

2021 Retos Vitales

para una nueva era

**La programación informática como reto
para la generación del bienestar futuro**

Jordi Martí Pidelaserra



Claves para entender y mejorar el mundo



Reial Acadèmia Europea de Doctors
Real Academia Europea de Doctores
Royal European Academy of Doctors

BARCELONA - 1914



La programación informática como reto para la generación del bienestar futuro



Dr. Jordi Martí Pidelaserra

Profesor de la Universidad de Barcelona.

Académico de Número de la Real Academia Europea de Doctores (RAED).

INTRODUCCIÓN

Este capítulo pretende explicar cómo una larga tradición de investigación científica llega a la configuración de los ordenadores actuales. Los grandes avances en la informática se sitúan a partir de la Segunda Guerra Mundial. Es un dato preocupante que nuestro bienestar presente provenga de una actividad bélica. Todos esos conocimientos son aprovechados en la Guerra Civil española y en la Segunda Guerra Mundial pero no son su objetivo. Frente a la riqueza material que supone la construcción de máquinas de calcular mecánicas y de ordenadores electromagnéticos surge una nueva riqueza de tipo inmaterial: la programación informática.

Nuestro bienestar futuro y nuestra paz está en manos de los nuevos lenguajes de programación. El reto vital es utilizar correctamente esta inmaterialidad para mejorar las condiciones de vida de todo el planeta.

La digitalización de la sociedad es hoy una realidad, sus orígenes son más remotos de lo que a menudo se ha explicado públicamente. Numerosos científicos a lo largo de la historia han ido desarrollando instrumentos de cálculo (ábacos, reglas logarítmicas) y también ingenios mecánicos de todo tipo (véase el anexo I). A mediados del siglo XX se construyen algunos grandes ordenadores que desarrollan una nueva actividad antes solo vislumbrada

por Ada Lovelace cuando en 1854 escribe unas *Notas sobre la máquina de calcular de Charles Babbage*.

La historia de la informática se ha percibido como la de la construcción de máquinas cada vez más potentes, olvidando que tras ellas hay un gran trabajo de programación (véase el anexo II). La riqueza material de los ordenadores debe comprenderse con la riqueza inmaterial de los lenguajes de programación.

Ada Lovelace (nacida Augusta Ada Byron, única hija legítima de Lord Byron) fue una matemática ávida, considerada como la primera programadora de ordenadores del mundo, después de que escribiera un algoritmo para la máquina analítica mecánica de Charles Babbage. Fuente: Wikipedia.



Las primeras personas que se dedicaron a la programación eran mujeres y su invisibilidad era fruto del impacto que generan los nuevos productos informáticos cada vez más potentes y cada vez más pequeños. La miniaturización de las máquinas va destacando cada vez más la importancia de los lenguajes de programación.



Desde los POL («*Program Oriented Languages*») hasta llegar a la actual programación por objetos, los paradigmas van cambiando y nuestra sociedad debe saber estar a la altura ante los nuevos retos que se avecinan.

Nuestro bienestar futuro, nuestra paz está en manos de los nuevos lenguajes de programación. El reto vital es utilizar correctamente esta inmaterialidad para mejorar las condiciones de vida de todo el planeta. Imagen cedida por Pixabay.

PROGRAMACIÓN POR OBJETOS

La programación por objetos ha ido cambiando el perfil de los informáticos. Los primeros programas eran instrucciones dirigidas a activar impulsos eléctricos en los ordenadores. Poco a poco se vislumbró la necesidad de ir generando lenguajes de programación cada vez más comunes. Se trataba de evitar que cada constructor de ordenadores desarrollase un lenguaje propio.

Con el tiempo se acordaron estándares de programación y, durante la década de 1970, había un lenguaje muy desarrollado para la gestión empresarial llamado COBOL y otro llamado FORTRAN para la actividad académica. Los desarrolladores de nuevos lenguajes se basaban en las experiencias anteriores y esos listados de instrucciones se iban poniendo en común y se generaban pequeñas rutinas que podían ser incorporadas a un nuevo programa ahorrando mucho tiempo de diseño. Con el tiempo las rutinas llegan a ser multiplataforma y ayudan a mejorar aplicaciones en los más diversos entornos informáticos.

Esta «cooperación» entre programadores generó un nuevo mercado. Se intercambiaban rutinas para conseguir avanzar más rápido y asegurar que el resultado tuviera muchos menos errores. Aparecieron organizaciones especializadas en desarrollar esas rutinas y los programadores se iban convirtiendo en ensambladores de instrucciones previamente compiladas por otros programadores.

Llegados a los años ochenta del siglo pasado, los ordenadores se convirtieron en pequeñas máquinas que podían instalarse en cualquier casa particular, en cualquier oficina, en cualquier negocio. Aparecieron los ordenadores personales que sustituían a los centros de cálculo donde se habían armado grandes instalaciones con ordenadores que daban servicios a múltiples clientes.

En esos años, los fabricantes de procesadores podían incorporar en una caja metálica, cada vez más pequeña, la potencia de cálculo que cinco años antes necesitaba todo el volumen de una vivienda. La obsolescencia de los

procesadores obligaba a desmontar las antiguas instalaciones y a repensar la nueva informática.

Pero mientras los chips iban ofreciendo capacidades de cálculo con crecimientos exponenciales en espacios más reducidos (en 1985 ya se comercializan ordenadores portátiles), los programadores informáticos no podían avanzar al mismo ritmo. No se podía aprovechar la capacidad de los ordenadores personales con la programación tradicional.

Apareció en esos momentos la programación por objetos. Los lenguajes de programación ya no se basarían en las instrucciones y rutinas anteriores. Ahora a mediados de los años ochenta del siglo XX, los programadores van desarrollando programas más complejos que una rutina y que pueden ser aprovechados por otros programadores que no deben conocer las interioridades de ese subprograma, para aprovecharlo. Solo deben saber ensamblar los subprogramas correctamente y el resultado está garantizado.

Más «cooperación» permitió ver que no era necesario volver a programar dígito a dígito. Se trabajaría a partir de lo creado por especialistas. Se empieza a programar sobre lo ya programado por otros que saben más de presentaciones gráficas, o de operaciones de cálculo, o de impresión de textos, por ejemplo.

Esta es la parte más evidente de la programación por objetos. Pero en su interior los lenguajes que permitían ese ensamblaje de subprogramas debían superar algunas concepciones muy tradicionales. Los creadores de lenguajes informáticos eran personas que provenían de haber cursado estudios de ingeniería, física o matemáticas. Su formación se basaba en los números. Toda la programación se había desarrollado a partir de expresiones numéricas.

A finales de los años ochenta del siglo XX, algunos de esos matemáticos, físicos o ingenieros empezaron a descubrir que la base de todo su trabajo no debía ser forzosamente un número. Podían trabajar con vectores y con matrices, que son conjuntos de números. Podían programar sumas de números,

pero también de vectores y también de matrices. Y así podían ir realizando subprogramas donde se ejecutasen estas operaciones básicas.

Con el tiempo fueron comprobando que podían operar con «objetos» que no necesariamente eran los números sobre los que habían estudiado las matemáticas y los procesos de cálculo tradicionales.

Estos mismos programadores empiezan a utilizar conceptos que no habían sido divulgados abiertamente en las facultades de matemática. Las duplas, las tuplas, los «arrays» o los «dataframe» son »objetos» que pueden ayudar a mejorar los cálculos ya conocidos y a descubrir nuevos cálculos.

Se pueden sumar esos «objetos», se pueden restar y se pueden programar acciones sobre ellos independientemente del contenido que haya dentro del objeto. Una vez ejecutadas las operaciones con el «objeto», será necesario traducir e interpretar los resultados según la naturaleza interna del mismo.

Si hasta el momento las facultades de matemáticas se habían centrado principalmente en el estudio de las variables cuantitativas (a excepción de la geometría descriptiva y la topología) que se expresaban a través de números, ahora con los objetos, las variables ya no debían ser necesariamente numéricas y podían ser variables cualitativas. Se abría la puerta a la realización de cálculos, hasta ese momento reservados para las disciplinas cuantitativas, a muchas otras disciplinas que no habían podido utilizar las variables cuantitativas tradicionales.

INFORMÁTICA Y MATEMÁTICAS

En los estudios de matemáticas, siempre, a lo largo de la historia, se han buscado formas de cálculo que pudiesen ayudar a encontrar soluciones seguras. Si además se podía hacer el cálculo de forma rápida, mejor y, si además se conseguía de forma sencilla, entonces se estaba ante una resolución brillante.



La programación informática permitió avanzar mucho en el cálculo. En las facultades de matemáticas era necesario tener expertos informáticos para avanzar en esa dirección. Imagen cedida por Pixabay.

Los problemas más complejos de la matemática podían intentarse solucionar a base de repetir muchos cálculos, conseguir iteraciones a través de algoritmos que ya no ejecutaban los humanos sino esas máquinas llamadas ordenadores. La estadística ha sido la disciplina que más se ha beneficiado de estas innovaciones. Las simulaciones automatizadas han llegado a unos niveles de sofisticación (*bootstrapping*) que permiten proyectar soluciones muy rigurosas para los problemas que conllevan aleatoriedad.

Sobre estas bases, la programación por objetos ha permitido desarrollar aplicaciones que pueden resolver problemas mediante soluciones algebraicas. Ya no era necesario ir alimentando una fórmula con muchos números para encontrar cada una de las soluciones y a partir de esas soluciones numéricas generalizar la solución algebraica. Ahora se podía operar con los ordenadores de la misma forma cómo se había ido haciendo con la matemática moderna, con abstracciones. Una integral indefinida, una derivada, un límite ya no era necesario calcularlo mediante la simulación de muchas soluciones numéricas. Con la programación por objetos ya podía resolverse de forma algebraica, se hallaba la solución de x , de y o de la variable (del objeto) que se hubiese identificado.

Una variable x es un objeto, una integral de x puede ser otro objeto y hay subprogramas que van resolviendo estos problemas de cálculo. Los matemáticos pueden ir buscando nuevos problemas sin las restricciones tradicionales del cálculo. Esto conlleva que la frontera entre informática y matemáticas se rompa.

Los planes de estudio de matemáticas han ido incorporando los contenidos de programación informática y viceversa. Con ello se está dando un ejemplo de cómo deben irse modificando las estructuras docentes para adecuarse a las nuevas realidades.

LOS NUEVOS PLANES DE ESTUDIO

Las facultades de ciencias naturales en la medida que provienen de una tradición basada en el uso de variables cuantitativas deben aprovechar los nuevos desarrollos de cálculo para conseguir nuevos avances. Por tanto, el uso de los nuevos programas informáticos resulta ya imprescindible para todo tipo de estudios.

La programación por objetos ha cambiado la forma de entender el cálculo y ha conseguido desarrollarse en todas las disciplinas académicas. Resulta evidente en las disciplinas que mayor relación tenían con las matemáticas, pero el hecho de que puedan tratarse variables cualitativas lleva la informática hacia las ciencias sociales y las humanidades.

Algo tan complejo como un idioma, que no había podido ser comprendido con los métodos analíticos propios de las ciencias naturales, ahora puede ser estudiado aprovechando las capacidades de los programas por objetos. Incluso en el campo del color se pueden encontrar fórmulas cuantificadas que permiten reproducciones con grados de homogeneidad nunca hallados.

Una especialidad en la que las matemáticas y la informática ya llevan años de convivencia es la dedicada a las redes neuronales. Hoy, los traductores informatizados de un idioma a otro que consiguen más fiabilidad utilizan



estos conocimientos. Esto implica que los estudiantes que hasta ahora se especializaban en materias que se consideraban muy distantes pueden reencontrarse para resolver problemas en común.

Los traductores automáticos precisan filólogos, traductores e intérpretes trabajando codo con codo con matemáticos e informáticos. Imagen cedida por Pixabay.

Otro ejemplo, entre muchos, de esta superación de fronteras académicas puede hallarse en el estudio del estilo artístico de un escritor determinado.

A través de la programación por objetos, la producción literaria se somete al escrutinio de un programa informático. Este irá presentando relaciones sintácticas, gramaticales, semánticas que pueden ser características esenciales que se intuían por los grandes especialistas en ese autor pero que no eran fácilmente describibles. Por tanto, los nuevos programadores informáticos aplican sus conocimientos sobre materias académicas que hasta ahora parecían no conciliables.

En estas nuevas estructuras de colaboración académica tiene mucho que ver otra de las especialidades matemáticas que más vigor ha conseguido con la programación por objetos: la teoría de grafos. Con los datos que se han ido acumulando mediante la interconexión de los ordenadores, los especialistas en teoría de grafos ponen de relieve relaciones entre variables que hasta ahora no se contemplaban. Estas relaciones se representan mediante pequeños gráficos, pero no tienen sentido si no son interpretadas por los mejores conocedores de esas variables, que no son los especialistas en grafos.

Los planes de estudio deben tener en cuenta que los nuevos avances académicos se generarán por el contacto entre estudiosos de disciplinas diversas. Imagen cedida por Pixabay.



La enseñanza del futuro no podrá concentrarse en la transmisión de conocimientos o en la formación de profesionales. Habrá que preparar a las nuevas generaciones para saber escuchar a esos especialistas que parece que no hablan, ni piensan de la misma manera. Esos profesionales especializados que solo aplican unas técnicas muy sofisticadas disponen de terreno de juego durante períodos muy breves. Los profesionales que sepan cooperar con los de disciplinas distintas avanzan con más rapidez y superan esas técnicas tan concretas.

LA MULTIDISCIPLINARIEDAD Y LA INTERDISCIPLINARIEDAD

Un programador debe estar siempre dialogando con el usuario que precisará sus desarrollos. Esta es una tarea muy ardua. No siempre es fácil comprender al usuario y viceversa. Las quejas de unos y de otros quedan reflejadas en numerosas anécdotas y relatos humorísticos. Pero tales discusiones evidencian la necesidad de un nuevo entorno formativo.

La física y las matemáticas han tenido, históricamente, muchos puntos de contacto. La química y la física otros tantos. La biología y la medicina también comparten conocimientos y aplicaciones. Pero las relaciones entre matemáticas y filología nunca se habían desarrollado como puede hacerse en estos momentos. Los estudios de jurisprudencia pueden mejorar muchísimo si se programan determinadas aplicaciones que hagan más evidente cuál es el resultado de un juicio. Y así podrían irse detallando nuevas relaciones que nuestro mundo va reclamando de forma más evidente.

Se presenta un cambio relevante que obliga a repensar esas disciplinas acomodadas a unos planes de estudio y a unas producciones académicas distintas de las que se han desarrollado hasta el momento. Desde la redacción de la *Enciclopedia* por los revolucionarios franceses no se enfrentaba la comunidad académica ante un reto de tales magnitudes.

En la actualidad se dispone de unos ordenadores con unas capacidades de cálculo enormes, y de unos programas informáticos que las aprovechan para tratar unas bases de datos, cuantitativas y cualitativas, de dimensiones impensables para nuestro cerebro.

La naturaleza humana es incapaz de tomar decisiones cuando se presentan de golpe múltiples informaciones. Los servicios secretos de los Estados más avanzados se autodenominan servicios de inteligencia, pero se paralizan cuando llegan olas de información que no estaban esperando.

La informática actual permite tratar muchísima información simultánea (*big data*), y se pueden resolver problemas a unas velocidades nunca vistas (in-

teligencia artificial). Para aprovechar toda esta capacidad de cálculo es imprescindible que la comunicación entre especialistas se desarrolle de una forma innovadora. Para ello, la especie humana ha de conseguir plantear correctamente los problemas.

Seguramente todos los humanos aceptarían que el principal problema es mejorar la calidad de vida en el planeta en que viven. En una frase tan simple hay incorporadas restricciones de todo tipo. Restricciones que obligan a trabajar y a estudiar desde múltiples puntos de vista.

La multidisciplinariedad no es lo mismo que la interdisciplinariedad aunque se usan de forma sinónima. Determinados problemas pueden ser abordados desde distintas disciplinas académicas, son problemas multidisciplinarios. La interdisciplinariedad indica que la solución del problema no es total si no concurren distintas disciplinas de manera coordinada.

Reconocer que muchos problemas son multidisciplinarios ya es un gran logro, pero en los momentos actuales es preciso que esas disciplinas colaboren y cooperen. Los programas informáticos pueden ser utilizados por muchas disciplinas, pero cada una obtendrá unos resultados completos para su entorno pero parciales para el conjunto de la sociedad.

La formación de los actuales estudiantes ya es multidisciplinaria, pero no llega a conseguir la interdisciplinariedad. Los lenguajes que usan los científicos son altamente especializados y van configurando unas mentes que, lejos de abrirse a nuevos entornos, únicamente se muestran seguras y confortables en materias cada vez más restringidas. Por eso sería menester, epistemológicamente hablando, prestar atención a la investigación de nuevas síntesis, adecuadas a la gran capacidad de análisis de la que se dispone.

Cuando se organiza un equipo de investigación solo con médicos se pueden conseguir avances, pero seguramente la incorporación en el equipo de ingenieros podría facilitar que esos avances se materialicen en la construcción de unos dispositivos más eficientes y que aparezcan más rápido. Si en

un equipo de médicos también colaboran químicos, físicos, biólogos matemáticos o estadísticos las terapias para curar una dolencia podrían ser más robustas que si solo se basan en criterios clínicos.

Hoy, que se dispone de traductores automatizados de un idioma a otro, parece increíble que haya tanta incomunicación entre profesionales de distintas áreas académicas. Al alcance de todos están las grandes redes de telecomunicación que funcionan a la perfección para desarrollar proyectos en núcleos homogéneos de conocimiento. Pero la experiencia demuestra que aún serían más productivas si esos núcleos homogéneos fuesen permeables y entrasen en contacto con otros.

FUENTE DE CONOCIMIENTO

Nuestra especie ha ido grabando sobre distintos formatos su conocimiento. Cuando se tomó conciencia de que oralmente no se podía transmitir todo el conocimiento, la escritura fue el sistema para poder agrupar y transmitir lo que se iba descubriendo. La imprenta permitió la divulgación de esos escritos como nunca se había realizado.

En el siglo XX, la grabación de información en fichas perforadas, en cintas magnéticas, en discos de pasta, de vinilo, en formatos magnéticos flexibles o duros, permitió aún más la difusión de la información. Pero el cine, la televisión y el vídeo permitieron que la información fluyese por canales nuevos. En el siglo XXI las telecomunicaciones permiten que toda esa información esté disponible en los rincones más recónditos del planeta.

Sin embargo, la necesidad de privacidad ha empujado a promulgar regulaciones donde se prohíbe preguntar a los humanos sobre ciertos aspectos de su comportamiento individual. Esto contrasta con la facilidad que tienen los científicos para poder consultar fuentes primarias de conocimiento en tiempo real.

Resulta que no se puede preguntar a nuestros conciudadanos sin resultar ser políticamente incorrectos. Pero todos dejan rastro de su actividad en múltiples dispositivos que están interconectados. Por consiguiente, hay bases de datos que tienen la información que no puede preguntarse directamente.

Es conocido por todo el mundo que en algunos almacenes de datos hay un conocimiento muy preciso sobre nuestro comportamiento individual. Puede saberse lo que no se puede preguntar a un humano, preguntándolo a la base de datos donde se graba esa información.

Hasta ahora los humanos escribían, grababan y publicaban a su parecer. Ahora, se escriben y se graban datos sin ser la persona consciente de este proceder. Alguien que tuviese acceso a esas bases de datos podría realizar preguntas que no serían correctas en una entrevista personal.

Esto significa que en los últimos treinta años se ha ido generando un conocimiento que está plenamente disponible para su uso. Se han desarrollado los programas y los ordenadores para tratar esa información. Pero la humanidad no sabrá utilizarla correctamente si no se organizan equipos interdisciplinarios que sepan formular correctamente las cuestiones que pueden ser resueltas con esa fuente de conocimiento.

La fuente puede manar con tanta fuerza que impida beber correctamente. Hay que prepararse para ese diálogo con las bases de datos. Bases de datos que pueden contener datos sobre las personas, sobre el clima, sobre muchísimas especies y realidades, pero que deben ser correctamente gestionadas. De lo contrario, ocurrirá lo mismo que ocurre con el cerebro humano, que se bloqueará cuando le llegue un aluvión de información.

Sin ser conscientes de ello, los científicos en estas últimas décadas han ido aproximando más sus conocimientos. La informática ha ayudado a ello, las telecomunicaciones han contribuido también de forma relevante. Pero ahora hay que ser conscientes de la imperiosa necesidad de dialogar y poner en común los conocimientos con aquellos que parece que no pueden en-

tendernos rápidamente. Estos últimos, lejos de ser unos ignorantes en nuestra materia, pueden ser unos iluminadores de nuevos caminos.

CONCLUSIÓN

Los lenguajes de programación han ido variando sus paradigmas. Hoy se dispone de una gran capacidad de cálculo y de potentes lenguajes de programación. Acompasar los desarrollos entre nuevos tipos de procesadores y nuevas programaciones es un gran reto vital.

A lo largo de la historia se han visto saltos hacia delante en proceso de datos que no podían ser seguidos por los avances en programación. Adecuar los planes de estudio para que la informática se introduzca como un instrumento para avanzar en los conocimientos más diversos es un reto vital.

Conseguir que los nuevos lenguajes de programación sean vías de paz y prosperidad exige que se desarrolle una formación multidisciplinaria e interdisciplinaria. Los nuevos lenguajes deben servir para acercar conocimientos y generar nuevas soluciones a problemas muy antiguos que los humanos no hemos resuelto convenientemente hasta el momento.

ANEXO I

LA RIQUEZA MATERIAL

Cuando alguien se refiere a los inicios de la informática piensa en Alan Turing, quien en 1936 teoriza sobre las máquinas que podían resolver problemas. De ahí surge el concepto de «máquina de Turing».

Esta concepción innovadora lleva en su seno el final de una época. Turing aún piensa como la mayoría de la población del siglo XIX y XX, que las máquinas, los ingenios mecánicos, mueven el futuro.

Quien se pregunta sobre la evolución de la informática obtiene respuestas basadas en grandes innovaciones materiales. Los computadores u ordenadores son la última expresión del maquinismo desarrollado a partir de la primera Revolución industrial.

Sin duda, todos los historiadores de la informática sitúan a Charles Babbage como el gran desarrollador de un prototipo de máquina analítica en 1837 (anteriormente había proyectado dos máquinas de diferencias). Pero muy pocos explican que para hacer funcionar correctamente esa máquina había que generar un conjunto de instrucciones que con el tiempo darán lugar a los lenguajes de programación.

De la misma forma que es muy difícil determinar cuándo aparece el primer ordenador, tal como hoy los entendemos, también es muy complicado determinar el inicio de los lenguajes de programación. Babbage introduce las funciones matemáticas en su máquina gracias a unas tarjetas perforadas que había desarrollado la industria de telares francesa (Jean-Baptiste Falcon y Joseph Marie Jacquard entre los más reconocidos). Cuatro años más tarde, en 1841, Ada Lovelace (a quien nos hemos referido al principio del capítulo) colabora con Babbage y publica unas notas que son consideradas el primer lenguaje de programación.

Desde ese 1841 aparecen muchas otras máquinas mecánicas con el objetivo de realizar operaciones de cálculo que apoyasen la investigación y la producción. Maurice d'Ocagne (1862-1938) sistematiza desde la Academia de Ciencias de París una clasificación para las máquinas de calcular estableciendo cinco grandes grupos para entender la evolución de la actividad en esta senda de conocimiento. El criterio de clasificación depende de la naturaleza gráfica o mecánica de las relaciones que establece la máquina y del sistema de inscripción de los datos que puede ser mediante un inscriptor de dígitos (si se introducen los números de 0 a 9) o mediante un inscriptor a cotas (si se señalan los números sobre una escala métrica o funcional).

Maurice d'Ocagne indica el inicio de lo que hoy conocemos como digitalización. Una actividad centrada en la inmaterialidad. Las máquinas cada vez son de tamaño más reducido y en cambio son más importantes las formas para programar estos datos.

En 1946 se hace público que en Estados Unidos disponen del ENIAC (siglas de *Electronic Numerical Integrator And Computer*). En esos momentos el concepto ordenador no era utilizado y se hablaba de analizadores e integradores. Esta máquina superaba los engranajes y bielas mecánicos utilizando electrónica. Pero su gran aportación no es la material. Con el ENIAC llega la imperiosa necesidad de programar informáticamente.

Desde Ada Lovelace, la programación no destacaba ante los ingenios mecánicos. Con la electrónica la forma de introducir los datos, la digitalización de las instrucciones de funcionamiento aparece como la parte básica de los nuevos desarrollos. Con el ENIAC ya no es necesario mover manualmente engranajes o realizar nuevas soldaduras, son seis mujeres las que desarrollan la programación del ordenador: Betty Snyder Holberton, Jean Jennings Bartik, Kathleen McNulty Mauchly Antonelli, Marlyn Wescoff Meltzer, Ruth Lichterman Teitelbaum y Frances Bilas Spence.

Betty Snyder Holberton es quien continuará este trabajo y participará posteriormente en el proyecto ALGOL. Pero antes de llegar a este nuevo desarrollo el 21 de junio de 1948 en Manchester se ejecutó un programa informático con 17 instrucciones elaborado por Tom Kilburn (1921-2001) para calcular el máximo divisor de 2 elevado a 18. Este se considera el primer programa informático desarrollado para un ordenador digital electrónico con memoria (tubos de rayos catódicos de Williams).

Este tipo de ordenador es el que nos ha llegado a nosotros siguiendo la arquitectura diseñada por John von Neumann (registrado al nacer en 1903 como Neumann János Lajos, en Budapest y fallecido en 1957 en Washington D.C.) en *First draft of a report on the EDVAC*, publicado en 1945 y que desarrolla la que había concebido Konrad Zuse. Mientras Turing teorizaba en Reino

Unido, el científico alemán Zuse ya en 1936 diseñó una máquina (construida en 1938) con separación de la unidad central de cálculo, los dispositivos de entrada y salida y la zona de memoria.

Pero los programas informáticos de entonces son muy simples comparado con los que actualmente se utilizan. En esos momentos las programadoras de ENIAC o los del SSME (siglas de *Manchester Small-Scale Experimental Machine*) utilizaban términos como «palabra» para referirse a las combinaciones de bits y de allí surge que una combinación de palabras sea un lenguaje informático.

Lenguajes informáticos contruidos mediante palabras, así empezó el desarrollo de una riqueza inmaterial que nadie había vislumbrado.

La digitalización, la entrada de datos descrita por Ada Lovelace, fue continuada por las programadoras de ENIAC y sus aportaciones son las que hoy hacen funcionar todo nuestro sistema económico. Nuestra calidad de vida ya no proviene de grandes máquinas, proviene de grandes lenguajes informáticos.

En la tabla A.I se resume, no exhaustivamente, la evolución de las máquinas de calcular hacia los ordenadores.

Tabla A.I. Evolución de los ordenadores

Año	Denominación	Inventor	Desarrollo	Operaciones
1642	Pascalina	Pascal (1623-1662)	Fabricadas 50 unidades	Sumas y restas
1694	<i>Stepped drum</i>	Leibniz (1646-1716)	Prototipo	4 operaciones algebraicas
1761	Cronómetro de precisión	John Harrison (1693-1776)	Premio Gobierno británico (20.000 libras)	Longitud con precisión de 2 minutos
1769	Cronómetro isócrono	Pierre Le Roy (1717-1785)	Premio Academia Francesa de Ciencias	Longitud con precisión de 8 segundos

Cont...

Año	Denominación	Inventor	Desarrollo	Operaciones
1784	Máquina de sumas	Johann Helfrich von Müller (1746-1830)	Prototipo	Generar tablas matemáticas
1804	Telar Jacquard	Joseph Marie Jacquard (1752-1834)	Fabricado	Máquina que controla el telar mediante cartones perforados
1822	Primera máquina de diferencias	Charles Babbage (1791-1871)	Prototipo	Raíces de un polinomio a través de sumas y restas
1833	Segunda máquina de diferencias	Charles Babbage (1791-1871)	Prototipo	Ecuaciones de segundo grado con precisión de 6 cifras
1837	Máquina analítica	Charles Babbage (1791-1871)	Prototipo	No ejecutaba una operación aislada. Ejecutaba un conjunto sin intervención humana. Primer programa informático por Ada Lovelace (1815-1852)
1838	Máquina analítica	Per Georg Scheutz (1785-1873)	Construida 1853	Ecuaciones en diferencias de cuarto orden
1876	Analizador diferencial (<i>Tide predictor</i>)	James Thomson, hermano de Lord Kelvin	Prototipo	Previsión de mareas
1880	Planímetro	Bruno Abakanowicz (1852-1900)	Fabricado por Gottlieb Coradi en 1915	Integrales indefinidas
1881	Aritmómetro	Pafnuti Chebyshev (1821-1894)	Construido en 1930	4 operaciones algebraicas
1884	Máquina censo	Herman Hollerith (1860-1929)	Construida	Censo estados americanos
1893	Máquina algebraica	Leonardo Torres Quevedo (1852-1936)	Prototipo	Raíces de ecuaciones algebraicas de orden determinado
1907	Balanza algebraica	Paulino Castells (1877-1956)	Construidas 3 unidades	Ecuaciones algebraicas una incógnita

Cont...

Año	Denominación	Inventor	Desarrollo	Operaciones
1912	Ajedrecista	Leonardo Torres Quevedo (1852-1936)	Renovado en 1920 por su hijo Gonzalo Torres-Quevedo	Torre y Rey blancas por parte del autómatas contra Rey negro. Uso de electroimanes. Algoritmo árbol de decisiones
1914	Aritmómetro electromecánico	Leonardo Torres Quevedo (1852-1936)	Prototipo	4 operaciones algebraicas
1919	Máquina de congruencias	Pierre y Eugène Carissan (1806-1876)	Prototipo	Soluciona ecuaciones diofantinas
1923	Máquina <i>Enigma</i>	Arthur Scherbius (1878-1929)	Fabricada	Scherbius no contaba con financiación. Se asoció con Willie Korn, propietario de <i>Enigma Chiffiermaschinen AG</i> en Berlín. Ambos mejoraron el diseño y en 1923 la presentaron en la Exhibición Postal Internacional de Berlín para el cifrado de secretos comerciales
1927	Analizador diferencial	Vannevar Bush (1890-1974)	Construido por el MIT, engranajes y bielas	Ecuación diferencial de sexto grado o sistema de tres ecuaciones diferenciales de segundo grado. Cálculo de la trayectoria de proyectiles
1934	Analizador diferencial	Reino Unido, Noruega, Suecia, etc.	Armadas navales. En 1938, el noruego Sven Rosslund construye un analizador diferencial mejorando el de Bush pero queda enterrado en su jardín durante la invasión alemana	Cálculos balísticos
1935	Polipasto algebraico	Paulino Castells (1877-1956)	Construido	Sistemas de tres ecuaciones lineales con tres incógnitas
1936	Teoría	Alan Turing	Teoría	<i>On Computable Numbers, with an Application to the Entscheidungs problem</i>

Cont...

Año	Denominación	Inventor	Desarrollo	Operaciones
1938	<i>VersuchsModell 1 = Z1</i> [Modelo experimental 1]	Konrad Zuse (1910-1995)	Fabricado en 1938 y destruido en 1943 por bombardeos sobre Berlín. Posteriormente se comercializaron hasta el modelo Z4	Calculador mecánico que se adelanta a la arquitectura de Von Neumann. Presenta lector de cintas perforadas, independiente de la unidad de control y la unidad de entrada y salida de datos
1938	Algebraico eléctrico	Paulino Castells (1877-1956)	Prototipo	Solución de sistemas de ecuaciones lineales
1942	RDA2	<i>Rockefeller Differential Analyzer number 2</i>	Uso secreto durante la Segunda Guerra Mundial. Cálculo de trayectorias de V2	Sustitución de algunos engranajes por conexiones eléctricas y un dispositivo que leía una cinta de papel con instrucciones. Contenía 18 integradores
1946	ENIAC (<i>Electronic Numeric Integrator and Computer</i>), sistema decimal	John W. Mauchly (1907-1980) y John Presper Eckert (1919-1995)	Funcionó entre febrero y noviembre de 1946. Se amplió memoria en 1947 y funcionó hasta 1955	Máquina que utilizaba tubos de vacío. Aparecieron los primeros programas informáticos realizados por un grupo de mujeres: Betty Snyder Holberton, Jean Jennings Bartik, Kathleen McNulty Mauchly Antonelli, Marlyn Wescoff Meltzer, Ruth Lichterman Teitelbaum y Frances Bilas Spence
1945	Teoría	John von Neumann	Teoría de la arquitectura del ordenador	<i>First draft of a report on the EDVAC</i> , aplicada a EDVAC en 1949-1951
1948	SSEM (<i>Manchester Small-Scale Experimental Machine</i>)	Frederic C. Williams (1911-1977), Tom Kilburn (1921-2001) y Geoffrey Tootill (1922-2017)	Se utilizan por primera vez los tubos de rayos catódicos de F.C. Williams como memoria	Trabaja con longitud de palabra de 32 bits y una memoria de 32 palabras. El primer programa tenía 17 instrucciones, escrito por Tom Kilburn, se ejecutó el 21 de junio de 1948. Primer programa informático en un ordenador electrónico digital con memoria
1949	MADDIDA (<i>Magnetic Drum Digital Differential Analyzer</i>)	Floyd Georg Steel (1917-1995) para <i>Northrop Aircraft</i>	Prototipo completamente electrónico. Representa bits por niveles de voltaje	John von Neumann destacó que es la primera máquina en utilizar únicamente álgebra booleana

Cont...

Año	Denominación	Inventor	Desarrollo	Operaciones
1949	BINAC (<i>Binary Automatic Computer</i>)	John W. Mauchly (1907-1980) y John Presper Eckert (1919-1995)	Fabricado para <i>Northrop Aircraft</i>	Dispone de cintas magnéticas para almacenar información
1949	EDVAC (<i>Electronic Discrete Variable Automatic Calculator</i>), sistema binario	John W. Mauchly (1907-1980), John Presper Eckert (1919-1995) y John von Neumann (1903-1957)	Laboratorio de Investigación en Balística de EE.UU.	Lector-grabador de cinta magnética, unidad de control con osciloscopio, unidad para recibir instrucciones del control y la memoria, y para dirigirlas a otras unidades, unidad computacional para realizar operaciones aritméticas en un par de números a la vez (y mandarlos a la memoria después de corroborarlo con otra unidad idéntica), cronómetro, y unidad de memoria dual
1949	CSIRAC (Australia)	Trevor Pearcey (1919-1998) y Maston Beard (1939-)	Council for Scientific and Industrial Research Automatic (CSIRAC)	Ordenador con tubos de vacío y memoria de retardo de mercurio. Primer ordenador en reproducir música
1951	UNIVAC I	John W. Mauchly (1907-1980) y John Presper Eckert (1919-1995)	Los inventores venden su compañía a Remington Rand	Primer computador con fines comerciales

CONCLUSIÓN

La gran destreza de los relojeros permitió que Pascal y Leibniz diseñaran máquinas de calcular. Esta habilidad en la construcción de precisos engranajes permitió la invención de sofisticados ingenios mecánicos para resolver cálculos matemáticos.

La riqueza material surgida del maquinismo ha tenido su aplicación en el mundo de la construcción y en el de los ordenadores. En el siglo XXI las máquinas se van miniaturizando y cada vez resulta más evidente la importancia de la programación de estos nuevos ingenios. Programación donde las mu-

jeros han tenido un papel destacadísimo que aún refuerza más su invisibilidad en el desarrollo de las máquinas de cálculo.

REFERENCIAS

González de Posada F. *Leonardo Torres Quevedo: el más prodigioso inventor de su tiempo*. Madrid: Real Academia Nacional de Medicina, 2007.

Lusa G., Roca A. *La balança algebraica (1908) i el ternal algebric (1932). El càlcul mecànic de Paulí Castells i Vidal (1877-1956)*. Barcelona: Patrimoni ETSEIB, 2018.

Williams M.R. *Differential analyzers*. IEE Stars program 2013.

ANEXO II

LA RIQUEZA INMATERIAL

Tras los primeros ordenadores, tales como ENIAC, CSIRAC, EVAC y especialmente del UNIVAC fabricado por Remington Rand, aparece en este mercado IBM. Surgida en los años veinte del siglo XX era el resultado de la fusión entre empresas relojeras y el inventor de la máquina de censos de Estados Unidos, Herman Hollerith (ya mencionado en la tabla A.I).

A finales de los años cincuenta, IBM inicia sus relaciones con el MIT, el Massachusetts Institute of Technology, para desarrollar un ordenador. Pero lo importante no es el retraso de IBM frente a Remington Rand, ni su fulgurante ascenso en el mercado comercial de ordenadores; la aportación más destacada de IBM es que en 1957 lanzan un lenguaje de programación de alto nivel llamado Fortran (*FORMula TRANslation*) generado por el equipo de John W. Backus (1924-2007). Fortran permitía escribir programas independientes del constructor de la máquina, pero funcionaba con la velocidad del propio lenguaje del ensamblador. La comunidad científica se lanzó a utilizarlo de forma espectacular.

La comunidad empresarial observando el éxito de Fortran demandó un programa más simplificado que permitiese una programación aplicada a los negocios. Apareció en 1960 un nuevo lenguaje llamado COBOL (*COmmon Business-Oriented Language*). Su generación parte de una cooperación entre los fabricantes de ordenadores, los usuarios y el Departamento de Defensa de Estados Unidos, llamada CODASYL (1959), quienes en seis meses lanzaron este lenguaje con una sintaxis parecida al inglés y sin símbolos extraños que hasta entonces utilizaban los programadores.

Paralelamente un gran visionario del mundo de la informática John McCarthy (1927-2011), a quien se le atribuye el concepto de inteligencia artificial, ofrecía un nuevo lenguaje de programación denominado LISP (*LISt Processor*). Steve Russell (1937) aplicó LISP a un ordenador IBM 704 y descubrió para sorpresa de McCarthy que algunas funciones podían implantarse directamente en el código máquina. De ahí que en LISP tanto los datos como los programas se reducen a listas.

Sobre estos tres grandes lenguajes puede explicarse la genealogía de los restantes, algunos son dialectos, otros se crean sobre sus raíces. Pero falta una pieza para poder explicar todo el desarrollo de los lenguajes de programación. Se trata de ALGOL (*ALGOriithmic Language*). En 1958 se reúnen en el Instituto Federal Suizo de Tecnología un comité de expertos europeos y americanos para desarrollar un «*International Algebraic Language*» que se denominó finalmente Algol58.

Este es el primer intento para crear un estándar de lenguaje de programación que fuese aceptado por los distintos fabricantes de ordenadores. El resultado más fructífero de este lenguaje madre ha sido el conocido lenguaje C, que ha tenido múltiples versiones.

En la figura A.1 se muestra un gráfico muy simplificado citado por Rigaux.org en el que puede observarse la genealogía derivada de estas cuatro iniciativas.

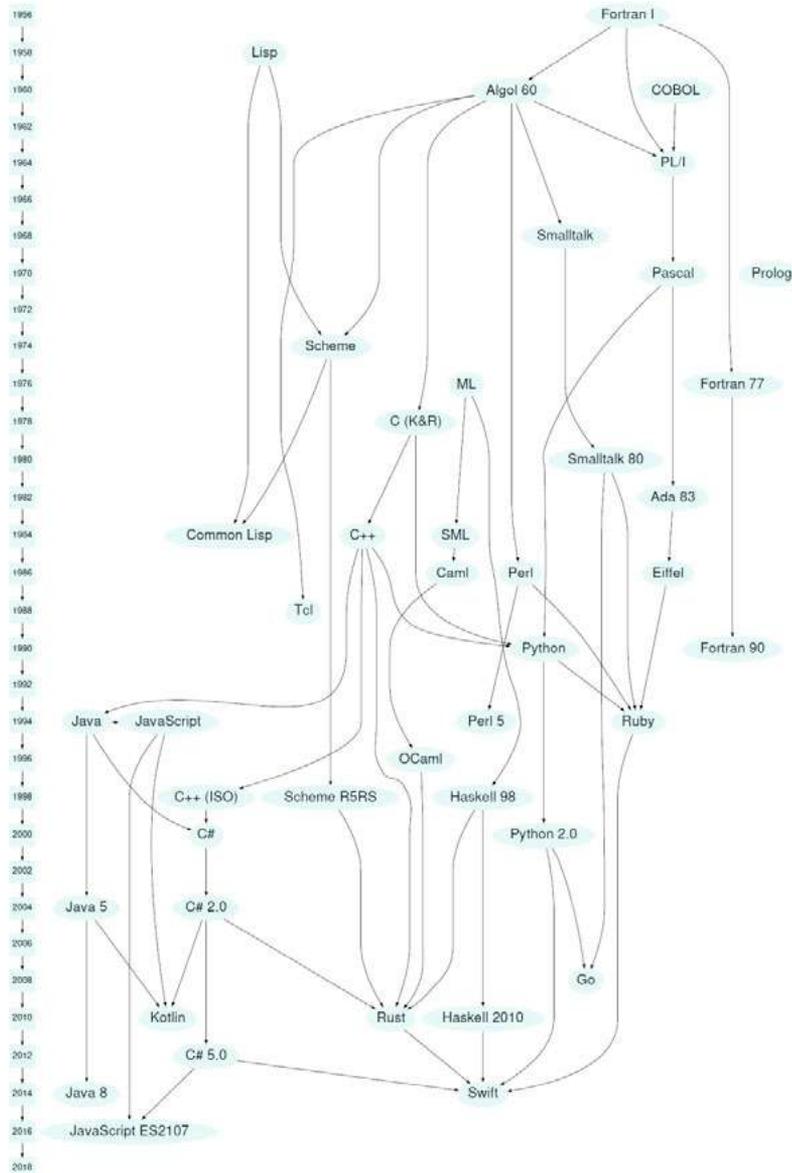


Figura A.1. Gráfico resumido sobre genealogía de lenguajes de programación.

De acuerdo con estos datos, en la actualidad hay una convergencia hacia otros tres lenguajes: Javascript (muy influenciado por C), Swift (acogido por los desarrolladores de aplicaciones de Apple) y Go (para los desarrolladores de aplicaciones Google).

Fortran ha sido actualizado, pero la comunidad académica ha ido desarrollando nuevas iniciativas. Una datación de los principales lenguajes de programación que más se han extendido entre los científicos se podría presentar, sin ánimo de exhaustividad, como refleja la tabla A.II.

Tabla A.II. Cronología del desarrollo de los lenguajes de programación

Lenguaje	Año	Desarrollador
FORTTRAN	1957	John W. Backus
LISP	1958	John McCarthy
ALGOL	1958	Instituto Federal Suizo de Tecnología
COBOL	1959	CODASYL (<i>Committee on Data Systems Languages</i>) y Departamento de Defensa de EE.UU.
Simula	1960	Dahl-Nygaard
BASIC	1964	John George Kemeny y Thomas Eugene Kurtz
PASCAL	1970	Niklaus Wirth
Scheme	1975	Guy L. Steele y Gerald Jay Sussman
S	1976	Rick Becker, Allan Wilks y John Chambers
Objective-C	1980	Brad Cox-1992, código libre por Steve Jobs
Maple	1982	Gastón Gonnet y Keith Geddes
C++	1983	Bjarne Stroustrup
Matlab	1984	Cleve Moler y Jack Little
Mathematica	1988	Steve Wolfram
Java	1990	James Gosling
Python	1991	Guido van Rossum
R	1993	Ross Ihaka y Robert Gentleman
Windows NT	1993	Microsoft Corp.
Sage	2005	William A. Stein
Julia	2012	Stephan Karpinsky y Alan Edelman

Sin embargo, debe tenerse en cuenta que los entornos de programación son mucho más amplios y el trabajo inmaterial de los programadores puede desarrollarse en distintos entornos de programación (tabla A.III).

Tabla A.III. Entornos de programación

Programación sobre	Desarrollo	Productos comerciales
Entornos extremos	1940-actualidad	Supercomputadores, ordenadores cuánticos, biocomputadores
Sistemas operativos	1970-2000	DR-DOS, MS-DOS, Apple II, UNIX
Web	2000-actualidad	JAVA, Netscape, Explorer, Chrome
Aplicaciones móviles	2010-actualidad	IOS, Nokia, Samsung, Android
Nube	2015-actualidad	Amazon Web Services, OneDrive, Google Drive, Facebook

CONCLUSIÓN

La riqueza inmaterial que se genera con la programación informática va a ir creciendo a medida que más y más entornos humanos quieran evitar actividades repetitivas y deseen concebir nuevas soluciones originales a los problemas más cotidianos.

REFERENCIAS

Para observar gráficos sobre la genealogía de los programas informáticos:
<http://rigaux.org/language-study/diagram.html>

Para encontrar más de 50 categorías en las que clasificar a los lenguajes de programación: https://en.wikipedia.org/wiki/Generational_list_of_programming_languages

Para una cronología exhaustiva de los lenguajes de programación:
https://en.wikipedia.org/wiki/Timeline_of_programming_languages

ANEXO III

EL PRIMER ORDENADOR DE BARCELONA

El 31 de mayo de 1967 ingresaba en la Real Academia de Ciencias y Artes de Barcelona el ingeniero Enrique Freixa Pedrals pronunciando un discurso titulado *Cálculo digital*.



El discurso es un homenaje al Dr. Paulino Castells Vidal, el especialista más importante de este país hasta la Guerra Civil en cálculo analógico, inventor de la balanza algebraica para resolver ecuaciones de una incógnita en 1908.

Balanza algebraica de Paulino Castells. Imagen cortesía de Adolfo G. Yagüe en www.ccapitalia.net.

La máquina de Paulino Castells es un ejemplo de la inmensa capacidad de trabajo de los intelectuales en la España de la época. Seguramente el trabajo más conocido en este ámbito es el de Leonardo Torres Quevedo quien desarrolló dos importantes máquinas de calcular y sorprendió a todo el mundo cuando en 1912 presenta en París su «ajedrecista», primera combinación de mecánica y electromagnetismo para resolver problemas matematizables.

Castells, quien había coincidido con Torres Quevedo en Madrid, los dos eran esperantistas, continuó sus trabajos. Pasó de la balanza al polipasto para resolver sistemas de tres ecuaciones con tres incógnitas mediante un sistema distinto del de Cramer. Este nuevo ingenio mecánico descrito en 1932, reconocido internacionalmente y patentado en 1938 como «polipasto algébrico» fue el inicio de una nueva investigación, el algébrico eléctrico.

Guillermo Lusa y Antoni Roca han estudiado todas estas piezas y las explicaciones que el propio Castells publicó. Especialmente destacable es su conferencia de 1945 «Aportación al cálculo matemático» homenajeada por Enrique Freixa en su discurso de ingreso de 1967.

La historia que hay detrás no deja de ser sorprendente. El 17 de marzo de 1938 las legiones aéreas de la Italia fascista bombardearon Barcelona desde su base en Mallorca. En un alarde de precisión lograron lanzar una bomba que cayó sobre un camión cargado de trilita en el centro de Barcelona. La explosión mató a muchos niños refugiados en un edificio de la Gran Vía.



Polipasto algébrico de Paulino Castells.
Fuente: Escuela Técnica Superior de Ingenieros Industriales de Barcelona.



Bombardeo sobre Barcelona, 17 de marzo de 1938. Fuente: Fondo Brangulí, Archivo Nacional de Cataluña.



Destrucción del edificio de Gran Vía al lado del cine Coliseum. Fuente: Wikipedia

Pero este bombardeo también afectó a la Escuela de Ingenieros Industriales de Barcelona. En la cuarta planta se encontraba el laboratorio del catedrático Paulino Castells. En esos momentos estaba desarrollando una nueva máquina de cálculo con la que se conseguía gran precisión al sustituir los engranajes por pulsos electromagnéticos.

Mientras las tropas franquistas se habían dotado de máquinas de criptografía *Enigma*, provenientes de la Alemania nazi, las tropas italianas destruyeron el invento que hubiese ayudado a descifrar esos códigos y alejaron a este país de los avances en el mundo de los ordenadores. Castells marchó a Colombia y cuando regresó a Barcelona ya no pudo completar por falta de medios los avances

que había previsto. Sin embargo, en la Escuela de Ingenieros Industriales de Barcelona se avanzó algo más. Hoy puede visitarse en el Museo Nacional de Ciencia y Tecnología de Cataluña, en Terrassa, el último vestigio de todas esas investigaciones, la calculadora Planell.



Calculadora Planell. Imagen cortesía del Museo de Ciencia y Tecnología de Cataluña, Terrassa.

Esta calculadora totalmente eléctrica está diseñada por Francisco Planell Riera (1886-1973). En un estudio realizado por Jordi Gabernet Vidal se describen las relaciones de esta calculadora con el algébrico eléctrico de Paulino Castells.

Castells ingresó en la Real Academia de Ciencias y Artes de Barcelona en 1913 con un discurso sobre «Las representaciones mecánicas de los impulsos eléctricos». Planell también ingreso en la misma Academia con un discurso sobre «Las máquinas eléctricas a velocidad variable» en 1922.

Tras la Guerra Civil los dos se encontraron en Barcelona formando a los nuevos ingenieros industriales y allí se construyó este prototipo de calculadora eléctrica que es el primer ordenador que tuvo Barcelona. Una calculadora analógica de fundamento electromagnético capaz de resolver sistemas de cuatro ecuaciones lineales con cuatro incógnitas, donde ya está separada la unidad de cálculo de la unidad de entrada y salida de datos.

Mientras se van desarrollando estas máquinas nace una figura disruptiva, Alan Turing, quien aplicando estos conocimientos a la criptografía consiguió hacer funcionar su máquina *Enigma*, para descifrar las comunicaciones de las fuerzas nazis en la Segunda Guerra Mundial.

Tras la contienda, los ingenios mecánicos fueron incorporando más y más electrónica. Los avances en el cálculo matemático fueron saltos cualitativos inimaginables unos pocos años antes. La eliminación de engranajes y bielas sustituidos por tubos de vacío, lámparas y reóstatos abrió la puerta a los transistores y los chips.

De estos trabajos surge la máquina ENIAC, construida entre 1943 y 1946 en la Universidad de Pennsylvania, considerada el primer ordenador porque supera las máquinas de calcular gracias a la introducción de la electrónica.

Pese a que las técnicas nuevas «en materia de memoria, la electrostática y la de líneas de mercurio» fueron ensayadas en las universidades británicas de Manchester y Cambridge, en 1954 en Estados Unidos aparecen los primeros ordenadores fabricados de forma industrial. En 1954 IBM construyó 14 máquinas contables a base de fichas perforadas modelo 702.

Pero simultáneamente a los inventos materiales, se iban generando unos inventos inmateriales que hoy son indispensables para nuestro día a día. La programación informática que había descrito Ada Lovelace (hija de Lord Byron), en 1841 precisó casi un siglo para que otras mujeres, las programadoras del ENIAC, estructurasen lo que hoy conocemos como lenguajes informáticos.

Esas mujeres que llamaron «palabras» a las combinaciones de bits, fueron las que abrieron el camino a muchos científicos y técnicos que han ofrecido a la humanidad una riqueza inimaginable. Los lenguajes de programación han ido cambiando nuestras sociedades, han cambiado los planes de estudio y aún han de cambiar más cosas, de las más cotidianas a las más excepcionales.

Enrique Freixa en 1967 expone cómo en estos ordenadores «se funde la triple corriente científica, técnica y económica» y augurando la enorme transcendencia de este hecho propone la nacionalización del «cálculo digital». Con todos los datos reunidos en ese año, viendo el «galopar cada vez más vertiginoso del progreso» es capaz de proponer algo que estamos viviendo «si una nación o aun un grupo de naciones no es capaz de sostener tal ritmo de investigación, debería en tal caso ser revisada la misma noción de independencia política».

En esos años ya se había puesto en funcionamiento la alianza supranacional ALGOL: desde Suiza se reunió a los europeos y norteamericanos para acordar las estructuras de un lenguaje de programación común. Es la época de los POL, los «*Program Oriented Language*», cuyos casos más exitosos fueron Fortran y COBOL. Se propuso llegar a unos lenguajes UNCOL, de «*UNiversal Computer Oriented Languages*», pero no se completó esta andadura.

CONCLUSIÓN

La Guerra Civil española puso fin a los avances que se habían conseguido en la construcción de ingenios mecánicos para calcular y que estaban siendo modificados para incorporar las innovaciones electromagnéticas del momento.

En Barcelona se estuvo muy cerca de poder presentar una calculadora que hubiese situado a nuestro país al nivel de los países más avanzados del momento. Pero lo más grave de este parón en seco de las investigaciones en

informática fue que no se desarrolló ningún equipo de programación. Los lenguajes informáticos se concentraron en las mentes de los científicos anglosajones.

En el discurso de ingreso de Enrique Freixa a la Real Academia de Ciencias y Artes de Barcelona destaca que deberían aparecer los lenguajes UNCOL, un cambio de paradigma desde los lenguajes orientados al problema hacia los lenguajes orientados al objeto, un gran paso en esa línea, y sugiere que aún vendrán otros cambios de paradigma que universalizarán aún más los lenguajes.

REFERENCIAS

Freixa E. *Cálculo digital*. Barcelona: Real Academia de Ciencias y Artes de Barcelona, 1967.

Gabernet J. *La calculadora Planell*. Terrassa: Museu Nacional de la Ciència i de la Tècnica de Catalunya, 2013.

Lusa G. *La escuela de ingenieros en guerra (1936-1938)*. Barcelona: Documentos de la Escuela de Ingenieros Industriales de Barcelona núm. 17, 2007.

