Un clúster de 16 nodos contruidos con computadores personales convencionales basados en el procesador Intel 486 cumplió el objetivo. Desde entonces, los clústeres se han convertido en una alternativa de bajo coste para construir máquinas de altas prestaciones.

Los clústeres utilizan componentes de tipo estándar, por lo que pueden aprovechar rápidamente las mejoras tecnológicas. El hecho de no estar sujetos a una única fuente (un fabricante, una tecnología) evita los problemas ocasionados por su desaparición. Por otra parte, pueden configurarse para satisfacer unas necesidades concretas, son escalables y fácilmente expandibles. Como software del sistema, causa del fracaso de muchos sistemas, emplean un sistema operativo convencional.

En un clúster HPC podemos ejecutar desde múltiples aplicaciones secuenciales hasta aplicaciones paralelas. En el primer caso, los clústeres incorporan sistemas de gestión de trabajos que equilibran la carga entre sus recursos. El usuario lanza sus aplicaciones desde un único punto de entrada y el sistema se encarga de repartir la carga entre los servidores disponibles. Por otra parte, en lo relativo a las aplicaciones paralelas, teniendo en cuenta que el clúster se compone de múltiples nodos, y que cada nodo puede incorporar uno o varios procesadores, actualmente multinúcleo, la mejor forma de sacarle partido es la utilización de un modelo de programación híbrido: paso de mensajes entre nodos y variables compartidas dentro de un nodo.

Un clúster simple de alta productividad está compuesto por un nodo maestro, múltiples nodos servidores y un servidor de almacenamiento. Todos los componentes se interconectan mediante una red de datos. El nodo maestro acepta los trabajos de los usuarios y realiza funciones de equilibrado de carga. El almacenamiento puede componerse también de varios nodos, a los que se conectan varios discos a través de una red SAN (*Storage Area Network*), una red por la que circulan bloques de disco. Este clúster podría

realizar las funciones, asimismo, de un servidor web de alta disponibilidad con equilibrado de carga. Varios nodos maestros (en este caso nodos directores) aseguran un funcionamiento continuo (alta disponibilidad) incluso en caso de fallo de alguno de ellos, y reparten la carga entre los servidores disponibles. El funcionamiento continuo es de vital importancia en los grandes servidores, puesto que una interrupción del servicio produce grandes pérdidas económicas. Habitualmente nos referimos a la disponibilidad de un sistema indicando el porcentaje del tiempo que está disponible. Se ha popularizado el modelo de los “nueves”. Por ejemplo, un sistema con una disponibilidad del 99,9%, 99,99% y 99,999% tiene unos tiempos sin servicio de 9 horas, 52 minutos y 5 minutos al año, respectivamente.

Un clúster de altas prestaciones tendrá, además de una red de datos, una red de altas prestaciones, capaz de ofrecer una baja latencia en las comunicaciones de las aplicaciones paralelas, a la vez que una alta productividad. En este sentido, en los últimos años ha habido grandes avances en esta línea, habiéndose comercializado redes de interconexión de alto rendimiento como, por ejemplo, InfiniBand.

La excelente relación entre precio y prestaciones de los clústeres ha originado que esta arquitectura se haya popularizado notablemente en los últimos años. Su gran flexibilidad de configuración los hace idóneos para múltiples aplicaciones, desde servidores en pequeñas y medianas empresas y corporaciones hasta grandes servidores de Internet. Los servicios de Internet que han alcanzado tanta popularidad los proporcionan sistemas basados en clústeres a gran escala, con instalaciones de gran tamaño para proporcionar la potencia de cómputo necesaria, incluso repartidas en diversos lugares para tolerar problemas derivados de catástrofes naturales. Por otra parte, los grandes supercomputadores también están, cada vez más frecuentemente, basados en clústeres. Desde el año 2007, más del 80% de las máquinas de la lista *top500* son clústeres. Por citar algunos ejemplos, el Tianhe 1A (primer puesto en noviembre de 2010, 2,57 PFLOPS), primera vez que China ocupa la primera posición, se compone de 14.336 procesadores Intel Xeon más 7.168 aceleradores basados en GPU NVIDIA Tesla. El K Computer (número uno en las dos ediciones de 2011 con 10,51 PFLOPS) está compuesto por 705.024 núcleos SPARC64 o el Tianhe 2 (primer puesto en noviembre de 2014, 33,86 PFLOPS), con 16.000 nodos, cada uno con dos procesadores Intel Ivy Bridge Xeon y tres aceleradores Intel Xeon Phi, arrojando un total de 3.120.000 núcleos.

BIBLIOGRAFÍA

CULLER, D.; SINGH, J. P. (1998), *Parallel Computer Architecture: A Hardware/Software Approach*, Morgan Kaufman.

HENNESSY, J; PATTERSON, D. (2012), *Computer Architecture: A Quantitative Approach*, Morgan Kaufman.

HWANG, K; BRIGGS, F. A. (1981), *Arquitectura de Computadoras y Procesamiento Paralelo*, Mc Graw-Hill.

MEINADIER, J. P. (1973), *Estructura y Funcionamiento de los Computadores Digitales*, Editorial AC.

PFISTER, G. (1998), *In Search of Clusters*, Prentice-Hall PTR.

Antonio Martí Campoy
Universitat Politècnica de València

INTRODUCCIÓN

Decir que los ordenadores forman parte de nuestra vida, que están en todas partes y que participan en muchas de las actividades que realizamos a diario, no es decir nada nuevo. Desde que Apple fabricó el Apple II en 1977 e IBM fabricó su IBM PC en 1981, los ordenadores no han dejado de proliferar a nuestro alrededor. Empezamos a verlos en las oficinas de los bancos, en las universidades, institutos y colegios, en las oficinas de las grandes empresas y también de las pequeñas, en la administración pública, en nuestras casas. Y con el paso de los años empezamos a verlos en todas partes.

Todo el mundo es capaz de señalar y decir “eso es un ordenador”. Cualquiera que vea un ordenador personal es capaz de identificarlo. Incluso más allá de “pantalla, teclado, ratón y caja”, mucha gente intuye que aunque no tenga forma de ordenador personal, es un ordenador. Puede que usted lo sepa, lo intuya, lo sospeche, o quizá nunca se lo haya planteado, pero los cajeros automáticos, las cajas de los supermercados, las tabletas e incluso los teléfonos móviles y las videoconsolas son ordenadores. En conclusión, si vemos un ordenador, somos capaces de saber, o al menos sospechar que lo es.

Pero ¿qué pasa cuando no lo vemos? La respuesta es sencilla: si no lo vemos no podemos saber que está allí. Y si no sabemos que está, tampoco podemos saber que es un ordenador. Pues de eso trata este texto, de los ordenadores que están pero no se ven.

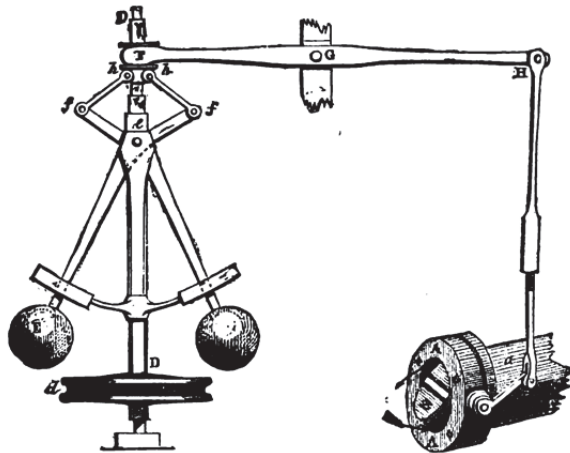
CONTROLAR

Desde la prehistoria la humanidad ha desarrollado herramientas y máquinas que le han servido para superar sus limitaciones físicas. La lista de ingenios que el hombre ha desarrollado es enorme, comenzando por las puntas de sílex, pasando por azadas, arados, poleas o herramientas como los tornos y taladros¹, y barcos, y molinos, y la imprenta, y muchos más, y llegando hasta la máquina de vapor, y los motores eléctricos y de combustión interna. Todo ello con el objetivo de obtener mayor trabajo con menor esfuerzo.

¹ Muchas de las herramientas y máquinas que se utilizan en la actualidad existen desde hace milenios, como los tornos y los molinos. El principal cambio que han sufrido a lo largo del tiempo es el origen de la fuerza que mueve estas herramientas, siendo humana, animal o aprovechando la naturaleza, como el viento y el flujo de los ríos.

Pero todas estas invenciones, para ser útiles y productivas, necesitan un gobierno, algo que las controle para que operen de forma que obtengamos el beneficio que queremos. Al principio y durante mucho tiempo este control se realizó por la inteligencia del hombre a través de sus manos. Pero, si se pudo sustituir el músculo humano por una máquina o ingenio, ¿por qué no sustituir también la inteligencia?

De este modo aparecen nuevos dispositivos cuyo objetivo directo no es producir un trabajo o un rendimiento aprovechable por el hombre, sino gobernar y controlar a otra máquina para liberar al hombre de dicha tarea. Existen algunos ejemplos históricos que se remontan a los tiempos de Alejandría², pero podemos decir que los primeros reguladores modernos, los reguladores centrífugos, aparecen en el siglo XVII en los molinos de viento. Casi dos siglos después James Watt utiliza uno de estos reguladores centrífugos para su máquina de vapor, controlando y regulando la presión del vapor de forma automática (Figura 10.1). Un siglo después, el control es una nueva ciencia que evoluciona de forma constante y junto a las demás ciencias, todas ellas en una simbiosis que nunca había sido tan fecunda en la historia de la humanidad.



Fuente: “Discoveries & Inventions of the Nineteenth Century” by R. Routledge, 13th edition, published 1900

Figura 10.1. Regulador centrífugo (izqda.) y válvula reguladora (dcha.)

Los reguladores se diseñan y construyen de forma específica para el trabajo que deben realizar y pueden ser mecánicos, hidráulicos, neumáticos, eléctricos, electrónicos o una combinación cualquiera de ellos. Pero el punto en común que tienen todos ellos, independientemente de la tecnología utilizada y de su construcción física, es que se diseñan y fabrican para realizar una tarea determinada. Y si esta tarea cambia de forma sustan-

² En la Alejandría del siglo III a.C., el matemático Ktésibios construye un reloj de agua que es el primer sistema de control del que se tiene constancia.

cial, o queremos realizar una nueva tarea, el controlador no es capaz de adaptarse y deberá ser sustituido por un nuevo diseño *ad hoc*.

LOS ORDENADORES

La aparición de los ordenadores marca un punto de inflexión en el proceso de creación de máquinas para superar las limitaciones del hombre. Por dos motivos: el primero, porque ahora ya no se trata solo de limitaciones físicas, sino también intelectuales. A los primeros ordenadores se les llamaba máquinas pensantes, y aunque en realidad no lo son, están más cerca del pensamiento racional que de la fuerza física.

El otro motivo por el que la aparición de los ordenadores es importante es porque se trata de una máquina que no hace nada en concreto, excepto ejecutar una secuencia de órdenes, un programa³, sobre unos datos de entrada. Y en función de esa secuencia de órdenes, de ese programa, el ordenador servirá para escribir textos, realizar cálculos, jugar, o controlar una fábrica, un tren o la calefacción de casa. El ordenador no nace con un propósito concreto, sino con todas las posibilidades que su programador sea capaz de imaginar.

Esta versatilidad, flexibilidad y adaptabilidad que al igual que presentan los seres humanos ofrecen los ordenadores, llamó rápidamente la atención de los ingenieros de control. El computador era la máquina ideal para sustituir a la gran mayoría de los dispositivos de control existentes. Sus ventajas eran muchas y en varios aspectos. Por un lado, podía fabricarse un único modelo de ordenador de forma masiva, reduciendo por tanto su coste, e introducirle un programa diferente según el sistema o máquina que debiera controlar. También se reducían los costes de desarrollo, ya que si el programa no hacía lo que se esperaba, solo había que cambiar el programa e introducirlo en el mismo ordenador, sin necesidad de construir uno nuevo, como sucedía por ejemplo con los circuitos electrónicos, que una vez construidos no podían ser modificados con facilidad. Pero puede que su mayor ventaja sea la complejidad y sofisticación de los procesos y algoritmos de control que pueden desarrollarse y ponerse en práctica usando un ordenador. Esta es una de las llaves que ha abierto la puerta al mundo actual, desde el punto de vista de la tecnología.

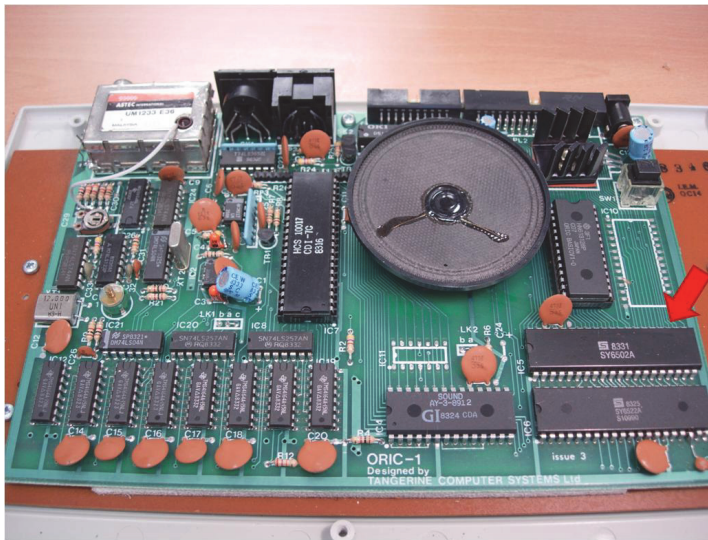
EL CIRCUITO INTEGRADO

Los primeros ordenadores eran máquinas gigantescas, que además consumían grandes cantidades de energía. Evidentemente estas máquinas no servían para controlar a otras máquinas, pues su coste de construcción y uso era superior al beneficio que producían. Pero todo era cuestión de tiempo. Los avances en electrónica permitieron reducir el tamaño de los componentes utilizados para construir los ordenadores reduciendo por tanto

³ Es común definir al ordenador como máquina de programa almacenado.

su coste, peso, tamaño y consumo. La aparición en 1959 del circuito integrado, llamado microchip o simplemente chip, hizo que los circuitos electrónicos con los que se construye un ordenador fueran mucho más baratos, pequeños, con menor consumo de energía y mucha más fiabilidad, transfiriendo estas ventajas a los ordenadores. La electrónica empezaba a convertirse en un producto de masas, y con ella los ordenadores.

Para construir un ordenador utilizando estos chips era necesario utilizar varios de ellos, del orden de decenas, para construir la unidad central de proceso, la memoria principal y el sistema de entrada y salida (Figura 10.2). Evidentemente, a mayor número de chips usados para construir un ordenador, más caro resultaba el mismo, no solo por el mayor número de chips, sino por el montaje y la interconexión entre ellos. Por ello, uno de los objetivos de las empresas fabricantes de circuitos era integrar tanto como se pudiera dentro de un microchip. Y así, recién estrenada la década de los 70 del siglo XX, nacieron casi a la vez el primer microprocesador y el primer microcontrolador comerciales. El primero de la mano de Intel. El segundo de la mano de Texas Instruments.

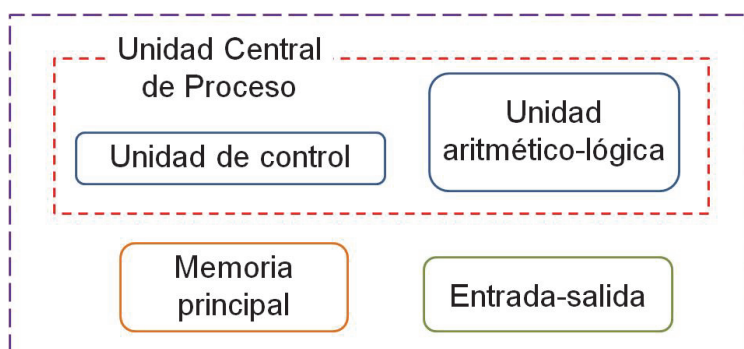


Fuente: Antonio Martí Campoy

Figura 10.2. Microordenador doméstico Oric 1, de 1985. El microprocesador es el chip indicado por la flecha roja. Los 8 chips idénticos de la parte inferior izquierda son la memoria principal (48 Kilobytes). La mayoría del resto de chips forman el sistema de entrada-salida

MICROCONTROLADOR Y MICROPROCESADOR

Para entender la diferencia entre un microcontrolador y un microprocesador es necesario dividir el ordenador en sus unidades funcionales básicas⁴ (Figura 10.3): la unidad de control, que gobierna a todos los demás elementos; la unidad aritmético-lógica (UAL o ALU en las siglas inglesas), encargada de realizar las operaciones aritméticas y lógicas; la memoria principal donde se encuentra almacenado el programa que se está ejecutando y todos los datos que dicho programa maneja; y por último, la entrada-salida, que permite introducir y obtener datos en y desde el ordenador.

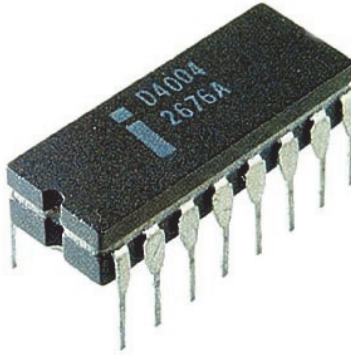


Fuente: Antoni Martí Campoy

Figura 10.3. Diagrama de bloques de un ordenador

Cuando se habla conjuntamente de la unidad de control y de la unidad aritmético-lógica se le da el nombre de procesador o Unidad Central de Proceso (UCP o CPU de *Central Process Unit*). Y cuando estas dos unidades se integran en un microchip, se le da el nombre de microprocesador. El primer microprocesador comercial de la historia fue el Intel 4004 (Figura 10.4) que apareció en 1971.

⁴ La división de un ordenador en bloques funcionales depende mucho del gusto del autor. El número de bloques puede ser mayor o menor en función del nivel de agrupamiento elegido por el autor, ya que no existe un criterio único.



Fuente: Luca De Tomi

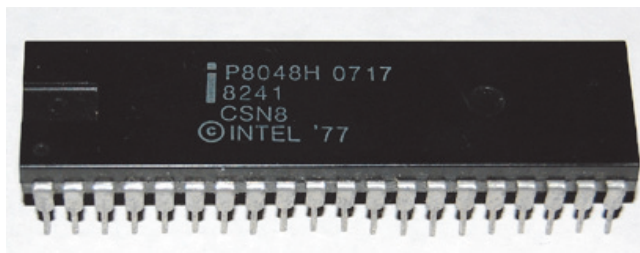
Figura 10.4. Intel 4004, primer microprocesador comercial

En el microcontrolador todas las unidades funcionales se integran en un solo microchip. La unidad de control y la unidad aritmético-lógica, la memoria principal y la entrada-salida, es decir, todo el ordenador, se construye dentro de un único microchip y recibe el nombre de microcontrolador. El primer microcontrolador comercial de la historia fue el TMS 1000 (Figura 10.5) de Texas Instruments, comercializado en 1974. Solo un año después, Intel comercializó su primer microcontrolador, el 8048.



Fuente: Antonio Martí Campoy

Figura 10.5. Texas Instruments TMS 1000, el primer microcontrolador comercial



Fuente: Rhondle

Figura 10.6. Microcontrolador Intel 8048

Inicialmente el microcontrolador recibió el nombre de microordenador, ya que efectivamente se trataba de un ordenador en un microchip. Sin embargo, el tiempo hizo que el término microordenador se utilizara para llamar a los ordenadores que utilizaban un microprocesador como unidad central de proceso, para distinguirlos de sus predecesores en los que la unidad de control y la unidad aritmético-lógica estaban separadas en dos o más chips o contruidos con componentes electrónicos discretos.

Aunque tanto el microprocesador como el microcontrolador nacen al mismo tiempo, con prestaciones parecidas y por las mismas razones comentadas anteriormente, pronto su evolución los separará significativamente.

Los microprocesadores crecerán en potencia, es decir, en la capacidad de ejecutar instrucciones por unidad de tiempo y en la cantidad de memoria que pueden manejar, adaptándose así a cualquier aplicación de propósito general: edición de textos, hojas de cálculo, cálculo científico, videojuegos, reproducción y creación de música y vídeo y casi cualquier aplicación que se nos ocurra. Así, un ordenador personal de 1980 podía tener 16 Kilobytes de memoria principal y ejecutar medio millón de instrucciones por segundo (0,5 MIPS), mientras que los ordenadores que solemos utilizar hoy en día tienen alrededor de 4 Gigabytes de memoria principal (1 Gigabyte = 10^6 Kilobytes) y pueden ejecutar decenas de miles de millones de instrucciones por segundo (>50.000 MIPS). El objetivo, manejar mayor cantidad de información y de forma mucho más rápida.

Los microcontroladores, por el contrario, fijan su campo de aplicación en el control de sistemas, donde una máquina, sistema o ingenio integra un ordenador⁵, en la mayoría de los casos en forma de microcontrolador. Para realizar el control de estos sistemas una parte importante, podríamos decir que crítica, es la interfaz, es decir, la forma en que se realiza el intercambio de información entre el programa y el sistema gobernado. Lo que se espera de un microcontrolador es diferente de lo que se espera de un ordenador de propósito general y por ello los microcontroladores siguen una evolución diferente a los microprocesadores.

Para controlar un sistema es necesario conocer el estado actual del sistema, calcular mediante una fórmula o algoritmo la acción necesaria a aplicar sobre el sistema, y aplicar dicha acción, repitiendo este proceso periódicamente. Un ejemplo muy sencillo es el control de la temperatura de una habitación durante el invierno, donde es necesario medir la temperatura de la habitación, compararla con la que el usuario quiere, y encender o apagar la calefacción, repitiendo este proceso o algoritmo dos veces por segundo, por ejemplo.

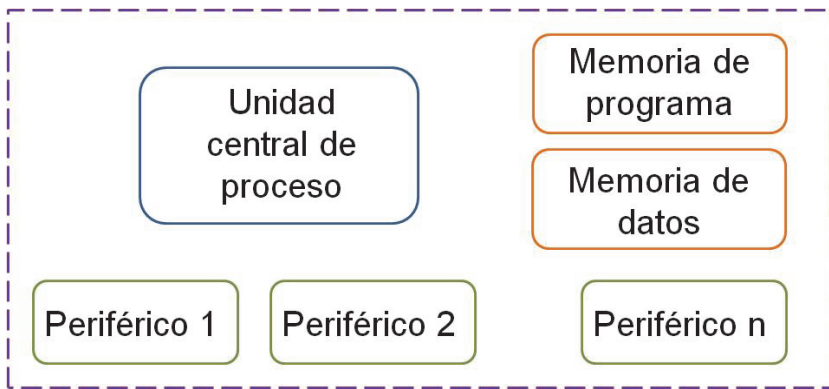
Para almacenar la temperatura de referencia no se necesita mucha memoria, tan solo una docena de bytes. Y para calcular si se enciende o se apaga la calefacción no hace falta ejecutar muchos millones de instrucciones por segundo. Pero para medir la tempe-

⁵ Una máquina o sistema que utiliza un ordenador para su gobierno recibe el nombre de sistema empotrado o embebido, independientemente de si el ordenador es un microcontrolador, microordenador u otra variante de ordenador.

ratura y actuar sobre la calefacción hacen falta dispositivos de entrada-salida especiales, que no es normal encontrar en un ordenador personal. Y aquí es donde los microcontroladores marcan la diferencia.

EL MICROCONTROLADOR EN DETALLE

Un microcontrolador se puede dividir en cuatro bloques: unidad central de proceso (CPU), encargada de ejecutar las instrucciones; memoria de programa, donde se almacena de forma permanente el programa que realiza el control del sistema; memoria de datos, volátil⁶, utilizada para almacenar los valores medidos del sistema y los resultados de los cálculos y algoritmos; y un conjunto de periféricos que permiten obtener el estado del sistema a controlar y actuar sobre dicho sistema. A continuación se describen los bloques que componen un microcontrolador genérico (Figura 10.7).



Fuente: Antonio Martí Campoy

Figura 10.7. Diagrama de bloques de un microcontrolador

Unidad central de proceso: la unidad central de proceso de un microcontrolador es similar a lo que podemos encontrar en un microprocesador, pero normalmente no presenta la misma complejidad ni las mismas prestaciones que la de un microprocesador. En primer lugar porque no es necesario. La mayoría de sistemas empotrados son sencillos y no necesitan una gran potencia de cómputo. Pero si el sistema que se quiere controlar es realmente complejo, se utilizan varios microcontroladores que cooperan entre ellos controlando diferentes partes del sistema, lo que recibe el nombre de sistema distribuido. Dividir el problema en pequeñas partes hace que sea más fácil de resolver.

⁶ Una memoria volátil pierde su contenido cuando se desconecta la alimentación eléctrica, es decir, cuando se apaga el dispositivo.

La segunda razón por la que no pueden competir en prestaciones con los microprocesadores es porque estos últimos utilizan todo el espacio disponible en el microchip para ser más rápidos y potentes, mientras que los microcontroladores deben dejar espacio libre para la memoria y especialmente para los periféricos. Como ejemplo, el microcontrolador Intel 8051 comercializado en 1980 no llegaba a 1 MIPS, mientras que las versiones actuales pueden alcanzar los 400 MIPS. Para aquellas aplicaciones empotradas donde es necesaria una gran potencia de cómputo, es posible encontrar microcontroladores basados en la arquitectura ARM Cortex que llegan a los 2.000 MIPS. Aún así, muy por debajo de las prestaciones de los microprocesadores de propósito general.

Memoria de programa: cada vez que un microcontrolador se pone en marcha debe encontrar un programa en memoria principal para ejecutarlo. Esto es posible gracias al uso de memorias no volátiles, como las memorias ROM, EEPROM o las modernas FLASH. Estas memorias no pueden reescribirse, o al menos no de forma trivial. Pero esto no es un problema, ya que al contrario que en los ordenadores personales, los sistemas empotrados se programan con una aplicación que no es necesario cambiar durante el funcionamiento normal. Los programas de control no suelen ser muy grandes, por lo que la memoria disponible en la mayoría de los microcontroladores oscila entre los 4 y los 256 Kilobytes, aunque es posible encontrar algunos modelos con unos pocos Megabytes. Como comparación entre las aplicaciones de control y una aplicación de oficina común, la versión del procesador de textos Microsoft Word para el Apple iPhone ocupa más de 250 Megabytes (1 Megabyte = 10^3 Kilobytes).

Memoria de datos: al contrario que el programa que ejecuta el microcontrolador que no suele cambiarse nunca⁷, los datos que se recogen del sistema y los resultados que se obtienen de la ejecución del programa sí cambian, por lo que se almacenan en una memoria RAM. Este tipo de memoria es volátil, pero permite tanto leer como escribir datos fácilmente. Los primeros microcontroladores disponían de unos pocos bytes, entre 128 y 512, para almacenar las variables del sistema. Aunque se siguen fabricando y utilizando en muchas aplicaciones modelos con muy poca memoria de datos, la aparición de máquinas y sistemas cada vez más complejos y sofisticados y las mejoras en la tecnología de integración de circuitos ha propiciado la aparición de microcontroladores con centenares de Kilobytes e incluso algunos Megabytes de memoria de datos.

Periféricos: cada modelo de cada fabricante incluye un conjunto de periféricos que intentan dar respuesta a diferentes necesidades del control de sistemas. Aunque la variedad de periféricos es muy grande, hay cinco o seis que son los más comunes y que se encuentran en casi todos los modelos de microcontroladores: interrupciones⁸, que

⁷ En caso de mejora o actualización del sistema, es posible cambiar el programa de un microcontrolador, pero normalmente se desecha todo el circuito y se instala uno nuevo.

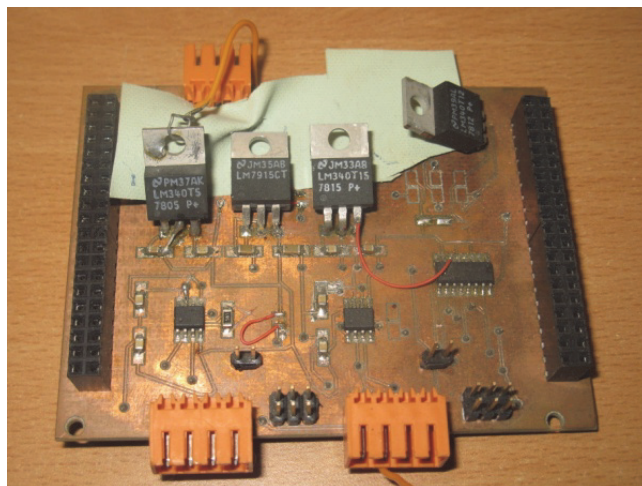
⁸ Los microprocesadores también disponen de interrupciones, pero en los microcontroladores suelen ser más numerosas y con un papel protagonista.

permiten que el programa responda de forma eficiente a eventos externos que se producen de forma asíncrona; temporizadores, dispositivos que permiten medir el tiempo y generar eventos periódicos. Posiblemente el temporizador sea el periférico más útil, pues el tiempo es fundamental en la mayoría de los sistemas y procesos que utilizamos y que encontramos a nuestro alrededor; entradas y salidas digitales, que permiten obtener información del exterior y actuar sobre el sistema que gobierna mediante señales del tipo sí/no o encendido/apagado; convertidores de analógico a digital y de digital a analógico, que traducen el mundo analógico al digital, permitiendo al microcontrolador manejar información continua como la temperatura; y las comunicaciones, que permiten al microcontrolador intercambiar información con otros microcontroladores, con otros sistemas informáticos y con decenas de periféricos externos.

USANDO UN MICROCONTROLADOR

El primer paso para usar un microcontrolador es elegir el modelo que se necesita. Hay más de dos decenas de fabricantes de microcontroladores que entre todos pueden ofrecer centenares de modelos de microcontroladores, con diferentes características en cuanto a potencia de cálculo, memoria disponible y, sobre todo, una gran variedad en los periféricos incluidos en cada modelo. Elegir el más grande, el más potente y el que tenga todos los periféricos posibles sería una opción fácil, pero posiblemente demasiado cara. No hay que olvidar que el microcontrolador nace, entre otros motivos, para reducir costes. Y que en muchas aplicaciones se trata de productos de consumo de masas, por lo que cuanto más se ahorre en cada unidad, más beneficio se obtendrá. Así pues, se debe elegir el microcontrolador más ajustado a las necesidades del sistema que se quiere controlar.

Una vez elegido el microcontrolador hay que construir un circuito impreso donde soldarlo y añadirle algunos e incluso muchos componentes electrónicos. Por ejemplo, podemos diseñar un sencillo prototipo para medir las salidas analógicas de dos sensores, uno de posición y otro de presión, y enviar la información a través de una red de comunicaciones (Figura 10.8). En la actualidad es fácil encontrar equipos de desarrollo o evaluación con todos los componentes soldados y algunos periféricos como botones y luces led para poder hacer pruebas durante el desarrollo (Figura 10.9).



Fuente: Antonio Martí Campoy

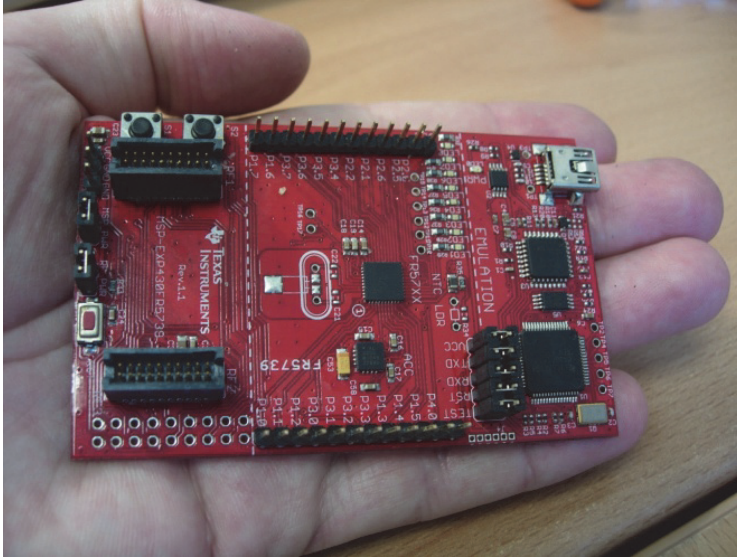
Figura 10.8. Prototipo para la adquisición de señales analógicas y comunicación por CAN. Los cables rojos son correcciones a errores en el diseño hardware

Una vez se dispone del hardware, bien un prototipo, un equipo de evaluación o el diseño definitivo, hay que escribir el programa, introducirlo en el microcontrolador y realizar las pruebas necesarias para garantizar que funciona correctamente. Este es un proceso que se realiza en varias iteraciones, modificando el programa según se detectan fallos o se modifica el comportamiento del sistema. Incluso puede afectar al diseño hardware, siendo necesario crear nuevos prototipos.

Para escribir el programa se pueden utilizar diferentes lenguajes, aunque el más utilizado actualmente es el lenguaje C. El lenguaje C es el mismo para cualquier tipo de microprocesador o microcontrolador, pero para que pueda ejecutarse necesita ser traducido al lenguaje propio del microcontrolador. Por ello los programas se desarrollan en un ordenador personal, muy cómodo para trabajar, utilizando una herramienta que se llama compilador cruzado (*cross compiler* en inglés). Un compilador cruzado es un traductor de lenguaje C (o de cualquier otro lenguaje) a lenguaje del microcontrolador. Tiene el apellido de cruzado porque el traductor se ejecuta sobre un ordenador, por ejemplo personal, pero genera un programa que se ejecuta en un ordenador distinto, en nuestro caso en el microcontrolador que vamos a utilizar.

Una vez escrito el programa y traducido al lenguaje del microcontrolador, hay que introducirlo, o descargarlo, en la memoria del microcontrolador para poder ejecutarlo. Esto se hace conectando el microcontrolador al ordenador personal. Una vez descargado el programa es posible depurar (*debug*) el programa ejecutándolo paso a paso e inspeccionar cómo cambian las variables, observando con detalle la evolución del pro-

grama y permitiendo detectar errores⁹. Una vez se dispone de la versión final del software y del hardware, puede comenzarse su producción en masa.



Fuente: Antonio Martí Campoy

Figura 10.9. Placa de experimentación MSP-EXP430FR5739 de Texas Instruments, con 8 leds, dos botones, sensor de temperatura y acelerómetro. Una de las principales ventajas es que la placa y el microcontrolador incorporan todo lo necesario para conectarla por USB a un ordenador personal y desarrollar una aplicación, con un precio inferior a los 40€. El microcontrolador es el chip situado en el centro de la placa, aunque hay un segundo microcontrolador para realizar la comunicación USB

El proceso descrito en el párrafo anterior parece sencillo, pero hace poco más de un par de décadas no lo era tanto. Eran necesarias varias herramientas software y hardware para descargar el programa en el microcontrolador y para realizar la depuración. El precio de estas herramientas podía alcanzar y sobrepasar los seis mil euros, lo que dejaba fuera del mundo de los microcontroladores a los aficionados a la electrónica y la informática. Las mejoras en la integración de circuitos han permitido incluir dentro del microcontrolador las herramientas hardware que antes se compraban por separado. Esto recibe el nombre de *on-chip debugger* y permite desarrollar aplicaciones para el microcontrolador utilizando adaptadores USB cuyo precio oscila entre los 10 y los 100 euros para microcontroladores de gama baja y media. De este modo, cualquier estudiante de informática o aficionado a la informática y la electrónica puede estudiar y desarrollar sistemas con microcontrolador con un presupuesto que empieza en los 30 euros.

⁹ La aparición de herramientas software y hardware que permiten depurar cualquier programa, ha hecho que el desarrollo de software sea más rápido, barato y fiable y menos tedioso.

DÓNDE ENCONTRAR UN MICROCONTROLADOR

¿Dónde puedo mirar y saber que hay un microcontrolador? Los puedo encontrar en cualquier industria que utilice máquinas y procesos productivos con un funcionamiento automático o semiautomático. Dentro de esas máquinas o en sus procesos productivos es seguro que hay un microcontrolador, o casi con toda seguridad, varios. Y hablamos de cualquier actividad industrial realizada por el hombre: alimentación, química, farmacéutica, agricultura, metalúrgica, etc.

Pero también podemos encontrarlos en objetos y máquinas más próximas a nuestra vida privada, como juguetes, mandos a distancia, lavadoras, hornos microondas, sistemas de climatización e iluminación, ascensores, máquinas expendedoras de bebidas y aperitivos, en las tarjetas inteligentes llamadas "de chip", y en muchos, muchos objetos y máquinas más. Como ejemplo podemos hablar del automóvil¹⁰, uno de los paradigmas del uso de microcontroladores. En un automóvil moderno podemos encontrar entre 50 y 100 microcontroladores que se encargan de la gestión del motor, de los sistemas de seguridad activos como el ABS, el airbag y el control de estabilidad, del panel de instrumentos, de la climatización o del control de la transmisión automática, entre otros sistemas del automóvil.

Y seguro que si pensamos un poco más encontraremos muchos más microcontroladores a nuestro alrededor, ya que las expectativas de venta para el año 2014 eran de unos 18.000 millones de microcontroladores en todo el mundo. Para comprender la importancia de esta cifra, la podemos comparar con el número de microprocesadores vendidos para construir ordenadores personales, servidores, videoconsolas, tabletas y teléfonos móviles, que todos juntos se estimaba que llegaría a la nada despreciable cifra de los 3.000 millones de unidades en el año 2014.

En fin, si usted pensaba que en el mundo hay muchos ordenadores, estaba en lo cierto, pero posiblemente son más, muchos más, de los que imaginaba. Y la mayoría de ellos no los ve, aunque los use.

¹⁰ En Europa hay casi un automóvil por cada dos habitantes, aunque esta cifra varía según el país.

BIBLIOGRAFÍA

AUGARTEN, S. (1983), *State of the art. A photographic history of the integrated circuit*. New Haven and New York: Ticknor & Fields.

BENNETT, S. (1986), *History of Control Engineering 1800-1935*, London, UK: Peter Peregrinus Ltd.

BENNETT, S. (1993), *A History of Control Engineering, 1930-1955*, London, UK: Peter Peregrinus Ltd.

PONT, M. J. (2002), *Embedded C*. London, UK: Addison-Wesley.

SIEWIOREK, D. P., GORDON, C., NEWELL, A. (1982), *Computer Structures: Principles and Example*. Caledonia: McGraw-Hill, Inc.

SPASOV, P. (1993), *Microcontroller technology: the 68HC11*. Englewood Cliffs: Regents/Prentice Hall.

Martín Mellado Arteche
Universitat Politècnica de València

Resumen: La complejidad de los sistemas robotizados es muy alta y ha crecido durante años. Los robots son cada vez más potentes, tanto física como “intelectualmente”, más flexibles y autónomos. Este desarrollo es elevado y rápido, y todavía no se vislumbra su final. Probablemente solo los más agoreros se atreven a describirlo, con perspectivas muy pesimistas para la humanidad. La robótica es un campo muy multidisciplinar. Si tradicionalmente ha sido cubierto por áreas ingenieriles como la mecánica, la automática, la electrónica y la informática, hoy en día se requiere más personal de otros campos que intervengan en el desarrollo de nuevos productos, como médicos, psicólogos, filósofos, etc. Pero cada vez más, los pilares básicos que sirven de sustento para los nuevos desarrollos de investigación en robótica caen en el área de la informática, en sus diferentes campos como teoría de la computación, arquitectura de computadores, ingeniería del software o inteligencia artificial. Todo ello con el objetivo de encontrar nuevos sistemas robóticos que se aproximen cada vez mejor al comportamiento humano. Este artículo se centra en resaltar cómo la evolución de la informática ha permitido el desarrollo de la robótica, que ha alcanzado niveles impensables hace muy pocos años.

ANTECEDENTES DE LA ROBÓTICA

La noción de robot proviene de dos conceptos genéricos, humanoide y autómatas. El humanoide es un concepto ancestral, que ya en leyendas de Egipto, Babilonia o Sumeria de hace 5000 años, reflejan la imagen de la creación, con dioses que dan vida a modelos de arcilla. Una variación es la idea del *gólem*, asociada a la Praga del siglo XVI, en la que un modelo de arcilla cobra vida y se vuelve destructivo. Un precursor del Frankenstein de Mary Shelley, que en su libro *The modern Prometheus* (1818) combina la noción de humanoide con los peligros de la ciencia.

El concepto de autómatas significa literalmente entidades dotadas de movimiento y ha tenido siempre un gran interés para el hombre. Los primeros modelos de autómatas dependían de palancas y manivelas, o bien estaban basados en la hidráulica. Después se aplicaron los avances en la ciencia de la relojería (siglo XIII) y posteriormente en las máquinas de vapor y la electromecánica. Los primeros autómatas tenían como propósito el entretenimiento, si bien las nuevas tecnologías del siglo XX los convirtieron en dispositivos útiles para la industria.

A partir de estos términos, surgió la palabra robot derivado del término checo *robota*, que significa trabajo forzado o servicio obligatorio, o bien del término *robotnik*, que significa siervo. Se utilizó en primer lugar por el dramaturgo Karel Čapek, en 1918, en un relato breve y luego en 1921, en la obra de teatro titulada R.U.R. (Rossum's Universal Robots). Rossum era un inglés de ficción que utilizaba métodos biológicos para inventar y producir en serie “hombres” que servían a los humanos. Estos seres se rebelaron y acabaron como raza dominante, sometiendo a la humanidad. Esta idea original hace que se asocie a los robots con seres de apariencia humanoide, imagen reforzada con la película alemana *Metrópolis* (1926), en la que aparecía un andador eléctrico. Las películas de ciencia ficción han potenciado esta idea (Figura 11.1).

La ciencia ficción ha mostrado robots de todo tipo, como por ejemplo cómicos (Johnny 5 en *Cortocircuito*), agresivos y destructivos (*Terminator*) o infantiles (David en *A.I.*). También aparecen con variadas funciones, como de traductores (C3-PO en *La guerra de las galaxias*), asistentes (Andrew Martin en *El hombre bicentenario*) o como exoesqueletos (usado por Sigourney Weaver en *Alien, el regreso*). Algunos han sido estáticos (HAL en *2001: una odisea del espacio*) mientras que otros cuentan con una alta movilidad (Eva en *Wall-E*). Pero ya antes de aparecer en el cine, en 1942, Isaac Asimov redactó las Leyes de la Robótica:

1. Un robot no puede dañar a un ser humano o, por inacción, permitir que un ser humano pueda resultar dañado.
2. Un robot debe obedecer las órdenes dadas por los seres humanos excepto cuando tales órdenes entren en conflicto con la Primera Ley.
3. Un robot debe proteger su propia existencia hasta donde esta protección no entre en conflicto con la Primera o Segunda Ley.

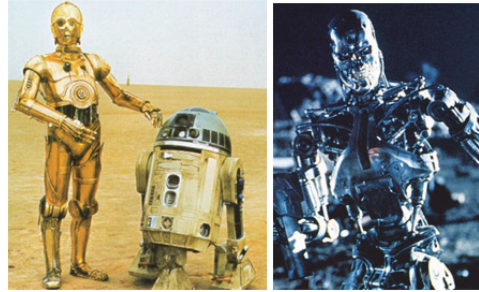
Estas leyes, formuladas dentro del entorno de la ciencia ficción, se han tomado como base y referencia histórica en la robótica y la inteligencia artificial, si bien hay abierta una larga controversia respecto a su implantación en nuestro mundo. De hecho, multitud de obras del propio Asimov se encargan de analizar casos conflictivos que pueden presentarse mediante la aplicación literal de dichas leyes. Tras un análisis de dichos conflictos, posteriormente se añadió una Ley 0 orientada a proteger a la humanidad, ajustándose las anteriores en consecuencia.

Resulta al menos curioso constatar la influencia que la ciencia ficción ha aportado al campo de la robótica. En 1982 Asimov cuenta que su compañero de estudios en la Facultad de Físicas de la Universidad de Columbia, J.F. Engelberger, presidente fundador de Unimation Inc. (primera industria dedicada a la fabricación de robots), le reconoció que se interesó por los robots hasta el punto de dedicarse a su construcción tras leer sus obras. El término Robótica (*Robotics*) lo acuñó Asimov en 1942 como la ciencia o arte relacionada con la inteligencia artificial (para razonar) y con la ingeniería mecánica (para realizar acciones físicas sugeridas por el razonamiento). La primera y más importante empresa de fabricación de robots (Compañía de Robots y Hombres Mecánicos de los Estados Unidos) y los primeros ingenieros que se dedicaron a la ro-

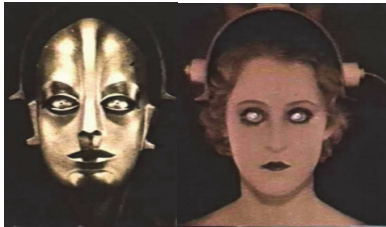
bótica, lo fueron en la ciencia ficción (Susan Calvin, Gregory Powell y Mike Donovan), si bien sobre robots positrónicos.



Helena y el Robot Radius (R.U.R.)



C3-PO y R2D2 y Terminator



Rotwang y Maria (Metrópolis)



Robocop y Wall-E

Figura 11.1. Robots en ciencia ficción (cortesía UFA, Lucas F., Warner Bros, MGM, Disney Pixar)

INICIO DE LA ROBÓTICA

Inmediatamente después de la Segunda Guerra Mundial comienzan los primeros trabajos que llevan a la aparición de los primeros robots industriales. A finales de los 40 se inician programas de investigación en los laboratorios Oak Ridge y Argonne National Laboratories para desarrollar manipuladores mecánicos para elementos radiactivos. Estos manipuladores eran del tipo “maestro- esclavo”, diseñados para que reprodujeran fielmente los movimientos de brazos y manos realizados por un operario. Efectivamente, fue alrededor de 1948 cuando surgió el progenitor más directo del robot, el *telemanipulador* (Figura 11.2), que fue desarrollado por R.C. Goertz del Argonne National Laboratory, para manipular elementos radiactivos evitando riesgos al operario. Era un dispositivo mecánico en el cual el operario accionaba los controles del manipulador desde una zona segura, mientras observaba a través de un cristal las evoluciones del mismo con los objetos radioactivos. Posteriormente se sustituiría el control mecánico por un control eléctrico.

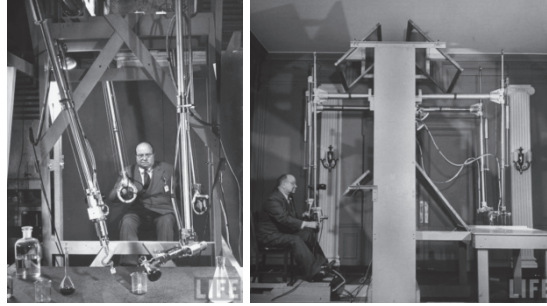


Figura 11.2. Manipuladores teleoperados de Goertz (cortesía de LIFE archive hosted by Google)

Más tarde Ralph Mosher de General Electric desarrolló un dispositivo consistente en dos brazos teleoperados controlados por un dispositivo maestro. El desarrollo de telemanipuladores fue seguido muy de cerca por la industria submarina, nuclear y en los años 60 por la industria espacial, pero el concepto de telemanipulador ha quedado muy reducido a estas industrias hoy en día. Pronto surgió la necesidad de sustituir el control del operario por un ordenador con un programa que controlase los movimientos del manipulador, dando así paso al concepto de robot.

Esta evolución vino fuertemente influenciada por el desarrollo del primer computador electrónico (1946), el desarrollo de los primeros dispositivos mecánicos con servomecanismos de posición y velocidad (1947), la introducción de la realimentación de fuerza (1948) y la construcción de la primera máquina herramienta con control numérico por el Instituto Tecnológico de Massachusetts (1952).

A mediados de los años 50, George C. Devol desarrolló un dispositivo de transferencia programada articulada (según su propia definición). Se trataba de un manipulador cuyas operaciones podían ser programadas y almacenadas. La primera patente robótica la solicitó en 1954 C.W. Kenward, pero el primero que estableció las bases del robot industrial moderno fue Devol con su dispositivo, que se patentó en Estados Unidos en 1961.

Los desarrollos posteriores de este concepto por el propio Devol y Joseph F. Engelberger condujeron al primer robot industrial, introducido en 1959 por Unimation Inc. (Universal Automation) la empresa pionera en el campo de la robótica fundada por Engelberger. La primera máquina UNIMATE se instaló en la fábrica de General Motors en 1960 para una aplicación de fundición por inyección. Este dispositivo utilizaba un computador junto con un manipulador dando lugar a una máquina que podía ser enseñada para la realización de tareas variadas de forma automática. Se puede considerar este punto como el inicio de la era de la robótica tal como la conocemos, mediante la utilización de los robots programados, una nueva y potente herramienta de fabricación.

LOS PRIMEROS ROBOTS

Durante la década de los 60, un nuevo concepto surge en relación con los anteriores avances. En vistas a tener una mayor flexibilidad, se hace necesaria la realimentación sensorial. H. A. Ernst publica en 1962 el desarrollo de una mano mecánica con sensores táctiles controlada por computador, llamada MH-1. Este modelo evolucionó adaptándole una cámara de televisión dentro del proyecto MAC. También en 1962, Tomovic y Boni desarrollan una mano con un sensor de presión para la detección de objetos que proporcionaba una señal de realimentación al motor. En 1963 se introduce el robot comercial VERSATRAN por la American Machine and Foundry Co. (AMF) y se desarrollan otros brazos manipuladores (Roehampton y Edinburgh).

En 1967 y 1968 se incorporaron varios robots UNIMATE serie 2000 en las cadenas de fabricación de automóviles de la General Motors. En 1967 General Electric Corporation construye el primer vehículo de 4 patas. En 1968 se publica el desarrollo de un sistema con manos, ojos y oídos (manipuladores, cámaras y micrófonos) por parte de McCarthy en el Stanford Artificial Intelligence Laboratory del Stanford Research Institute. Este sistema era capaz de reconocer mensajes hablados, detectar piezas distribuidas en una mesa y manipularlas de acuerdo con ciertas instrucciones. En el mismo año, Pieper estudia el problema cinemático de un manipulador controlado por un computador.

También en 1968, la compañía japonesa Kawasaki Heavy Industries negocia con Unimation la licencia de sus robots. Este momento marca el inicio de la investigación y difusión de los robots industriales en Japón. Más tarde Nissan formó la Asociación de Robótica Industrial de Japón en 1972 mientras que en 1974 se formó el Robot Institute of America que en 1984 pasó a denominarse Robot Industry Association (RIA).

En la década de los 70 la investigación en el campo de la robótica estaba en expansión y los desarrollos se fueron sucediendo. Durante toda la década se enfatizaron los esfuerzos no solo en desarrollos de nuevos y más potentes modelos de robots, sino que se resaltaron los trabajos en el uso de sensores externos para facilitar las operaciones de manipulación.

Las primeras aplicaciones industriales en Europa, aplicaciones de robots industriales en cadenas de fabricación de automóviles, datan de los años 1970 y 1971. En este último año, Kahn y Roth analizan el comportamiento dinámico y el control de un brazo manipulador. En 1973 ASEA (actual ABB), industria sueca construyó el primer robot con accionamiento eléctrico, el IRB 6 y más tarde el IRB 60. Entre 1973 y 1974, Cincinnati Milacron introduce el T3 (*The Tomorrow Tool*), su primer robot industrial controlado por computador. Este manipulador podía levantar cerca de 50 kg y seguir objetos móviles en una línea de montaje. Durante la década de los 70, la investigación en robótica se centra en gran parte en el uso de sensores externos para su utilización en tareas de manipulación. Así, en 1973, Bolles y Paul utilizan realimentación visual y de fuerza en el brazo Stanford para el montaje de bombas de agua de automóvil.

En 1974, Nevins y sus colaboradores, en el Draper Laboratory, investigaron técnicas de control basadas en la coordinación de fuerzas y posiciones, y Bejczy, en el Jet Propulsion Laboratory, desarrolló una técnica para el control de par basada en el robot Stanford. El mismo año, Inoue, en el Artificial Intelligence Laboratory del MIT, desarrolló trabajos de investigación en los que aplicaba la inteligencia artificial en la realimentación de fuerzas. En 1975, Will y Grossman, en IBM, desarrollaron un manipulador controlado por computadora con sensores de contacto y fuerza para montajes mecánicos. En 1976 los robots espaciales de la NASA Viking 1 y Viking 2 son utilizados para tomar imágenes en Marte. En 1977 Machine Intelligence Corporation desarrolla un sistema de visión comercial que se puede integrar en aplicaciones industriales robotizadas. A mediados de la década de los 70, la General Motors financió un programa de desarrollo en el que el investigador Víctor Scheinman del Instituto de Tecnología de Massachussets inventó un brazo mecánico para trabajos de producción. El brazo, comercializado por Unimation e instalado en fábricas de General Motors, se denominó manipulador universal programable para ensamblaje (*Programmable Universal Manipulator for Assembly*, PUMA). El PUMA marcó el inicio de la era de los robots industriales.

En 1980 se forma en Europa la IFR (*International Federation of Robotics*). Los primeros robots tenían configuraciones denominadas antropomórfica y esférica, de uso para la manipulación, pero es en 1982 cuando Makino de la Universidad de Yamamashi en Japón define el concepto de robot SCARA (*Selective Compliance Assembly Robot Arm*) que es un robot con 3 o 4 grados de libertad orientado al ensamblado de piezas. La industria de la robótica no se puede considerar como tal durante los años 60, aunque las primeras aplicaciones industriales datan de esa época. Únicamente se podían considerar algunas empresas pioneras con pedidos ocasionales cuyo volumen de ventas no alcanza el estatus de industria especializada.

Hasta la mitad de los años 70 no comienza a ser la robótica lo que puede considerarse como el inicio de una industria. Entre 1975 y 1977 se estima que las ventas de Unimation (prácticamente la única empresa existente) se multiplicaron por 2,5. A partir de ahí, otras empresas, bastante significativas (Cincinnati Milacron, ASEA, etc.), deciden entrar en el mercado de la robótica, comenzando también la industria del automóvil a realizar pedidos importantes. Las ventas pasan de 15 millones de dólares en 1976 a 45 millones de dólares en 1979, es decir, triplicándose en tres años. Otras industrias diferentes a las del automóvil comienzan a descubrir la robótica, produciéndose una espectacular expansión en tres fases temporales (Figura 11.3).

En resumen, puede decirse que durante la primera fase la robótica consistía en una limitada serie de usuarios y vendedores en el sentido clásico de la palabra, no realmente una industria. En la segunda fase las pocas empresas que vendían robots experimentaron un fuerte crecimiento. En la tercera fase la robótica deja de ser una novedad y un sueño para convertirse en una industria con bastantes empresas, muchos más usuarios y gran volumen de ventas.

Los datos recientes muestran que el mercado de la robótica, con una implantación de prácticamente 180.000 unidades en 2013, hasta superar el millón y medio de unidades operativas en la industria a nivel mundial, ha movido un total de 10.000 millones de dólares en el último año, cantidad que se triplica si se considera mercados que rodean a la robótica, como los periféricos, el desarrollo de software o la ingeniería de implantación. A nivel nacional existen más de 32.000 robots manipuladores industriales operativos en la industria española, habiendo sido más de 3.000 de ellos incorporados en 2013.

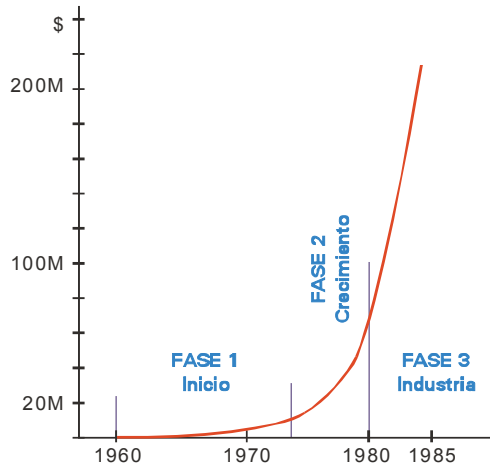


Figura 11.3. Fases en la evolución de la robótica

LA ECLOSIÓN DE LA ROBÓTICA

Los robots exhiben 3 elementos clave:

- **Capacidad mecánica**, que lo capacita para realizar acciones en su entorno y no ser un mero procesador de datos (el robot es una máquina).
- **Programabilidad**, lo que significa disponer de capacidades computacionales y de manipulación de símbolos (el robot es un computador).
- **Flexibilidad**, puesto que el robot puede operar según un amplio rango de programas y manipular material de formas distintas.

Por tanto, se puede considerar un robot como una máquina complementada con un computador o como un computador con dispositivos de entrada y salida sofisticados. Un robot también se puede ver como una combinación de inteligencia, capacidad de manipulación y un actuador final, que actúa como herramienta. Por ello, se asocia frecuentemente el robot a una reproducción de las cualidades antropomórficas de las

personas, que combinan inteligencia y capacidad de manipulación con la mano, o esta con una herramienta (destornillador, soldador, etc.).

La IFR define como robot industrial manipulador a un manipulador programable en tres o más ejes, controlado automáticamente, reprogramable y multifuncional, que puede estar fijo en un lugar o móvil para uso en aplicaciones automáticas de la industria (ISO 8373). La mayor parte de los robots manipuladores industriales (Figura 11.4) tienen solamente una semejanza lejana con robots antropomórficos. No suelen ser móviles y disponen únicamente de poco más que un brazo ligado a una base fija. Todos ellos tienen, sin embargo, una cierta forma de inteligencia debida a su control mediante computador. En un robot industrial de rango elevado, su computador de control es capaz de aportar un considerable grado de inteligencia (artificial), de forma que no solo guía al robot a realizar los movimientos previstos en su programa, sino que también lo integra con otros equipos y dispositivos subordinados a él y toma ciertas decisiones según información de sensores.

Se puede decir que hoy en día existen pocas variaciones en la componente mecánica de los robots industriales, presentándose cuatro o cinco configuraciones básicas que las diferentes marcas fabrican con pocas diferencias. Por el contrario, la evolución en los sistemas de control es rapidísima, tanto en hardware como en software, siendo quizá el aspecto que más resalta la diferencia entre diferentes marcas. Evidentemente, no hay que olvidarse que sin el sistema de control, el robot pasaría a ser un dispositivo muerto, una mera escultura.



Figura 11.4. Ejemplos de robots industriales (cortesía ABB, FANUC, KUKA y ADEPT)

El sistema de control es un ordenador industrial que incorpora numerosos elementos (tarjetas) que le permiten relacionarse con el entorno en que trabaja. Entre otras, este sistema se encarga de la interfaz de comunicación con el usuario, del control del movimiento del robot y de las comunicaciones con otros sistemas. En el sistema de control se programa el robot, es decir, se introducen instrucciones para su posterior ejecución.

Otro aspecto fundamental en los sistemas de control hoy en día es su posibilidad de comunicación de cara a la integración con otros sistemas de la planta. Además del uso de señales digitales y analógicas y las tradicionales comunicaciones serie y paralela, la

mayoría de sistemas ya permiten comunicación Ethernet, lo que permite una fácil integración en planta así como la comunicación con el fabricante para actualizaciones, mantenimiento y aviso de averías o problemas, etcétera.

Los robots industriales se componen de un sistema mecánico gobernado por un sistema de control al que se accede desde una unidad de programación. Si bien respecto al software de control de los robots industriales se dispone de poca información relativa a su implementación, no ocurre lo mismo con la forma de interactuar entre operario y sistema. Para esta función se dispone de la unidad de programación o *teach-pendant*, de aspecto muy variable (Figura 11.5), como siempre dependiendo de la marca concreta, pudiendo diferenciarse las siguientes características:

- Pantalla: De las típicas pantallas alfanuméricas de pocas líneas (incluso solo dos) se ha pasado recientemente a pantallas gráficas en color que permiten un interfaz más agradable.
- Teclado: suelen disponer de un teclado alfanumérico, una serie de teclas específicas, por ejemplo para el manejo de la herramienta, y unas teclas multifunción que se corresponden con opciones de los menús que van apareciendo en la pantalla.
- Dispositivo de guiado: puede ser mediante teclas o un dispositivo articulado tipo *joystick*. Este último puede ir desde dos o tres grados de libertad, el caso más común, a los más avanzados dispositivos de seis grados de libertad, si bien son menos intuitivos de manejar.
- Medidas de seguridad: el paro o seta de emergencia corta la alimentación a la vez que frena los motores del robot, ya que está pensado para casos de emergencia.

Para programar los robots se utilizan lenguajes de programación específicos, generalmente particulares para cada marca de fabricante. Han sido muchas y variadas las evoluciones de los lenguajes y metodologías de programar los robots (Figura 11.6). Algunos lenguajes han sido evolución o ampliación de lenguajes informáticos como el VAL (de BASIC) o el RAPID (de C). Otros fueron desarrollados *ex profeso* para programar robots, con un diseño muy específico para este campo, como el RAPT.



Figura 11.5. Unidades de programación (cortesía de KAWASKI, KUKA, FANUC y MOTOMAN)

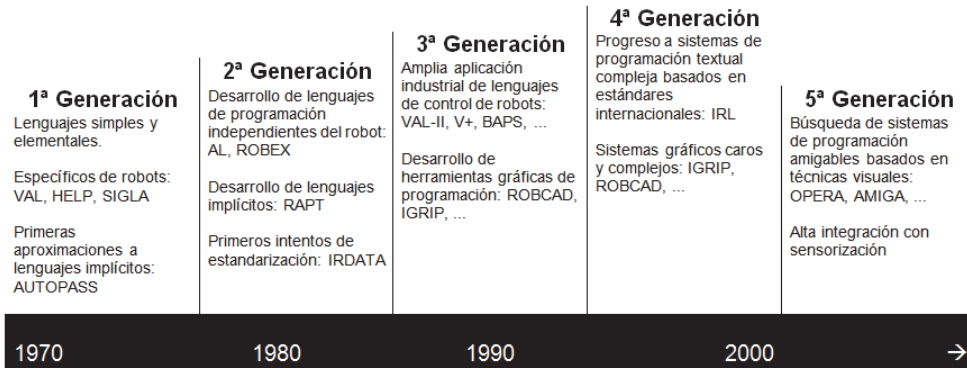


Figura 11.6. Fases en la evolución de los lenguajes y metodologías de programación de robots

LOS ROBOTS DE SERVICIO

Hasta ahora se ha estado contemplando una parte de la robótica, la robótica industrial, que ha sido la base del desarrollo de esta área tecnológica y campo fundamental en sus inicios. Sin embargo, en las últimas décadas ha surgido con gran fuerza otro tipo de robots no diseñados para su uso en la industria. Por ejemplo, han aparecido robots aspiradores, cortacéspedes o limpiafondos para piscina, muy útiles en los hogares de personas de cierto nivel adquisitivo, así como juguetes robóticos que hacen las delicias de los más pequeños y los no tanto. Este nuevo campo de aplicación se denomina robótica de servicio, distinguiéndose entre los robots de uso profesional y los robots para uso doméstico y personal. La IFR ha definido el robot de servicio como *un robot que funciona semi o totalmente autónomo para realizar servicios útiles para el bien de los seres humanos y equipos, excluyendo las operaciones de fabricación*. Sin embargo, la barrera que delimita los robots de servicio de los robots industriales manipuladores es muy difusa, siendo la aplicación a la que se destina el robot el término más distintivo.

Con unos simples datos se puede conocer la evolución de la robótica de servicio. En 2013 se vendieron 21.000 unidades de robots de servicio para uso profesional, presentando un mercado mundial de 3.600 millones de dólares americanos. El mismo año se vendieron 4 millones de robots de servicio para uso doméstico y personal, generalmente más económicos, con un mercado mundial de 1.700 millones de dólares americanos. Sin embargo, las previsiones del mercado para los próximos años son enormes. Por ejemplo, en Japón, donde se está presentando un gran crecimiento de la robótica asistencial doméstica para ancianos, se espera que en 10 años este sector emergente supere a la robótica industrial y que en 20 años el mercado se multiplique por 20. Uno de los factores que más potenciará este crecimiento es el hecho de que se espera que pronto existan robots para el hogar disponibles en alquiler, gestando un nuevo mercado asociado a la robótica de servicios. Si bien esta gran progresión es debido a muchas causas, como el desarrollo de nuevos materiales, la miniaturización de baterías energéticas

con mayor autonomía, o la reducción de costes en componentes, posiblemente sean los campos relacionados con la informática los que más influyan en este auge, considerando por ejemplo el desarrollo de sistemas empujados, las mejoras en sistemas de comunicación, la mayor integración de componentes electrónicos o los avances en aprendizaje y capacidad autónoma en la toma de decisiones.

La principal característica de los robots de servicio es que trabajan en entornos no estructurados y al menos parcialmente desconocidos. En ellos prima el servicio que ofrecen al acceder a zonas inaccesibles o peligrosas para el ser humano que su rentabilidad económica. En la mayoría de los casos se trataba en su origen de robots teleoperados, si bien la investigación ha avanzado enormemente en este campo y hoy en día contemplan cada vez mayor nivel de autonomía. El control aplicado sobre ellos está basado generalmente en comportamientos reactivos, de forma que ante un evento, reaccionan según su programación, sin perder una planificación previa global y contemplando nuevos aspectos de aprendizaje y capacidad de toma de decisiones.

Como se indica en su definición, los robots de servicio se usan en entornos no industriales, o más concretamente, excluyendo las operaciones de fabricación. Por tanto se utilizan en sectores como la medicina, agricultura, construcción, entretenimiento y un largo etcétera. Pero hay que separar los robots de producción, transporte y almacenaje que se usan en estos sectores de los robots móviles en hospitales, oficinas o bibliotecas que se pueden considerar como robots industriales destinados a la producción.

Entre los principales sectores en que se están introduciendo los robots de servicio (Figura 11.7) destacan: agricultura (robots rodantes tipo tractores para actividades de labranza, siembra, desinfección o recogida), construcción (robots articulados y teleoperados tipo grúas en camiones para la manipulación de material), entrenamiento (robots rodantes para el guiado y servicio de información en museos, juguetes), industria nuclear (robots rodantes para inspección y carga), medicina (brazos articulados para operaciones quirúrgicas y ayuda a personas discapacitadas), sector marítimo (exploración submarina, montaje de plataformas), servicios domésticos (robots rodantes como aspiradores para la limpieza de suelos y piscinas, atención de enfermos, y un largo etcétera.



Figura 11.7. Algunos ejemplos de robots de servicio para usos domésticos y profesionales (cortesía de iRobot, Totelius Domio, Da Vinci, Lokomat, ISE Ltd y la NASA)

ROBOTS HUMANOIDES Y COMPORTAMIENTOS HUMANOS

Los últimos avances en la evolución de la robótica se centran en los robots humanoideos. Hay que tener claro que se puede entender por humanoide al robot cuya estructura corporal se asemeja a la de un ser humano: cabeza, torso, piernas, brazos y manos. La evolución en su desarrollo ha sido muy rápida, alcanzando aspectos humanos en poco tiempo (Figura 11.8). En este sentido, los esfuerzos de investigación se centraron principalmente en dotar a estos robots del sentido del equilibrio y de la capacidad de andar de forma “natural”, entendiendo como tal el andar humano. Pero también se considera humanoide al robot diseñado para comportarse a semejanza del ser humano, siendo quizá este aspecto el más destacado. De esta forma es más importante dotar a un robot de una forma de razonar, deducir, inducir y discurrir, es decir, de pensar, que se acerque al de las personas, aunque no se disponga de piernas y utilice ruedas para su desplazamiento.

Uno de los aspectos que diferencian a los robots humanoideos de otros robots es que, mientras estos suelen estar diseñados para trabajar en escenarios específicos y para realizar tareas o aplicaciones determinadas, los humanoideos deben ser capaces de trabajar en ambientes humanos, directamente en su entorno (hogar, oficina, etc.) usando las mismas herramientas que los humanos (tijeras, bolígrafos, cubiertos) y sin modificaciones. Por tanto, la capacidad de adaptación a posibles cambios en el escenario es un reto que debe resolverse con avances en los sensores a utilizar y en la inteligencia dotada al sistema de control.

Para conseguir la máxima semejanza al comportamiento humano, se ha desarrollado la robótica cognitiva, que tiene por objetivo dotar a los robots de comportamientos inteli-

gentes, con una arquitectura de procesamiento de información con el que aprender y razonar sobre la forma de comportarse frente a problemas complejos en entornos complicados. Dentro del comportamiento también se incluye, como es obvio, la capacidad de movimiento del robot.

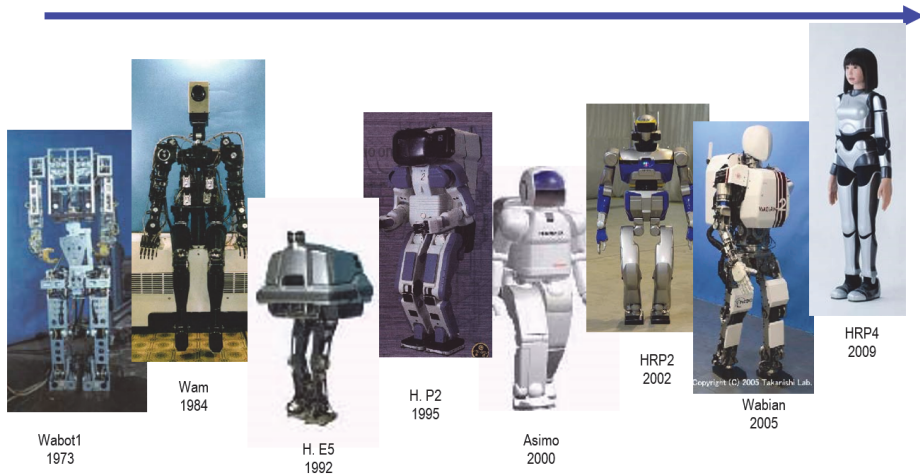


Figura 11.8. Evolución de los robots humanoides (cortesía de Waseda, Honda, aIST y Takanishi Lab)

CONCLUSIONES

El avance de la robótica es imparable e irá a más en un futuro cercano. En su desarrollo la informática ha sido y es pieza clave, tanto en lo que respecta al desarrollo del hardware y el campo de la ingeniería de computadores, en las capacidades de control que presta la automática, como aún más en los aspectos computacionales como la inteligencia artificial, el aprendizaje, la capacidad de decisión, etcétera. Sin los progresos alcanzados en estos campos, sería imposible contar con los robots que tenemos hoy en día o con las perspectivas que presenta la robótica a la sociedad para mejorar la calidad de vida.

BIBLIOGRAFÍA

ASIMOV, I. (1982), *The Complete Robot (Los Robots en versión española)*, Garden City, New York: Doubleday & Company Inc.

BARRIENTOS, A.; PEÑÍN, L. F.; BALAGUER, C.; ARACIL, R. (1997), *Fundamentos de robótica*, Madrid: McGraw-Hill.

CLARKE, R. (1993), “Asimov’s Laws of Robotics: Implications for Information Technology (1 & 2)”, *Computer Magazine*, IEEE Computer Society.

FU, K. S.; GONZÁLEZ, R. C.; LEE, C. S. G. (1998), *Robótica. Control, detección, visión e inteligencia*, Madrid: McGraw-Hill.

MELLADO, M. (2009), *Robótica*, Valencia: Editorial UPV.

OLLERO, A. (2001), *Robótica. Manipuladores y robots móviles*, Barcelona: Marcombo Boixareu Editores.

Miguel Sánchez López
Universitat Politècnica de València

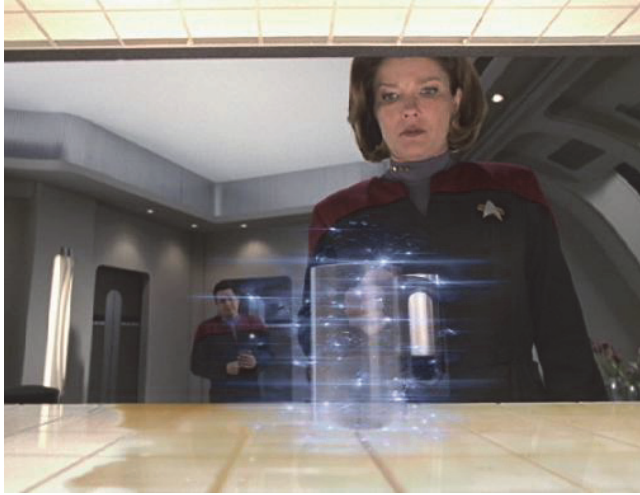
Resumen: La impresión 3D está acaparando la atención de los medios últimamente, con aplicaciones en múltiples campos desde aplicaciones industriales a aplicaciones médicas, pasando por su uso en el hogar y la oficina. En este artículo se repasan las diferentes tecnologías de impresión 3D, sus fundamentos físicos y las personas que las idearon. Asimismo, se presentan algunos de los servicios *on-line* que permiten al público en general acceder a diseños abiertos de piezas y a servicios de fabricación “en la nube”.

INTRODUCCIÓN

En la literatura de ciencia ficción se nos han presentado innovaciones que aunque no existían al escribir sobre ellas se han hecho realidad tiempo después. Bien es cierto que otras muchas han evolucionado de forma totalmente diferente a como se predecía en las novelas y así hemos llegado a la Luna en un cohete y no en una bala de cañón como anticipaba el escritor Julio Verne.

Quizá el precursor de la impresión 3D podría basarse en la misma idea que el teletransporte, que permite enviar objetos de un lugar a otro como si las partículas que los forman se pudieran controlar a voluntad y eliminarlas de su ubicación actual para hacerlas aparecer en el destino. Películas como *La mosca* de David Cronenberg en 1986 exploran esta idea así como las posibles consecuencias de pequeños fallos en el proceso.

Sin embargo, la impresión 3D se separa del concepto del teletransporte por cuanto que no se parte de un objeto original como se hacía con el “replicador” empleado en la serie *Star Trek* (Figura 12.1) sino que se da un paso más y se propone generar un objeto físico tan solo a partir de su diseño como un modelo teórico.



Fuente: memory-alpha.wikia.com

Figura 12.1. El replicador en un fotograma de la serie Star Trek

Es lo que Neil Gershenfeld vino a denominar en el MIT como la transformación de los bits en átomos, un proceso por el que se pasa de la información que representa a un objeto, su diseño, a una versión tangible del mismo, el objeto físico. Esta es la operación que normalmente hemos denominado fabricación y que se viene empleando por multitud de diferentes tecnologías, pero quizá el punto de inflexión en el paso desde la artesanía a la producción industrial lo podemos asociar a la aparición de la imprenta. La imprenta ha permitido acelerar lo que hasta entonces era un trabajo artesano a la hora de reproducir un grabado en una hoja de papel.

Muchas máquinas automáticas, incluidos los robots, nos han permitido automatizar muchos procesos de fabricación. La impresión 3D aparece como un nuevo procedimiento para poder colocar materia en aquellos lugares del espacio en donde es necesaria para poder crear un objeto con la geometría deseada. Es este carácter de añadir material donde antes no lo había lo que le ha venido a dar el nombre de **fabricación aditiva** por oposición a otros procesos existentes como el fresado o el taladrado que consiguen conformar la geometría deseada a base de retirar material de un bloque sólido hasta conseguir, también, la geometría deseada. Este segundo proceso se conoce como **fabricación sustractiva**.

La cuestión es cómo pasar de la idea de la impresión 3D, que podemos asociar al control de la materia y su colocación en los lugares deseados, a una tecnología práctica que haga eso exactamente y a unos costes que puedan ser competitivos con otros procesos de fabricación. Pero toda esta tecnología no habría sido posible sin la disponibilidad de ordenadores y programas capaces de abordar la tarea que supone orquestrar el movimiento de los distintos ejes y actuadores para conseguir crear un objeto a partir de su

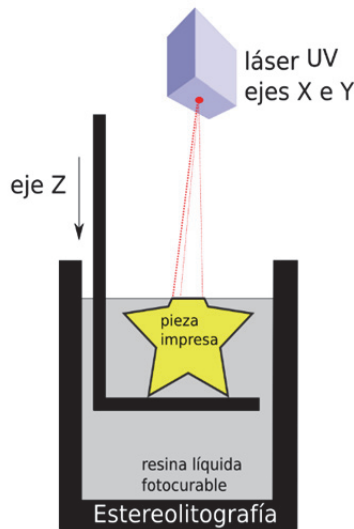
modelo tridimensional. La disponibilidad de programas para diseño CAD en tres dimensiones junto a dispositivos de escaneo 3D permitieron que se pudiera abordar una solución que, controlada por computador, actuaría como el cerebro de estos nuevos ingenios que denominamos impresoras 3D.

No debemos olvidar el espectacular desarrollo de la tecnología informática durante las décadas de los ochenta y noventa, pues resulta esencial para que la innovación de la impresión 3D haya sido posible. Sin suficiente memoria y capacidad de cálculo, disponer de un modelo 3D y su tratamiento posterior habrían resultado tareas inaccesibles.

INVENTOS, INVENTORES Y SUS PATENTES

La impresión 3D como tecnología surge a partir de invenciones que proponen una respuesta a la cuestión de cómo depositar el material en los lugares necesarios para poder obtener un objeto de la geometría deseada.

La primera de estas invenciones surge asociada al empleo de resinas fotosensibles. Su inventor, Chuck Hull, se da cuenta de que, puesto que la resina (empleada para endurecer la superficie de tableros de madera) se endurece por la acción de la luz, sería posible proyectar un patrón de luz para conseguir una impresión del mismo en forma de una fina capa de resina. Chuck Hull le dio nombre a este proceso: **estereolitografía** y lo patentó en 1986. Su proceso patentado emplea una resina fotosensible sobre la que se van proyectando patrones de luz, de forma que se va formando, capa por capa, la geometría deseada al endurecer selectivamente una resina líquida fotosensible (Figura 12.2).

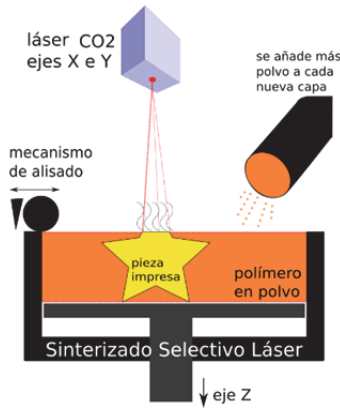


Fuente: elaboración propia

Figura 12.2. El proceso de estereolitografía

En la estereolitografía se emplea un polímero líquido que se endurece por la acción de un haz de láser ultravioleta cuya posición se va guiando para que realice el trazado del patrón deseado para formar cada capa. Una vez obtenidas nuevas patentes Chuck Hull funda la empresa 3D Systems en Valencia, California, empresa que a día de hoy es una de las dos empresas líderes en el sector en Estados Unidos.

La segunda invención relevante también se patenta en 1986, en este caso por el estudiante de la Universidad de Texas en Austin, Carl Deckard. La idea, denominada **sinterizado selectivo por láser**, consiste en agrupar partículas de polvo de material por la acción calorífica de un láser (Figura 12.3). El sinterizado es un proceso por el que dos partículas se unen sin tener que fundirse primero y solidificarse seguidamente, de forma similar a como dos hielos se pueden pegar entre sí. Carl Deckard licencia la tecnología de la Universidad de Texas y funda la empresa Desk Top Manufacturing que comercializa la invención y que años más tarde será adquirida por 3D Systems.



Fuente: elaboración propia

Figura 12.3. El proceso de sinterizado selectivo por láser

El tercer inventor es S. Scott Crump que en 1989 idea un modo de depositar finos filamentos fundidos de un termoplástico para formar, capa por capa, un objeto de plástico de la geometría deseada. El método se denomina **depósito de filamento fundido** (siglas FDM en Inglés) y se patenta [7] en 1992. El Sr. Crump y su esposa fundan la empresa Stratasys para comercializar su invención. Stratasys es estos momentos una de las dos mayores empresas de impresión 3D.

El proceso FDM no emplea láser sino que una resistencia que calienta una pequeña cámara en la que se funde el plástico momentos antes de ser depositado en el lugar apropiado de la pieza (Figura 12.4). Esta tecnología no permite, generalmente, conseguir piezas con un acabado comparable a los dos métodos anteriores debido al mayor espesor de las capas de material, pero resulta suficiente para muchas aplicaciones en las

que esta característica no es en absoluto un problema. Además, las piezas obtenidas son más resistentes que las de estereolitografía.



Fuente: elaboración propia

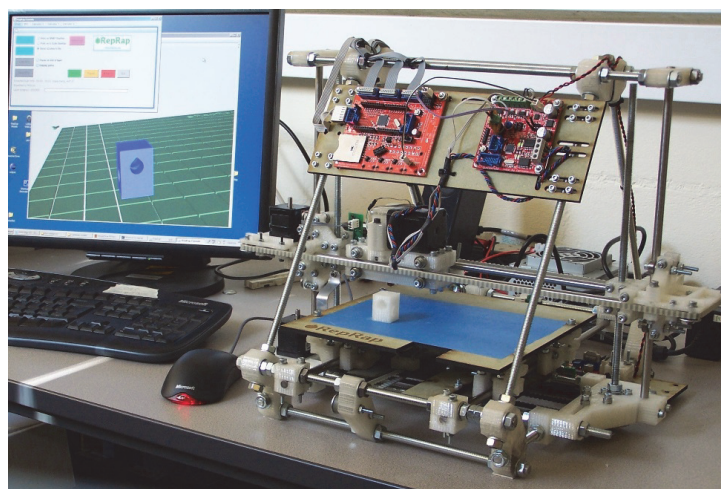
Figura 12.4. El proceso de depósito de filamento fundido

EL PROYECTO REPRAP

Si reflexionamos sobre la cobertura en los medios acerca de la impresión 3D en los años noventa o principios del siglo XXI, nos daremos cuenta de que, aunque no se trataba de una tecnología secreta, su presencia fue más bien escasa. El registro de titulares que emplean el término “3D printing” en Google Trends permanece plano y casi a cero hasta diciembre de 2010. A partir de ese momento el término empieza a crecer de manera importante.

Puesto que desde los años noventa existen empresas que comercializan máquinas de prototipado rápido (es así como también han llamado a las impresoras 3D) cabe preguntarse por qué hemos tardado tanto tiempo en ser conscientes de esta tecnología: uno de los principales argumentos ha sido el precio. Al tratarse de tecnologías patentadas la ausencia de competencia ha estado garantizada por ley.

En el año 2005, el profesor Adrian Bowyer de la Universidad de Bath, en el Reino Unido, inicia un proyecto cuyo objetivo es la creación de máquinas “autoreplicables”. Es el comienzo del proyecto RepRap (Figura 12.5).



Fuente: reppap.org

Figura 12.5. Impresora 3D del proyecto RepRap con un objeto recién impreso cuyo modelo se muestra en la pantalla del ordenador

Para que una máquina sea autoreplicable es necesario que sea capaz de producir piezas para poder construir otra máquina similar. Si bien es cierto que el objetivo de este proyecto aún está lejos de haberse conseguido por entero, el proyecto RepRap ha servido para el desarrollo de una familia de impresoras 3D basadas en el empleo de piezas creadas por otras impresoras 3D similares. Hemos asistido a lo que podríamos llamar un caso de evolución artificial, en el que modelos más antiguos de máquinas RepRap han impreso piezas para crear nuevos modelos de impresoras 3D, de tal suerte que la descendencia de unas máquinas ha superado en prestaciones a la generación que se usó para producirlas.

El proyecto RepRap se ha documentado en la web (<http://reppap.org>) y allí es posible encontrar un foro con enorme actividad en múltiples facetas de la impresión 3D que, a diferencia de otras iniciativas, es totalmente abierto y público. Es precisamente este proyecto el que, en mi opinión, ha propiciado el desarrollo y el conocimiento en esta área del saber en los últimos años, junto a los miles de participantes en el mismo.

En la actualidad, uno de los países donde este proyecto ha tenido mayor arraigo es en España, donde un grupo de entusiastas de esta tecnología liderado por el entonces profesor de la Universidad Carlos III, Juan González, vino a bautizar como Clone Wars. Cientos de máquinas se han construido basándose en diseños abiertos preexistentes sobre la base de que un nuevo miembro puede obtener el conjunto de piezas de plástico impresas en 3D por otro miembro del grupo y que luego el primero se compromete a hacer lo mismo por otros una vez tenga su impresora 3D funcionando.

Nos encontramos por tanto con que el proyecto RepRap, al distribuir la información sobre los modelos de máquinas desarrollados de forma abierta y sobre la base de un competencia abierta de ideas, ha propiciado la difusión y el avance de esta tecnología creando un movimiento que se asemeja, en mi opinión, al desarrollo de los primeros kits de microprocesadores a finales de la década de los setenta.

La empresa Makerbot (EE.UU.) fue, durante un tiempo, el exponente de un nuevo modelo en el que lejos de proteger los diseños de las máquinas que comercializaba, estos se hacían públicos y como resultado, muchas mejoras se debatían entre los usuarios, muchos de ellos también clientes. Sin embargo, con el tiempo la empresa fue creciendo, alguno de los fundadores la abandonó por desacuerdos con el resto y finalmente fue adquirida por la compañía Stratasys. En la actualidad ha abandonado por completo su política de diseños abiertos.

Otra empresa que nace al abrigo del proyecto RepRap y de forma similar a Makerbot se dedica a comercializar impresoras 3D es Ultimaker (Holanda), que ha venido siendo reconocida por modelos de impresoras que proporcionan excelente calidad y emplean relativamente poco espacio.

Pero quizá la figura más sorprendente de este proyecto es el joven estudiante y emprendedor checo Josef Jo Průša, que a los diecisiete años desarrolla una variación de un modelo existente de impresora 3D denominada Mendel y que denomina Prusa Mendel que se populariza rápidamente por la simplicidad de su montaje y su buen funcionamiento. Unos años después y tras mejorar su modelo, desarrolla una nueva impresora denominada Prusa i3 que de nuevo se ha convertido en estándar para muchos de los aficionados que buscan montarse su propia impresora 3D a base de piezas impresas por otra impresora 3D (además de una buena colección de otros componentes comprados).

LA IMPRESIÓN 3D EN EUROPA

La empresa EOS GmbH Electro Optical Systems fundada por los doctores Hans J. Langer (Figura 12.6) y Hans Steinbichler en 1989 se convierte en un referente para la industria europea al conseguir a BMW como su primer cliente para sus impresoras de estereolitografía.

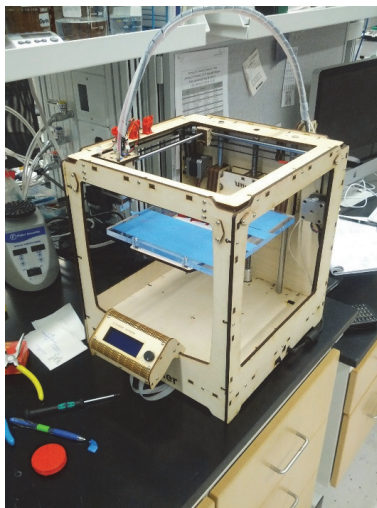


Fuente: www.eos.info

Figura 12.6. El doctor Hans J. Langer

Al año siguiente Hans Steinbichler vende sus acciones y abandona la compañía. En 1995 la empresa EOS incorpora también impresoras de sinterizado láser de polímeros en polvo, cubriendo así las dos tecnologías de prototipado rápido más populares hasta el momento. En 1996 incorporan el sinterizado de polvo metálico.

En el frente comercial de impresoras dentro del proyecto RepRap se encuentra la empresa holandesa Ultimaker, fundada en 2011 por Martijn Elserman, Erik de Bruijn y Siert Wijnia. Al igual que EOS, se trata de una empresa que vende impresoras 3D (en este caso con tecnología FDM) pero cuyos diseños son abiertos, esto es, podemos descargarnos los planos y esquemas de las impresoras que Ultimaker comercializa (Figura 12.7). Este es el denominado hardware abierto (*open source hardware*).



Fuente: Wikipedia

Figura 12.7. Una de las Impresoras 3D comercializada por Ultimaker

LAS IMPRESORAS 3D

Prácticamente la totalidad de impresoras desarrolladas en el proyecto RepRap se han agrupado en torno a la tecnología FDM, aunque dado que la empresa Stratasys registró las siglas FDM como marca el proyecto RepRap se vio obligado a acuñar un nuevo término para denominar a esa misma tecnología si no se quería ver expuesto a posibles conflictos. Ese nuevo término fue FFF (o fabricación de filamento fundido). En esta tecnología se dispone de un mecanismo de posicionamiento en el espacio de una boquilla que proporciona el flujo de plástico fundido a depositar capa a capa hasta construir el objeto deseado. Muchos de los mecanismos de posicionamiento son robots cartesianos de tres ejes, pero existen infinidad de otros diseños como robots SCARA o delta.

De forma análoga a como sucedió a principios de los años ochenta, multitud de pequeñas empresas han aparecido en este sector comercializando diseños propios o bien diseños y variaciones de máquinas del proyecto RepRap. Este último extremo es posible dado el carácter abierto de las licencias de los diseños de las máquinas RepRap.

Por otro lado, la oferta de todo tipo de impresoras 3D comerciales por las grandes marcas establecidas, fundamentalmente Stratasys y 3D Systems, comienza en sistemas de precios asequibles para un particular pero se expande hacia sistemas de grandes prestaciones, tanto mecánicas como técnicas y precios ciertamente solo al alcance de grandes compañías. También se han ido desarrollando sistemas más específicos para ciertos materiales, empleando otras fuentes de calor o luz diferentes del láser que han dado lugar a una gran diversidad de tecnologías de impresión 3D que ya no se limitan a la creación de piezas de plástico o de resina sino que incluyen metales, materiales cerámicos y materiales elásticos, siendo incluso posible la combinación de múltiples materiales de diferentes características en una misma pieza. Por ejemplo, es posible imprimir en una misma máquina una pieza que contenga varios plásticos de diferentes colores, rigidez y elasticidad para crear un prototipo con zonas que absorban impactos.

Aunque tradicionalmente los procesos de impresión 3D se han centrado en la reproducción del volumen de una pieza, el aspecto exterior de la misma no ha sido una prioridad. Así, muchas de las piezas tienen un color uniforme, ya sean opacas o translúcidas, que corresponde al material empleado. Sin embargo, algunos fabricantes reconocieron el interés de poder recrear no solo el volumen de un objeto sino su apariencia final, que incluye la coloración y textura del exterior del mismo. Así podemos encontrar diferentes tecnologías que incluyen un sistema que aporta color a la superficie exterior del objeto mientras éste se va imprimiendo, con lo que es posible crear por ejemplo un prototipo de un teléfono que no solo tiene la forma adecuada sino que además los botones y la carcasa tienen el color deseado, las teclas tienen números o letras, la pantalla muestra un texto y se puede ver el logo de la marca en el frontal.

APLICACIONES DE LA IMPRESIÓN 3D

La lista puede ser inmensa, pero esta tecnología de fabricación aditiva se ha venido empleando para el prototipado rápido de todo tipo de piezas para su aplicación en multitud de sectores. En el ámbito de la mecánica se producen prototipos físicos y en ocasiones funcionales de muchos de los componentes de coches, aviones y cohetes. En su aplicación a la medicina nos encontramos con prótesis óseas, articulaciones, reconstrucción facial, piezas dentales, y más recientemente la denominada bioimpresión que apunta a la creación de órganos funcionales para emplear en trasplantes. Y aunque pueda sonar a ciencia ficción, empresas como Organovo ya están trabajando para poder comercializar estos sistemas de bioimpresión.

La universidad de Loughborough en colaboración con el estudio de arquitectos Foster & Partners estudiaron la posibilidad de emplear la impresión 3D para la creación de bloques funcionales con los que construir edificios. Más recientemente, en 2014, empresas como la china WinSun han convertido esa idea en realidad comercial. Pero incluso particulares han aportado su granito de arena, nunca mejor dicho, y construido un castillo impreso en 3D en su propio jardín (Figura 12.8). Finalmente, cabe decir que también la robótica ha encontrado en la impresión 3D un valioso aliado a la hora de crear todo tipo de mecanismos para su aplicación a robots y líneas de producción.



Fuente: all3dp.com

Figura 12.8. Castillo de hormigón impreso en 3D por Andy Rudenko

LOS SERVICIOS ONLINE

¿Cómo de útil es una impresora convencional si no disponemos de un ordenador? ¿Y si disponemos de un ordenador pero no de un programa de tratamiento de textos? Las mismas preguntas se pueden formular al respecto de la impresión 3D. Muchos usuarios pueden estar interesados en reproducir determinadas piezas, por ejemplo las de un juego de

ajedrez, pero para que ello sea posible es necesario que las piezas se puedan encontrar en alguna parte. Algo parecido le ocurre a una persona que quiere imprimir en papel el formulario para hacer la declaración de la renta: solo es posible si alguien ha preparado previamente ese documento y lo ha puesto a disposición de quien lo quiera usar.

Mientras que todos esperamos que los ingenieros que diseñan un coche, una batidora o un teléfono sean capaces de utilizar un programa de diseño asistido por ordenador (CAD) lo mismo no es cierto para la gran mayoría. Por lo tanto, aunque la impresión 3D parece una tecnología que también puede interesar a casi cualquier usuario de un ordenador no es probable que muchos usuarios le saquen partido a menos que se puedan encontrar piezas en Internet de forma parecida a la que podemos encontrar información sobre, por ejemplo, Cervantes haciendo una simple búsqueda en Google. De otro modo, el uso de las impresoras 3D sería algo exclusivo de ingenieros y expertos en diseño mecánico por ordenador.

Afortunadamente, existen en la actualidad diversos repositorios en Internet que sirven como almacenes de diseños de piezas, en los que, por ejemplo, podemos encontrar esas piezas de ajedrez para imprimir en 3D o esa pequeña pieza de plástico que se nos ha roto del mando a distancia. Con este tipo de servicio, el usuario no avanzado puede obtener el diseño de la pieza que necesita sin tener que diseñarla o solo con pequeños retoques o ajustes.

Una de las principales aportaciones de la empresa Makerbot ha sido su visión de un repositorio público de diseños 3D. Para ello desarrollaron el sitio web thingiverse.com como un almacén para que los usuarios (clientes o no) puedan depositar sus diseños, especificando bajo qué tipo de licencia los publican. Este sitio se hizo muy popular entre la comunidad de usuarios de impresoras 3D pero un primer cambio en las condiciones de uso, que pasa a permitir a la empresa Makerbot a poder comercializar los modelos allí depositados (sin compensar a los autores) y posteriormente un incidente en el que la empresa Makerbot intentó patentar un diseño sospechosamente similar a un diseño libre que el usuario Whosa whasis había publicado en el sitio llevaron a muchos usuarios a abandonar en masa este repositorio. Una compañía rival, Ultimaker, pero que mantiene la política de hacer públicos sus diseños, ha puesto en marcha el repositorio youmagine.com que promete hacer lo correcto y no apropiarse de los derechos de los diseñadores.

La segunda opción para pasar del diseño al objeto físico. Aunque en principio requiera el uso de una impresora 3D no significa que todo el mundo necesite comprar una. De nuevo podemos encontrar diversos servicios en Internet que nos hacen las veces de imprenta 3D bajo demanda. Del mismo modo que muchas empresas encargan las tarjetas de visita a una imprenta, muchas personas pueden encargarse la impresión de piezas 3D a quienes ofrezcan ese servicio sin necesidad de tener que comprarse ellos una impresora 3D. Esta forma de proceder pone al alcance de todos nosotros la impresión 3D sin necesidad de tener que realizar inversión alguna.

Empresas como shapeways.com o sculpteo.com ofrecen un servicio de impresión bajo demanda, donde a partir de nuestro diseño, factor de escala y material seleccionado nos indican el precio del servicio de impresión 3D y, previo pago, se encargan de fabricarnos la pieza deseada y enviarla a nuestro domicilio. Es el modo de imprimir 3D sin tener que comprar los equipos, similar a lo que se hacía con el revelado de fotografías cuando no se disponía del equipamiento para revelarlas en casa. Aunque muchos aficionados revelaban fotografías en blanco y negro en sus casas, prácticamente la totalidad de las copias en color eran realizadas por laboratorios profesionales.

Con el tiempo, aquellos usuarios que aprecien la impresión 3D como una herramienta esencial en su arsenal de herramientas se pueden plantear adquirir o construir una para su propio uso. Algunos ya lo hemos hecho y la experiencia de idear una pieza, diseñarla, después imprimirla y finalmente colocarla haciendo la función para que fue diseñada, todo ello en la misma tarde, es francamente espectacular.

EL FUTURO DE LA TECNOLOGÍA DE LA IMPRESIÓN 3D

El principal aspecto que aporta la impresión 3D a la industria y en especial a la fabricación industrial, es la capacidad de personalizar. Si un juguete de plástico se hace con un molde de inyección, las millones de piezas realizadas serán todas iguales. Si un objeto se hace mediante impresión 3D sus dimensiones y forma se pueden adaptar para que se ajusten, por ejemplo, a la anatomía de la persona que tiene que usarlo.

Esta capacidad ha hecho que una de las áreas en las que la impresión 3D esté aportando importantes avances es en la creación de prótesis a medida en multitud de procedimientos médicos y veterinarios. Dada la existencia de procesos de impresión 3D en materiales como el titanio, la creación de prótesis a medida para cirugía traumatólogica y maxilofacial está desarrollándose a marchas forzadas.

De nuevo en España tenemos cierto liderazgo, en la figura del Instituto de Tecnología de Canarias, donde el ingeniero Donato Monopoli y su equipo llevan varios años colaborando en esta área con profesionales de la salud, creando prótesis de titanio sinterizado a medida para pacientes de trauma.

También en nuestro país se puede adquirir, desde hace ya varios años, una reproducción a escala de nosotros mismos en 3D a todo color. La misma idea de las fotografías pero ahora en tres dimensiones aunque ignoro si esta aplicación tendrá éxito comercial o no.

BIBLIOGRAFÍA

BARNATT, C. (2014), *3D printing*, CreateSpace Independent Publishing Platform, segunda edición.

VERNE, J. (1987), *De la Tierra a la Luna*, Editorial Porrúa.

GERSHENFELD, N. (2005), *Fab: The Coming Revolution On Your Desktop – from Personal Computers To Personal Fabrication*, Basic Books.

PATENTES

Apparatus for Production of Three-Dimensional Objects by Stereolithography (U.S. 4,575,330)

Selective Laser Sintering with Assisted Powder Handling (U.S. 4,938,816)

Method and Apparatus for Producing Parts by Selective Sintering (U.S. 4,863,538)

Multiple Material Systems for Selective Beam Sintering (U.S. 4,944,817)

Apparatus and Method for Creating Three-Dimensional Objects (A system and a method for building three-dimensional objects in a layer-by-layer manner via fused deposition modeling) (U.S. 5,121,329)

José Duato Marín
Universitat Politècnica de València

INTRODUCCIÓN

Es muy difícil predecir lo que va a ocurrir en un campo de la ingeniería como es el de la informática, en el cual la tecnología avanza tan rápidamente, y en el que además la evolución viene casi siempre determinada por el éxito o fracaso de los productos en el mercado.

Una observación que nos puede ayudar en esta tarea es que cuando ha aparecido una tecnología superior a otras existentes, si la inversión económica que había que realizar para hacerla competitiva en prestaciones era tal que no podía amortizarse con las ventas previstas para un intervalo de tiempo pequeño, dicha tecnología ha sido finalmente abandonada. Tal fue el caso de las memorias de burbuja magnética, que tuvieron el potencial de reemplazar a los discos duros por unos componentes sin piezas móviles y con tiempos de acceso más cortos. Pero los fabricantes de discos duros, mucho más poderosos económicamente, se pusieron a trabajar para aumentar su capacidad a un ritmo tal que las memorias de burbuja magnética nunca alcanzaron la capacidad de los discos duros y desaparecieron.

Sin embargo, el caso de los discos de estado sólido (SSD, *Solid State Drive*) es diferente, porque las memorias flash que constituyen la base de su tecnología han tenido un mercado mucho más grande que el de los propios discos duros: las cámaras fotográficas digitales, los móviles, las tabletas, y los dispositivos de almacenamiento de bolsillo. A pesar de ello, y contrariamente a lo que se ha predicho en numerosas ocasiones, los SSD no han reemplazado totalmente a los discos duros todavía, y es de prever que dicho reemplazo completo no ocurra a lo largo de la próxima década. El motivo es que la gran demanda inicial de memorias flash con capacidades crecientes se ha frenado. Muchos usuarios de portátiles se conforman con una memoria de bolsillo (el lápiz USB) de entre 8 y 32 GBytes, y ya no van a comprar otra de capacidad superior. Algo similar ocurre con las memorias para cámaras digitales y móviles. Por tanto, si la demanda de memorias flash de mayor capacidad se reduce casi exclusivamente al mercado de los SSD, entonces ya están en igualdad de condiciones competitivas con los discos duros en cuanto a mercado, pero van claramente por detrás en cuanto a capacidad de almacenamiento. Aun así, acaparan el mercado de los dispositivos en los que el peso es esencial (por ejemplo, los portátiles) o en los que la velocidad de acceso marca una gran diferencia en prestaciones globales.

EL PAPEL DE LA TECNOLOGÍA

Si queremos predecir cómo van a evolucionar los computadores, basándonos en la observación anterior, deberemos considerar la tecnología utilizada en los computadores y el nivel de inversión que se ha realizado en la misma. Actualmente se utiliza la tecnología CMOS (*Complementary Metal-Oxide Semiconductor*), que fue la tecnología que sobrevivió hace ya varias décadas a una dura batalla entre tecnologías alternativas tales como TTL (*Transistor-Transistor Logic*), ECL (*Emitter Coupled Logic*), PMOS (*Positive-channel Metal-Oxide Semiconductor*) y NMOS (*Negative-channel Metal-Oxide Semiconductor*). Curiosamente, CMOS era mucho más lenta que otras tecnologías como TTL y ECL, pero el hecho de ser la tecnología con menor consumo de energía eléctrica la hizo ser la preferida por los fabricantes, que invirtieron cantidades astronómicas de dinero de forma continuada para mejorar, no solo dicha tecnología sino también los procesos de fabricación asociados, así como para construir nuevas plantas de fabricación más avanzadas cada pocos meses. Y resulta curioso porque a pesar de dicha elección, el consumo de energía eléctrica y la correspondiente disipación de calor es lo que más está limitando actualmente el incremento de potencia de los ordenadores.

Se está investigando en tecnologías alternativas a CMOS, la mayoría de las cuales no están basadas en el silicio. Las tecnologías que más difusión han conseguido son los denominados ordenadores cuánticos, que han despertado gran interés en el Departamento de Defensa de Estados Unidos gracias a su característica de poder procesar un algoritmo para todas las posibles combinaciones de datos de entrada en el mismo tiempo que para un solo conjunto de datos. Esta característica permitiría romper rápidamente todas las claves criptográficas actuales; de ahí su interés militar. Sin embargo, están todavía en un estado de investigación muy poco avanzado. Incluso si se desarrollan lo suficiente como para poder romper claves criptográficas, es posible que nunca lleguen a reemplazar a la tecnología CMOS para construir ordenadores o móviles de propósito general, debido a las enormes inversiones que habría que hacer en dicha tecnología y en los correspondientes procesos y plantas de fabricación.

Pero se ha investigado en muchas otras tecnologías, desde procesadores totalmente ópticos hasta la utilización de moléculas orgánicas como elementos de memoria, pasando por la construcción de transistores a partir de nanotubos de carbono. Se han propuesto incluso tecnologías alternativas basadas en el silicio, que en lugar de propagar señales mediante pistas de cobre utilizan la inducción de un cambio en el espín electrónico a partir de dicho cambio en un átomo vecino, y que permitía el diseño de puertas lógicas muy compactas. Dicha tecnología prometía conseguir casi mil veces mayor escala de integración que la tecnología CMOS del momento, a pesar de lo cual no fue adoptada.

Así pues, aunque se ha investigado y se sigue investigando en tecnologías alternativas, a fecha de hoy ninguna de esas tecnologías está lo suficientemente desarrollada ni es previsible que lo esté durante las dos próximas décadas. Por ello, parece que los diferentes fabricantes van a seguir invirtiendo en la tecnología CMOS. La historia reciente ha demostrado que dicha tecnología se ha seguido manteniendo mucho más allá de las

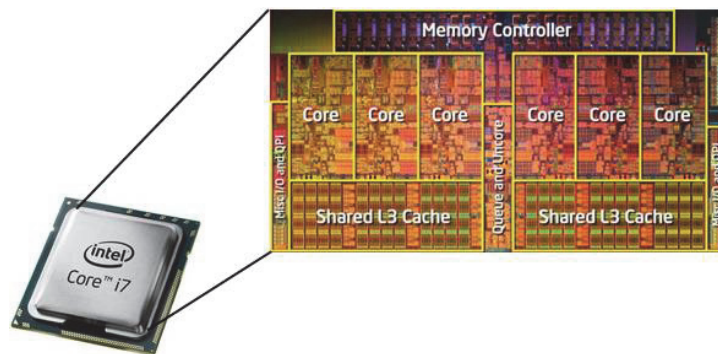
previsiones más optimistas del ITRS (*International Technology Roadmap for Semiconductor*), organismo que elabora la hoja de ruta de la tecnología de los semiconductores. En la actualidad no se prevén incrementos significativos en la frecuencia de reloj de los procesadores, como ya ha venido ocurriendo durante la última década. Sin embargo, se prevé que siga aumentando la escala de integración de forma sostenida durante al menos una década más. Eso sí, la utilización del creciente número de transistores disponibles seguirá limitada por la disipación de calor, o por su efecto directo, el incremento de temperatura.

La consecuencia directa es que se van a requerir avances tanto en tecnología como en arquitectura de computadores para diseñar transistores más rápidos y que se calienten menos, por un lado, y para descubrir la forma más eficiente de utilizar esos transistores sin que los ordenadores se calienten excesivamente, por otro.

En cuanto a la tecnología, cabe esperar nuevos avances en materiales dieléctricos que reduzcan aún más las corrientes de fuga en las puertas de los transistores CMOS y el correspondiente consumo de potencia estática, más avanzados que la tecnología High-K de Intel. También cabe esperar nuevos avances en la geometría de los transistores, en la línea de los transistores Tri-Gate de Intel, que permitan un mejor control del flujo de electrones en el canal de los transistores CMOS con un menor voltaje, con el consiguiente incremento de velocidad y reducción del consumo de los transistores.

LA ARQUITECTURA DE LOS PROCESADORES

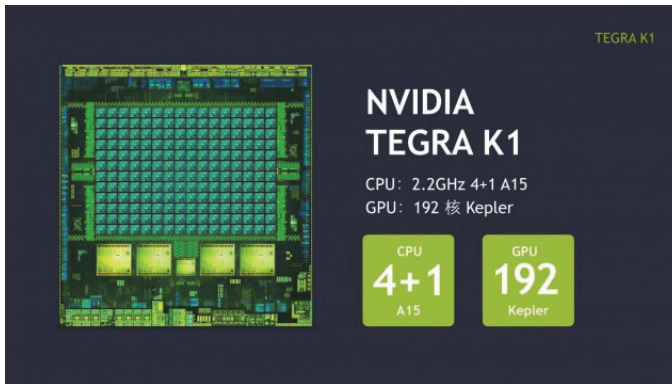
Con respecto a la arquitectura de los procesadores, en la mayoría de aplicaciones ya no sale a cuenta dedicar los transistores adicionales a incrementar el número de núcleos. De hecho, tras una década de fabricación de procesadores multinúcleo (*multicore processors*), tan solo se ha llegado a fabricar procesadores con poco más de 16 núcleos (Figura 13.1), excepción hecha de algunos diseños experimentales, o comerciales con muy poco éxito.



Fuente: www.intel.com

Figura 13.1. Arquitectura de un procesador multinúcleo

Varios son los motivos de este escaso crecimiento. En primer lugar, de poco sirve poner más núcleos si no podemos proporcionarles instrucciones y datos que procesar. El ancho de banda de acceso a memoria es crucial para obtener buenas prestaciones en un procesador multinúcleo, y dicho ancho de banda está actualmente limitado por el diseño del encapsulado del procesador. No es fácil añadir muchas más patillas al encapsulado sin incrementar excesivamente su coste. En segundo lugar, si aumentamos el número de núcleos se va a producir un incremento en la disipación de calor cuando estén todos activos, sobrepasando los límites aceptables en los encapsulados actuales. Obviamente, de poco sirve incrementar el número de núcleos si no van a poder activarse todos a la vez, o se va a poder hacer a costa de reducir mucho la frecuencia de reloj. Finalmente, para ordenadores con un solo usuario, pocas aplicaciones son realmente capaces de utilizar eficientemente varios núcleos.



Fuente: www.nvidia.com

Figura 13.2. Procesador multinúcleo con acelerador de cálculo y/o de gráficos en un mismo chip

Por ese motivo, los arquitectos han buscado y buscarán otras formas de aprovechar mejor esos transistores. La solución más inmediata fue incrementar el tamaño de la memoria cache de último nivel. También se han invertido transistores en dotar a los núcleos de instrucciones vectoriales más potentes y registros vectoriales de mayor tamaño. Tal es el caso de las instrucciones SSE (*Streaming SIMD Extensions*) y AVX (*Advanced Vector Extensions*) en sus diferentes ediciones.

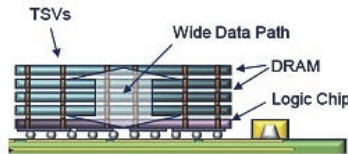
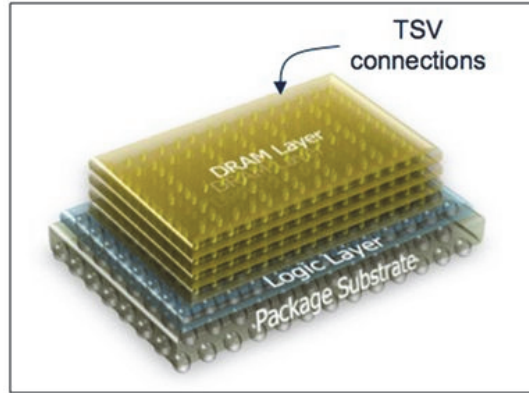
Una de las posibilidades que ya se están explotando comercialmente en portátiles y ordenadores de bajo coste para usar de forma efectiva los transistores adicionales es la integración en un mismo chip de un procesador con pocos núcleos de uso general y un dispositivo de procesamiento de gráficos con un gran número de núcleos especializados y muy sencillos (Figura 13.2). De este modo no hace falta incluir una tarjeta gráfica en el diseño del ordenador. Por otra parte, el diseño especializado del dispositivo gráfico, con potentes instrucciones vectoriales, es capaz de obtener mucha más potencia de cálculo sin incrementar proporcionalmente el consumo de energía. De hecho, esta propiedad ya se está usando en los clústeres para computación de altas prestaciones (HPC, *High Performance*

Computing) para conseguir mayores potencias de cálculo sin incrementar drásticamente el consumo de energía eléctrica. La diferencia es que en el caso de los clústeres, los aceleradores de procesamiento están ubicados en tarjetas separadas con su propia memoria, siguiendo la arquitectura de las tarjetas gráficas de las que derivan.

LA ARQUITECTURA DE LA MEMORIA

Pero el mayor problema de cara a conseguir altas prestaciones sigue siendo el limitado ancho de banda de acceso a memoria a través del encapsulado del procesador. Muchos investigadores han trabajado en este problema, proponiendo soluciones ingeniosas tales como poner interfaces para fibra óptica en el encapsulado, usar encapsulados parcialmente transparentes para transmitir señales ópticas a unas guías de silicio grabadas en la placa base, o incluso usar acoplamiento capacitivo entre encapsulados superpuestos para reducir drásticamente el tamaño de cada línea de señal del encapsulado. Ninguna de estos enfoques ha tenido éxito comercial debido a su elevado coste o la dificultad de alineamiento mecánico.

Por ello, el mayor avance que cabe esperar en los próximos años es una solución eficiente y económicamente viable a los problemas de ancho de banda de acceso a memoria. A la vista del fracaso o la inviabilidad comercial de los anteriores intentos de incrementar drásticamente el ancho de banda a través del encapsulado, y teniendo en cuenta que las memorias flash actuales utilizan una pequeña pila de chips de memoria para conseguir mayor capacidad de almacenamiento, es de esperar que en un futuro cercano se desarrollen y comercialicen procesadores que incluyan una pila de chips de memoria RAM dinámica (DRAM, *Dynamic Random Access Memory*) dentro del mismo encapsulado (Figura 13.3). Dicha memoria permitirá incrementar el ancho de banda a memoria en más de un orden de magnitud, al no tener que pasar a través del encapsulado para acceder a memoria, además de reducir notablemente el tiempo de acceso. Ello será posible gracias a que las memorias DRAM están organizadas en bancos, en cada uno de los cuales se accede una fila entera cada vez que se necesita leer o escribir un bloque de cache de una fila no activa. El ancho de banda dentro del encapsulado hará posible enviar incluso una fila entera al procesador en un solo acceso, en lugar de transferir los datos palabra a palabra. Esto permitirá además atender adecuadamente la demanda de un mayor número de núcleos.



Fuente: www.allaboutcircuits.com

Figura 13.3. Estructura de un procesador con una pila de memorias 3D superpuestas

Pero las pilas tridimensionales de memoria DRAM constituyen además una excelente forma de aprovechar el creciente número de transistores que estarán disponibles en el futuro, además de consumir poca energía y generar muy poco calor, comparado con el que generan los núcleos de procesamiento. Por ello será factible incorporar toda esta memoria dentro del encapsulado, sin exceder los límites de temperatura de los chips. Dado que la mayor demanda de ancho de banda en la actualidad es la que generan las tarjetas gráficas y aceleradores de cálculo derivados de las mismas, es de esperar que esta tecnología de pilas tridimensionales de memoria dentro del chip se aplique en primer lugar a las tarjetas gráficas, y posteriormente a los procesadores de uso general. Es más, el mayor ancho de banda que esta tecnología permitirá también hará viable la integración de dispositivos gráficos mucho más potentes en el mismo chip que los núcleos de uso general. Estos procesadores híbridos, que incorporan gráficos y memoria dentro del encapsulado también tendrán grandes ventajas para dispositivos móviles en cuanto a prestaciones, tamaño, peso y consumo de energía, siendo posible que se comercialicen incluso antes en este mercado que en el de los computadores portátiles o servidores basados en clústeres.

FUNCIONALIDAD ADICIONAL

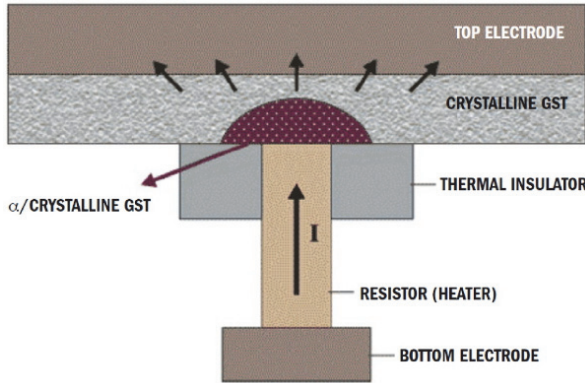
Pero el creciente número de transistores no solo se empleará en añadir memoria DRAM dentro del encapsulado. También permitirá incorporar en breve todos los controladores de periféricos (el denominado South Bridge) en el mismo chip que el procesador. Si esta incorporación no se ha hecho ya es porque los controladores de periféricos se fabrican en una tecnología menos reciente que la utilizada para los procesadores, lo que permite a los fabricantes seguir dando una cierta utilidad a las plantas de fabricación con tecnología ya obsoleta. En el futuro cabe esperar que se incorporen más aceleradores especializados de cálculo, además de los procesadores para gráficos ya incorporados. Estos aceleradores seguramente incluirán circuitos especializados para procesamiento de paquetes de comunicaciones, incluyendo el encapsulado y desencapsulado requeridos por los protocolos TCP (*Transmission Control Protocol*) e IP (*Internet Protocol*), utilizados actualmente en Internet. También incluirán circuitos especializados para ciertos cálculos, tales como las operaciones de criptografía o la búsqueda de virus, y/o circuitos programables, tales como pequeñas FPGA (*Field-Programmable Gate Array*). Estos circuitos especializados tienen dos grandes ventajas. En primer lugar consiguen acelerar mucho ciertos cálculos pero con un consumo bastante menor de energía, y en segundo lugar solo consumen energía y disipan calor cuando se usan.

LA TECNOLOGÍA DE LA MEMORIA PRINCIPAL

Para la memoria principal también se esperan grandes avances, además de la incorporación de parte de la misma dentro del encapsulado del procesador. Las actuales memorias DRAM se basaban inicialmente en la capacidad parásita de los transistores. Pero a medida que se aumenta la escala de integración y se reduce el tamaño de las celdas de memoria se está volviendo cada vez más complicado conseguir valores suficientemente grandes de dicha capacidad parásita en tan poco espacio. Por ello cabe esperar que durante la próxima década dicha tecnología se reemplace por otra que no tenga los problemas antes mencionados.

La tecnología que más promete en este sentido es la denominada memoria de cambio de fase (Figura 13.4). Esta tecnología es similar a la utilizada en los dispositivos CD (*Compact Disc*) y DVD (*Digital Versatile Disc*), almacenando información en función del estado de cristalización del material. Como en el caso de estos dispositivos, estas memorias no son volátiles, por lo que mantienen la información sin alimentación eléctrica. Pero hay diferencias muy importantes. En primer lugar, lo que se mide es el cambio en la resistencia eléctrica del material, por lo que no hace falta un diodo láser ni elementos mecánicos en movimiento para leer la información. En segundo lugar, tampoco hace falta un diodo láser para escribir la información. Basta con aplicar una cierta tensión durante el tiempo adecuado para que se caliente y cambie el estado de cristalización del material. Finalmente, el proceso es reversible, pues no se pasa de estado cristalino a amorfo, sino de un estado cristalino a otro y viceversa. Por todo ello, es probable que en el futuro la tecnología de memoria de cambio de fase sustituya a las memorias DRAM actuales, como éstas sustituy-

yeron a las memorias de núcleos de ferrita. Con ello se conseguirán memorias no volátiles, que no requieren refresco y con un consumo de energía inferior al de la tecnología actual. Además, gracias a la no volatilidad de la información en memoria, será posible apagar y volver a encender un ordenador, una tableta o un teléfono móvil en unos pocos segundos.



Fuente: www.ecnmag.com

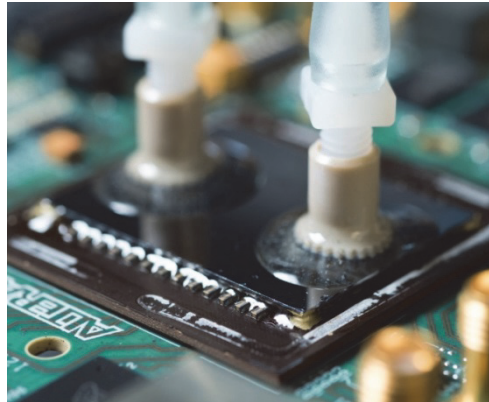
Figura 13.4. La tecnología de memoria de cambio de fase

Cuando la escala de integración lo permita, se integrará en un solo encapsulado un chip multinúcleo y una pila de chips de memoria RAM no volátil que constituirán toda la memoria principal del sistema. El chip multinúcleo incluirá los núcleos de procesamiento de uso general y sus múltiples niveles de memoria cache, los diferentes aceleradores de procesamiento incluido el procesador de gráficos, y los controladores de memoria y de los diferentes dispositivos periféricos, incluidas las interfaces de comunicaciones. Cuando este procesador forme parte de un sistema multiprocesador de memoria compartida o de un clúster, bastará con interconectar varios de estos dispositivos utilizando la interfaz de comunicaciones adecuada.

LA DISIPACIÓN DEL CALOR

Una vez resuelto el problema del ancho de banda de acceso a memoria principal mediante la inclusión de la memoria dentro del mismo encapsulado, el número de núcleos tanto de uso general como especializados seguirá estando limitado por la disipación de calor. En los casos en que se requiera una mayor densidad de potencia de cálculo, ésta se podrá conseguir con un mayor número de núcleos de procesamiento por encapsulado junto con un sistema de refrigeración más potente. La mayoría de los computadores actuales utilizan disipadores de cobre con ventilación forzada de aire, reservándose la refrigeración líquida para aquellos computadores en los que se requiere una mayor densidad de potencia de cálculo, pero con un importante sobrecoste. En el futuro, cabe esperar que la refrigeración por ventilación forzada de aire siga siendo la tecnología dominante, justificada por su

menor coste. Pero se desarrollarán nuevos sistemas de refrigeración líquida con circulación de fluido por dentro del encapsulado, para conseguir una resistencia térmica mucho más reducida mediante el contacto directo del fluido con los chips que generan el calor (Figura 13.5). Además, como los diferentes chips (multinúcleo, memoria) no generarán la misma cantidad de calor, se utilizarán microválvulas para regular el caudal por diferentes zonas de los chips en función de su temperatura, consiguiendo así la máxima eficiencia.



Fuente: www.phis.org, autor Rob Felt, Georgia Tech

Figura 13.5. Refrigeración líquida con circulación de fluido dentro del encapsulado

LAS INTERFACES PERSONA-COMPUTADOR

Finalmente, en cuanto a las interfaces con el ordenador, tableta o móvil, la mayor potencia de cálculo permitirá un incremento en la utilización de interfaces habladas. También cabe esperar un incremento en la utilización de pantallas de gran resolución y muy reducido tamaño. Estas pantallas, situadas muy cerca del ojo (por ejemplo, incorporando pantallas transparentes en el cristal de las gafas), permitirán crear la ilusión de una pantalla de gran tamaño en un dispositivo muy compacto.