

LA COMPUTACIÓN EN EL INSTITUTO DE CIENCIAS DE LA UAP

A. Licona, S. Angoa, J. L. Victoria, R. Bautista y M. A. Burgos

Agradecemos al Dr. Harold V. McIntosh sus valiosos comentarios.

1985

El presente documento es la recolección de los artículos:

- Una filosofía para el trabajo, 30/dic/1984.
- La evolución de la computación, 06/ene/1985.
- La historia de las microcomputadoras, 20/ene/1985.
- Historia del software I, 03/feb/1985.
- Historia del software II, 10/feb/1985.
- El cálculo numérico y el lenguaje Fortran, 03/mar/1985.
- Computadoras y procesos numéricos, 31/mar/1985.

que aparecieron en las fechas señaladas en el periódico Nueva Era en una serie titulada:

“LA COMPUTACIÓN EN EL INSTITUTO DE CIENCIAS DE LA UAP”.

Aquí lo presentamos en forma electrónica.

Las imágenes que aparecen a partir de la página 46 proceden de: “The Graphic Work of M. C. Escher” (1975).

Puebla, Pue., diciembre de 1998.

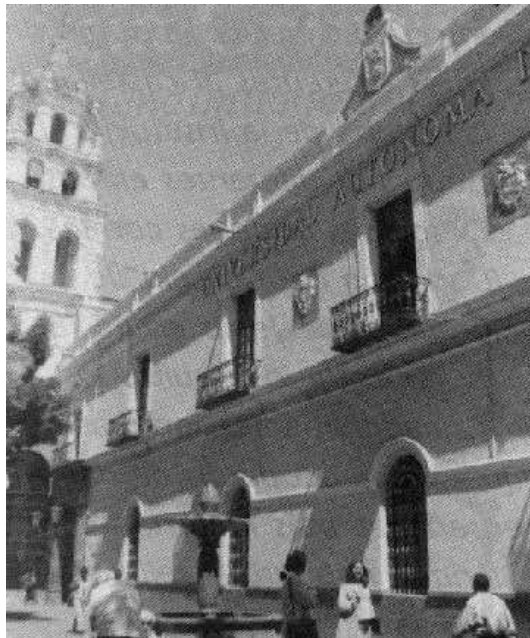
UNA FILOSOFÍA PARA EL TRABAJO

El propósito de la Universidad, consiste en la formación de ciudadanos capaces de ejercer una práctica profesional y valiosa para la sociedad, crítica de sus aspectos negativos y ejemplar en cuanto al tratamiento racional de sus problemas.

Sus funciones sustantivas son la investigación, en la que está presente el rigor más elevado de la conducta racional; la docencia, en la que dicha conducta debe imbuirse con el ejemplo y la información adecuada y la extensión, que es o debe ser su medio de servicio y conocimiento del entorno social.

También la Universidad realiza funciones en apoyo de las anteriores como son: la promoción, la difusión y la administración. Esta última tiene como propósito esencial cuidar de la armonía en el uso y desempeño de los elementos que conforma el patrimonio universitario y que consiste en: el acervo informativo, la planta física, los recursos financieros y -lo más importante- los recursos humanos.

En todas las actividades que se desarrollan en el ambiente universitario, se observa un denominador común, que consiste de dos elementos fundamentales: la información y el razonamiento.



Gaceta UAP, mayo 1998, foto: Rufino Vázquez P.

1.1 LA INFORMACIÓN

La información es objeto de intercambio constante en las actividades docentes. En la investigación, la información también juega un papel importante en el desarrollo de las actividades de administración, la promoción y la difusión, en algunos de estos casos, el papel de la información es determinante.

1.2 EL RAZONAMIENTO

La información no sólo es objeto de intercambio en la actividad universitaria. Tampoco la información solamente se intercambia sino que se transforma, organiza y elabora mediante conductas y procedimientos racionales. El razonamiento, la conducta intelectual, es la parte medular en la investigación científica.

Este comportamiento racional también está presente, aunque no con la intensidad deseable en la docencia, su presencia se insinúa mediante la exposición de esquemas axiomáticos elaborados como lo son la geometría, el álgebra, el cálculo y la física, que ponen en relieve los métodos y procedimientos del razonamiento científico, es decir los procedimientos de análisis y síntesis.

Esta conducta racional consiste en una disciplina de trabajo ordenada, organizada y sistemática que permite identificar los problemas, caracterizarlos, conmensurarlos, diseñar las opciones de solución y tomar las decisiones más pertinentes. Esta conducta racional es por antonomasia, la formación científico universitaria.

En esencia, el razonamiento ordenado está presente desde los niveles de excelencia de la investigación científica, hasta la decisión más sencilla de la vida diaria, tal como una compra o una visita amistosa; sin embargo, la distancia que separa a los profanos de los científicos, en buena medida se debe al uso de notaciones y lenguajes por parte de estos, que resultan incomprensibles para aquellos. Lo anterior muestra que un primer problema estriba en la comprensión de notaciones y lenguajes.

Por otro lado, el razonamiento lógico en un nivel particular, puede resultar práctica irracional, cuando se evalúa desde otros niveles; esto ocurre tanto al profano como al científico. Así, en la vida cotidiana, el arrojar basura en un lote baldío, nos “resuelve” el problema en casa, pero genera otro problema de orden colectivo; tomar una calle en sentido contrario a la circulación normal o estacionarse en doble fila, nos ahorra tiempo, pero arruina el de los demás, e inclusive puede costarnos la vida. En ese mismo sentido, los científicos que no se ubican en el entorno en que actúan, pueden estar trabajando muy racionalmente en el nivel de su disciplina, pero irracionalmente en otros niveles sociales; aún más, los resultados de su actividad científica pueden ser irracionales para los intereses de la sociedad.

El razonamiento tiene pues más de una dimensión. Saber las reglas del ajedrez, no significa por sí mismo, saber jugar ajedrez.

La responsabilidad ciudadana es grande para todos, pero es doblemente importante para el universitario que debe atender a la múltiple racionalidad de su quehacer científico y dar ejemplo de respeto a las más elementales normas de convivencia social, sin dejar de cuestionar aquellas que carecen de sentido práctico o resultan grotescas por su nulo valor social. Ciencia y convivencia se condicionan mutuamente.



Portada de Universidad Gaceta, año VIII, No. 4, abril de 1997.

1.3 INFORMACIÓN Y RAZONAMIENTO

La información y el razonamiento, son elementos que deben coincidir en nuestras actividades, ya que la falta de aquella las reduciría a la calidad de ejercicios de tipo sofista y a la ausencia de éste, a ejercicios de carácter enciclopédico en los que la información se recolecta, se acumulan, ordenan sin mayor elaboración y acaba por agobiarnos. Entre estos dos extremos indeseables existen puntos de equilibrio en los que se cuenta con información y también con conductas racionales que permiten obtener nuevos datos a partir de los originales, identificar similitudes y conformar patrones inteligentes sobre la naturaleza de los objetos que se estudian. La información y el razonamiento no se excluyen, se complementan.

1.4 LA NATURALEZA ENTRÓPICA DE LA INFORMACIÓN

La información posee una característica que le es muy peculiar, su cantidad siempre está en aumento y su calidad tiende a la uniformidad. Si en los periódicos del año anterior suprimimos las fechas, difícilmente podremos ubicar cronológicamente las noticias que contienen. Esta característica de la información se define como comportamiento entrópico. La información es un ingrediente que por esta naturaleza tiende al desorden, condición que secularmente ha tenido que reducirse al orden por medio de procedimientos racionales. Por la naturaleza entrópica de la información, su elaboración es mucho más que un simple proceso de acumulación.

1.5 LA CANTIDAD NOS AGOBIA

Las formas en que se ha manejado la información a través de la historia, configuran maneras diferentes de vida y relaciones diferentes entre el conocimiento y la práctica. Cuando la memoria individual era el principal almacén de la información, ésta se transmitía en forma oral pero principalmente, a través del ejemplo. Una forma objetiva para la transmisión de la información trascendental y genérica, se produjo con el uso de la inscripción y de los diferentes elementos estructurales en las construcciones monumentales de la mayoría de las culturas antiguas.

El desarrollo de la imprenta y la producción masiva de libros, desplazaron a esas antiguas formas de transmisión de información, pero no sólo eso, también sepultaron en el olvido muchos de los conocimientos que en la actualidad llamaríamos ecológicos.

En nuestros días el uso de las computadoras nos coloca en el umbral de una nueva época en la que puede ser posible el manejo útil y racional de considerables volúmenes de información. El mundo del libro separó el ejemplo vivo de la enseñanza escolar, aceleró el aumento de la cantidad de información disponible, pero la inmovilizó en bibliotecas y archivos y produjo la especialización de la actividad intelectual en disciplinas científicas de lo particular.

La computadora es un accesorio que nos permite volver a poner en movimiento ese acervo de información sin necesidad de agobiar nuestra memoria. El empleo de la computadora y el desarrollo de la programación tanto básica como de uso, debe ser común para cualquier persona como el empleo de las calculadoras electrónicas o el teléfono, con mayor razón, para los universitarios.

Con este apoyo, será posible devolver el sentido original a la tarea del pedagogo, que consiste en enseñar actuando, es decir, enseñar a investigar, investigando.



Centro de Cálculo de la UAP.

1.6 INVESTIGACIÓN REGIONALISTA O UNIVERSAL

En el mundo actual hay regiones que se ubican en la cúspide del desarrollo científico, tecnológico o cultural; otros se debaten en los niveles más bajos de pobreza intelectual y en consecuencia, también en lo material. No obstante todos participan del consumo de mercancías de alta tecnología. En un hogar mexicano, común y corriente, el televisor es considerado artículo de primera necesidad.

Nuestro país está implicado en una red mundial de consumo, pero no participa significativamente en la construcción, elaboración y diseño de esos bienes. Sin esta participación en la producción, nuestro futuro no sólo es incierto, sino aterrador.

La sociedad que consume los bienes de referencia, implícitamente ha elegido participar en el mundo contemporáneo, aceptando y haciendo indispensables estos factores y servicios.

Los elementos de nuestra sociedad no están dispuestos a privarse del uso del teléfono, de los automóviles y de los servicios de agua potable y energía eléctrica. Tampoco renuncian al uso de aviones, satélites, computadoras sin excluir el estereo, la lavadora y al televisor.

Existe diferencia entre aceptar las relaciones del mundo contemporáneo y participar en ellas. La aceptación de los resultados se traduce en una dependencia tecnológica y cultural permanente, y la participación, en incipientes primeros y después vigorosos desarrollos de la ciencia y la tecnología.

La sociedad contemporánea es en extremo complicada, hay elaboraciones científicas que requieren de la participación de todo el mundo para poderse realizar. Los aviones, los satélites, la industria química y de las comunicaciones así como las computadoras

entre otros logros culturales contemporáneos no son resultado del esfuerzo de un equipo de trabajo ni de equipos de trabajo de un solo país, ni de una disciplina científica, son producto de la participación colectiva de múltiples equipos de varios países, que emplean resultados de todas las disciplinas y capitalizan el conocimiento de todas las épocas.

El procesamiento de la información por medios electrónicos se puede ubicar en la cúspide de un gigantesco árbol. Las ramas y los nudos están conformados por complicadas elaboraciones científicas, tecnológicas y culturales, que incluye: administración, programas de control, equipo, suministros, servicios, etc.

En la medida que se desciende en el árbol, aparecen diferentes instancias con similares niveles de elaboración, por ejemplo una computadora está formada por diferentes unidades funcionales: unidad central, memoria, equipo periférico, incluidas impresoras, unidades de disco, graficadores, unidades de cinta magnética, etc.

Cada una de estas unidades funcionales está por su parte constituida por módulos y accesorios más simples como son: circuitos de control, motores, servo-mecanismos, fuentes de poder, ventiladores, filtros, chasis, gabinete, etc. Algunos de estos módulos, están contruidos a partir de circuitos impresos, circuitos integrados (semiconductores), elementos discretos, conectores, cables y otros.

Los recursos necesarios para la elaboración de elementos semiconductores son entre otros, los que se enlistan a continuación: encapsuladores, cuarto limpio, microscopio electrónico, soldadores de tecnología laser, materias químicas con alto índice de pureza, materiales químicos fotográficos, fuentes de poder, ventiladores, extractores, motores, con la participación de las industrias: metal-mecánica, petroquímica, metalúrgica, fotografía etc. No se mencionan todos, pero si nos atenemos a la descripción que presenta la matriz de insumo producto de los países que diseñan y elaboran elementos semiconductores, este rubro de la economía depende de insumos provenientes de la totalidad de los rubros económicos.

La elaboración de los elementos que aparecen en cada nodo del árbol, en sus diferentes instancias, requieren por su parte de otros recursos científicos, tecnológicos y culturales. Por ejemplo, la elaboración de materiales con alto índice de pureza, que requiere la industria de los semiconductores, sólo es posible a partir de los resultados obtenidos por la física del estado sólido, quien por su parte, requiere de los resultados que aportan las ciencias básicas como son la física, la química y las matemáticas, entre otras disciplinas.

La descripción se puede continuar hasta agotar la paciencia del lector.

Algunos de estos insumos, se producen en la región o en el país, pero hay muchos que aún cuando se podrían producir, no se producen porque no existe un mercado que los demande y algunos otros, requieren técnicas, procedimientos, instrumentales y equipos tan sofisticados, que sería necesario invertir varios años de trabajo para su desarrollo.

Como se puede apreciar, la transición desde una colección de nodos inferiores a otro de orden superior, exige un mínimo de condiciones ya establecidas y la realización de los trabajos encaminados a producir la transición de referencia. Por ejemplo, a partir de circuitos impresos, integrados, componentes discretos, conectores, se puede diseñar y elaborar un circuito de control para un equipo periférico. Con el circuito de control antes mencionado, motores, gabinetes, fuentes de poder, armazones, filtros, materiales magnéticos, es posible elaborar la unidad de disco para una computadora.

Desafortunadamente de todas las posibles transferencias entre los nodos del árbol, sólo algunas de ellas son realizables en nuestro contexto social. Algunas de ellas requieren desarrollos extraordinariamente elaborados que sólo pueden realizarse al cabo de varios años y con el concurso de muchas entidades.

La descripción mnemotécnica del procesamiento electrónico de datos en forma de una estructura ramificada, que ubica al objeto de la discusión en la rama más elevada, es aplicable a cualquier otra elaboración contemporánea de cualquier nivel de complejidad.

Pero nuestra sociedad, con todos sus elementos de regionalidad y provincialidad requiere de estos equipos para elaborar su información, también en algunos casos, los exige aunque no le sean indispensables. Es muy riesgoso aceptarlos en la medida que la sociedad contemporánea los ofrece para consumo sin participar en la elaboración. Es necesario, imperativo, participar en ese desarrollo, pero participar de una manera determinante, no sólo con la intención de atraer la atención de nuestro entorno inmediato, sino también, la perspectiva de incidir en ámbitos más amplios.

La participación de nuestra Universidad será manifiesta en la sociedad contemporánea cuando las transiciones que realiza a partir de algunos de los nodos del árbol para alcanzar otro de ellos, interesen a entidades en otros lugares del mundo. Y no sólo llame su atención por los elementos de originalidad e innovación que incorporen, sino que incida en su conducta.

Cuando las experiencias obtenidas al propiciar una transición en el árbol interesen en otros lugares del mundo, en ese momento, dejaremos de ser una Universidad casera para convertirnos en una Universidad contemporánea.

Resulta absurdo pensar que nuestra participación en el desarrollo mundial de la ciencia, la tecnología y la cultura deba repetir el camino original de desarrollo que transitaron otras naciones hoy poderosas y modernas; nuestros equipos de investigación deben avanzar a saltos y por atajos, que su misma actividad debe señalar, esto exige talento e imaginación; por ello la Universidad sabe reivindicar como su función sustantiva a la formación de ciudadanos con capacidad de ejercer sus habilidades racionales, características de la investigación, en sus actividades diarias.

LA EVOLUCIÓN DE LA COMPUTACIÓN

Desde tiempos inmemorables, la humanidad ha desarrollado y usado por su conveniencia o por necesidad, accesorios que le facilitan y abrevian la práctica del cálculo. Ejemplos de estos, han sido los guijarros y los ábacos de civilizaciones antiguas.

Cabe mencionar las tablas de cálculo de amplio uso en la Edad Media y la regla de cálculo. La primera versión de la regla de cálculo aparece en el año de 1620, en Inglaterra a cargo de Edmundo Gunter y William Oughtred; es hasta el año de 1750 cuando Leadbetter le incorpora la regleta imprimiéndole el aspecto actual.

La utilización de accesorios mecánicos ocurre hasta el siglo XVII. Casi simultáneamente, Schickard (1624) y Blaise Pascal (1645), conciben y realizan mecanismos capaces de sumar y construidos con engranajes de 10 dientes, cada diente representaba a uno de los dígitos del 0 al 9. En ambos casos, el mecanismo incorporaba el manejo de los acarrees. En época reciente, se encontró entre los restos del naufragio de una embarcación griega un mecanismo que, en opinión de algunos especialistas, se construyó para facilitar operaciones elementales.

En el año de 1675 Gottfried Wilhelm von Leibnitz construye, a partir de los mismos principios y con un mayor perfeccionamiento, una máquina que además de realizar las cuatro operaciones elementales, también permitía el cálculo de raíces. Más tarde, al iniciarse el siglo XVIII, Falcón introduce en algunas maquinarias de la época el control automático con el empleo de rollos de papel perforado, recurso que se empleaba en los campanarios y las pianolas para la repetición de ciclos melódicos. Por su parte, el francés Joseph Marie Jacquard introduce en el año de 1811 el control automático en los telares con el uso de tarjetas perforadas.

El inglés Jesse Ramsden, constructor de equipo para astronomía y navegación, realiza en el año de 1777 una máquina para dividir. El francés Bolle construyó en el año de 1877 un mecanismo que era capaz de realizar multiplicaciones a partir de sumas y corrimiento; las máquinas anteriores reducían la multiplicación a series de sumas.

En 1811 Charles Babbage, un ingeniero inglés, concibe un dispositivo mecánico que denomina “máquina diferencial”, sin llegar a construirla. Su diseño aprovecha el principio de diferencias finitas para realizar los cálculos.

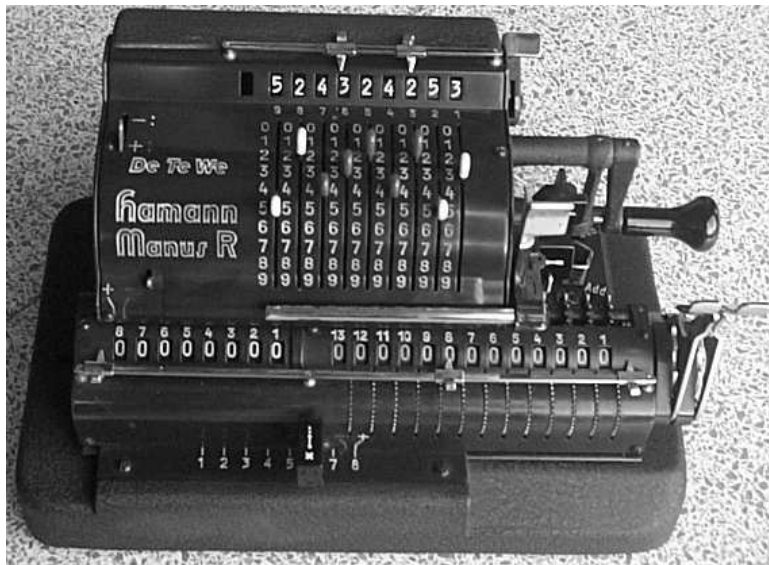
Posteriormente, en el año de 1822, somete a la consideración de la Royal Society, una nueva versión mejorada de su diseño, la “*máquina analítica*”, que se empieza a construir

en el año de 1840 sin llegar a concluirla, aunque el intento impulsó el desarrollo del instrumental mecánico.

En el diseño de la “máquina analítica” de Babbage, se perciben ya los cuatro elementos esenciales de las modernas computadoras: unidad aritmética, unidad de control, memoria y comunicación con el exterior. Inspirada en el trabajo de Jacquard, Lady Ada Lovelace, hija del poeta Lord George Gordon Byron, incorporó en el diseño de la máquina analítica de Babbage el concepto de programa de control. Para la elaboración y descripción del programa de control, desarrolló el concepto de diagrama de flujo.

En opinión de la mayoría de los especialistas contemporáneos, la máquina analítica de Babbage representa un diseño completo de una computadora que quizá por problemas de tipo financiero, por la falta de tecnología necesaria para realizarlo y también de un ambiente propicio, no se completó en su época. Es opinión unánime que lo ambicioso del proyecto de Babbage, sólo podía realizarse con el concurso de la tecnología electrónica y no con los elementos mecánicos disponibles en su época.

Algunos problemas que emergieron al final del siglo XIX requirieron la elaboración de considerables volúmenes de información. Uno de estos casos fue el censo de población de los Estados Unidos, el cual indujo a Herman Hollerith a perfeccionar la codificación de información en tarjetas perforadas.



Calculadora Mecánica.

Con el uso de esta técnica y el empleo de equipo electromecánico de conteo, elaboró los resultados del censo. Los equipos electromecánicos que utilizó Hollerith se pudieron realizar porque correspondían a diseños menos ambiciosos que la máquina analítica de Babbage, ya que sólo realizaban el conteo de la información.

El empleo de la tecnología electrónica tiene sus principios en el año de 1919, cuando Eccles y Jordan utilizan el doble tríodo para construir un circuito electrónico de conteo.

Con un enfoque diferente en el año de 1877, Lord William Thomson Kelvin, obtiene la solución de algunas ecuaciones diferenciales ordinarias con el empleo de dispositivos analógicos de integración.

Posteriormente, Vannevar Bush construye en el período 1925-1931, un calculador analógico con los mismos propósitos. Este enfoque diferente ha sido útil, pero limitado por la poca exactitud en los resultados y la complejidad del modelo. Lo anterior aunado al bajo costo de los circuitos digitales, ha orientado la preferencia en la actualidad hacia los modelos digitales.

A partir del año de 1930, Howard Aiken y George R. Stibitz inician el desarrollo de calculadores automáticos a partir de componentes mecánicos y eléctricos. Como resultado de su actividad, se producen 4 calculadores que se designaron con los nombres de MARK-1, MARK-2, MARK-3 y MARK-4, este último, construido en el año de 1945, incorporaba algunos componentes electrónicos (válvulas electrónicas), pero en su mayor parte estaba construido a partir de elementos eléctricos (relevadores) y mecánicos.

Estrictamente hablando, el término computadora, que se emplea para designar a los equipos de propósito general que son capaces de realizar cálculos arbitrarios, fue acuñado por el inglés Alan Mathinson Turing con su histórico trabajo “Computable Numbers”, publicado en 1937, en el cual desarrolla la teoría de las máquinas de Turing, establece la imposibilidad de resolver cierto tipo de problemas, entre otros, el problema de parar un proceso o completar un procedimiento.

En el fondo, la motivación de Turing para realizar el trabajo de referencia, se encuentra en el desarrollo de uno de los problemas planteados por David Hilbert en su habilitación como profesor, en 1905, referente a la revisión de la consistencia de los sistemas matemáticos. Este problema, entre otros, fue abordado por Bertrand Russell y Alfred N. Whitehead en sus Principia Mathematica. Turing expuso en términos de un mecanismo, que los sistemas matemáticos no pueden ser consistentes si se consideran como sistemas cerrados. Esto mismo había enunciado Alonzo Church y posteriormente por Kurt Gödel con el desarrollo de elaboraciones formales.

2.1 LA GUERRA Y EL DESARROLLO TECNOLÓGICO

Durante las hostilidades de la Segunda Guerra Mundial, Presper J. Eckert y John W. Mauchly desarrollaron en la Universidad de Pennsylvania un calculador electrónico, el ENIAC, que entre otras justificaciones tenía la de realizar el cálculo de las trayectorias de proyectiles y bombas.

Una de las primeras propuestas prácticas para desarrollar una computadora de propósito general, con el programa de control almacenado en la memoria, surge entre Eckert y Mauchly. El concepto fue desarrollado por John von Neumann, Goldstein y otros, en varios trabajos que se publicaron durante los años de 1945 y 1946, para el año de 1949, ya se encontraban en operación varios computadores, que incorporaban el concepto de programa de control almacenado en memoria, en varios laboratorios tanto de Inglaterra como de Estados Unidos.

Confluyeron en estos trabajos los intereses militares de al menos tres dependencias oficiales del gobierno de los Estados Unidos, la NSA (National Security Agency), la AEC (Atomic Energy Commission) y la SAC (Strategic Air Command). La primera interesada en el desarrollo de la criptografía; la segunda en el modelaje de las condiciones termodinámicas de la explosión de una bomba de hidrógeno y la tercera en problemas relacionados con el uso del radar para la vigilancia y control del espacio aéreo.

La empresa que en esa época poseía la planta para la atención de estos intereses, fue la posteriormente llamada gigante azul, la IBM, como consecuencia de una inversión astuta y afortunada, de T. J. Watson en el terreno de la computación electrónica.

2.2 INICIO DE LA COMERCIALIZACIÓN

El inicio de la comercialización ocurre en 1950 con la aparición de la UNIVAC I en el mercado. La construcción de esta computadora fue el resultado del concurso de varias compañías y de los diseñadores de la ENIAC, que al término de la guerra consumaron su intención de comercializar el producto de su trabajo.

En las décadas de los 60's y 70's, la industria de las computadoras recibió un fuerte apoyo con la realización de los programas espaciales de los Estados Unidos a través de la NASA. A la vez que la industria y los servicios acentuaron su demanda y con ello el proceso de comercialización.

Como resultado de la aparición de las computadoras comerciales, se desarrolló una amplia variedad en modelos y equipos. Tal diversidad hace imposible una reseña pormenorizada de cada uno de ellos. Para facilitar la descripción y comprender la evolución que la tecnología de las computadoras experimentó desde la década de los 50's hasta nuestros días, se recurre a una clasificación generacional.

Usualmente se emplean 4 niveles para describir los equipos. Una quinta generación se perfila con base en los más recientes desarrollos de la microelectrónica y las ciencias de la computación.

Los límites generacionales que se emplean son convencionales y se refieren a aspectos cronológicos, al tipo de tecnología que emplean en su fabricación, a las capacidades que las configuraciones típicas presentan, a la calidad y nivel de elaboración de la programación de apoyo y de uso que ofrecen. Los umbrales imprecisos.

2.2.1 La primera generación se ubica en el inicio de la década de los 50's.

Los equipos característicos en ella, son el UNIVAC I y la IBM 650. La capacidad de la memoria principal se reducía a algunos miles de localidades y era de tipo electrostático. Contaban con tambores magnéticos para almacenamiento y tenían ciclos de operación del orden 1,000 a 10,000 operaciones por segundo.

El equipo periférico se limitaba a lectoras y perforadoras de tarjetas y cinta de papel. Empleaban el lenguaje ensamblador como un recurso para depositar el programa en la memoria. El precio ascendía al millón de dólares.

2.2.2 La segunda generación se ubica al inicio de la década de los 60's.

Los equipos característicos son la IBM 704, la IBM 7090, CDC 3600. Utilizaban tubos de vacío miniaturizados y un incipiente empleo de la tecnología de transistor. La capacidad de las memorias se elevó a decenas de miles de localidades y eran de material magnético, el ciclo de operación se incremento un orden de magnitud con respecto a la anterior.

Hacen su aparición los primeros lenguajes de programación, tanto los orientados al cálculo numérico (FORTRAN), como los orientados a problemas de tipo lógico (no numéricos como LISP, COMIT y IPL V).

El equipo periférico se empieza diversificar. Hacen su aparición los primeros sistemas operativos rudimentarios. Los costos no se modificaron sustancialmente, con lo que se obtuvo mayor capacidad de procesamiento por el mismo precio.

2.2.3 La tercera generación se ubica al final de la década de los 60's.

Los equipos característicos son la IBM 360, Burroughs 6700, PDP-10, Univac 1100 y CYBER 170. Se afirma el empleo del circuito impreso y se inicia la utilización de memorias de semiconductor.

La capacidad de las memorias se elevó a varios cientos de miles de localidades, usualmente de tecnología semiconductor, el ciclo de operación se redujo en uno y hasta dos órdenes de magnitud.

Se inicia la diversificación de los lenguajes: COBOL, ALGOL, BASIC, PL/1 y otros entre los de tipo numérico, y SNOBOL, COMIT, PROLOG entre los no numéricos.

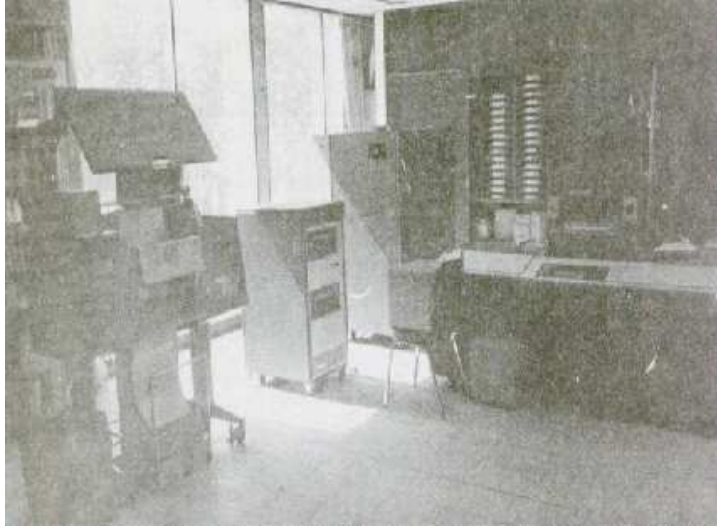
Se establecen los sistemas operativos y surgen las modalidades de operación en un tiempo compartido y multiprogramación.

Se incrementa notablemente la capacidad de los almacenamientos secundarios como consecuencia de los nuevos usos de las computadoras, aplicaciones con elevados volúmenes de información. Hacen su aparición las minicomputadoras y se inicia el uso de las telecomunicaciones.

Los costos experimentan desplazamientos en ambas direcciones, entre los diez mil y diez millones de dólares.

2.2.4 La cuarta generación se ubica al final de los 70's.

Los equipos representativos son los de las series 43 y 33 de IBM, Burroughs 7900, VAX 11/780 y CRAY. En esta generación no se percibe un cambio sustancial en el panorama, se registra un incremento similar en las capacidades y velocidades de los equipos generalmente en un orden de magnitud en los dos casos. Fundamentalmente las diferencias con relación a los equipos de la generación anterior se registran en la tecnología de construcción. Los costos de los equipos se mantienen.



Computadora IBM-1130, Centro de Cálculo UAP, tercera generación

Hacen su aparición los sistemas de bases de datos y se acentúa la participación de las telecomunicaciones.

También surge en este período la industria de las microcomputadoras.



Micromputadora Dual System UAP

Como ya se mencionó anteriormente, la clasificación generacional es un tanto subjetiva, aunque hay un consenso general que permite ubicarlas en función de ciertas características de los equipos. Sin lugar a dudas, la idea de la inminencia de la quinta generación, fue introducida en el ambiente por el Ministerio de Comercio Exterior del Japón (MITI).

2.3 LA LEY DE MOORE

En el desarrollo de las industria de las computadoras, se puede apreciar un comportamiento regular, en el que están involucrados la capacidad de proceso de los equipos y sus costos. Aunque identificada en forma empírica, su validez hasta la actualidad ha sido indiscutible. Este comportamiento se designa como Ley de Moore y se expresa en los siguientes términos: La capacidad de proceso de los equipos se duplica cada dos años y medio y los costos se mantienen. O en su forma equivalente, misma capacidad por la mitad de precio al término de 2 y medio años. La vigencia de la Ley de Moore se puede apreciar en los cuadros adjuntos.

2.4 Evolución de los componentes materiales de diferentes tecnologías.

Circuitos Integrados

<i>Generación</i>	<i>Año de Aparición</i>	<i>Nivel de Integración Transistores</i>	<i>Circuito Característico</i>	<i>Transistores por mm²</i>	<i>Costo Marginal</i>
1	1965	SSI	Compuerta flip flop	50	\$.35 Dls
2	1967	MSI	Contador Selector	500	.35 Dls
3	1973	SSI	Procesador 8 bits	10000	.35 Dls
4	1977	VLSI	Memorias Densas, procesador de 16 bits	50000	.35 Dls
5	1985	GSI		250000	.35 Dls

Discos Magnéticos

<i>Generación</i>	<i>Año de Aparición</i>	<i>Capacidad (millones de caracteres)</i>	<i>Costo por Carácter</i>
1	1964	7	\$.39 Dls
2	1968	30	.093 Dls
4	1973	200	.022 Dls
5	1977	317	.012 Dls
6	1981	500	.004 Dls

LA HISTORIA DE LAS MICROCOMPUTADORAS

3.1 LA PREHISTORIA

La prehistoria de las microcomputadoras nos ubica en los Bell Telephone Laboratories al final de la década de los 40's, cuando William Shockley, John Bardeen y Walter H. Brattain obtuvieron el primer elemento semiconductor de germanio. Sus esfuerzos se verían reconocidos con el Premio Nobel de Física, otorgado en el año de 1956.

La producción industrial de los semiconductores se inicia con la fundación de la Transistron Co. en la que participa David Bakalar. Al mismo tiempo, la Fairchild Camera Co. -productor de equipo fotográfico- establece su división de semiconductores con la participación de Shockley.

Ante la enorme perspectiva económica que ofrecía la producción de semiconductores, los gigantes del bulbo: RCA, Philco, General Electric (GE), Sylvania, Raytheon, Motorola y Westinghouse (W) entre otros; y de la industria de computadoras: IBM y Sperry Univac, se aprestan a transformar sus plantas de producción para participar en el negocio de los semiconductores. También la Texas Instruments (TI, instrumental geofísico) y la Unión Carbide (productos químicos) se incluyen.

En un principio la atención se concentra en la producción de diodos y transistores (series 1N y 2N) para aprovechar las características de éstos en la amplificación y de aquellos en la rectificación de señales. Los resultados de la industria del semiconductor no se hicieron esperar; al final de los 60's. Philco, IBM y Univac, anunciaban casi simultáneamente sus primeras computadoras comerciales completamente transistorizadas.

Ya anteriormente, RCA, Philco, GE y la W, habían puesto en el mercado sus electrodomésticos parcialmente transistorizados.

En poco tiempo, como consecuencia de la fragmentación de algunas de las empresas originales, el número se elevó a varias decenas. Aparecen así: Signetics, National Semiconductor, Internacional Silicon (INTERSIL), Mostek, Advanced Micro Devices (AMD) e Integrated Electronics (Intel), entre otras.

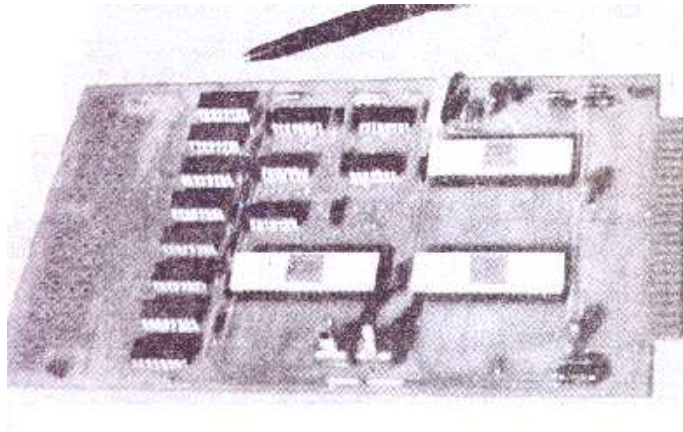
Con la diversificación de la industria, surgen también múltiples técnicas y procedimientos para la producción de elementos semiconductores. Este es el caso de la tecnología plana y el transistor de silicio, elaborado por National, que permitieron producir circuitos más estables, reducir los costos al propiciar la producción en gran escala e iniciar el proceso de integración de múltiples transistores en paquete.

Una vez iniciado este proceso de integración, la atención de la industria se orientó en la dirección de los circuitos digitales, en los que el transistor se utiliza como interruptor. Los progresivos avances que la industria experimentó en este aspecto parecían no tener límites.

La primera familia de circuitos digitales identificada con las siglas SSI (Small Scale Integration), surgió en el mercado en el año de 1965. Consistió principalmente de compuertas lógicas, flip flops y funciones lógicas básicas (and, or, not, nand, etc). Los circuitos de la serie 74 de TI son características de esta familia, que integró varias decenas de transistores por paquete.

Sólo tres años después estuvo disponible la segunda familia: la MSI (Medium Scale Integration), en la que ya se había conseguido conjuntar varios cientos de transistores en un solo paquete, dando lugar a circuitos con funciones mucho más complejas. Circuitos para conteo, codificación y selección así como memorias de baja y mediana capacidad, fueron comunes en esta familia.

La siguiente familia: la LSI (Large Scale Integration), apareció en 1972; contenía varios miles de transistores por circuito y los elementos más característicos consistían en memorias de alta capacidad y microprocesadores (unidades de control) de 4 y 8 bits. Un ejemplo típico fue el procesador 4004 que produjo Intel y que -si la ironía resulta verídica-, recibió esta designación por el número de transistores que requirió su diseño.



Micromcomputadora F8, construida por Fairchild con circuitos de la serie 38.

Una cuarta familia, la VLSI (Very Large Scale Integration) aparece en 1974, consiste fundamentalmente de microprocesadores de 8 bits y memorias de alta densidad. Los niveles de integración ascendían ya a los cientos de miles de transistores. Los circuitos típicos fueron el F8 de Fairchild, 8080 de Intel, 6800 de Motorola, 6502 de MOS Technology, COSMAC de RCA, SC/MP de National, Z80 de Zilog, controladores para discos y adaptadores para comunicación.

Actualmente están disponibles microprocesadores de 16 y 32 bits y procesadores aritméticos; seguramente no serán los últimos diseños.

3.2 PRODUCIR MI PROPIA COMPUTADORA (aunque no sirva)

Hacia el final de la época de los 60's, se perciben las primeras manifestaciones de la ruptura que ocurrió en el orden de la vida mundial. Los medios masivos de información centraron la atención en aspectos parciales como la protesta contra la guerra de Vietnam, el cuestionamiento a las tradiciones y valores sociales, el uso masivo de drogas entre los jóvenes, el rechazo al cumplimiento del servicio militar o a la violencia callejera. Sin embargo, no todos los aspectos de esta ruptura fueron sensacionales; hubo otros un tanto ignorados que apuntaron hacia profundos procesos de transformación a largo plazo.

Algunos universitarios inquietos se entrenaron en el uso de las computadoras en sus cursos, auxiliados con numerosos lenguajes de programación, entre los que destacaba señaladamente BASIC. En el medio extrauniversitario enfrentaron un ambiente en el que las computadoras estaban fuera de su alcance, tanto por sus precios como por las formas limitantes de administración y control del uso de los equipos.

Las computadoras alcanzaban precios de diez mil a varios millones de dólares. Al usuario se le cuantificaba -para así cobrarle- el tiempo del procesador central, el tiempo de terminal, la cantidad de líneas impresas, el uso del espacio en disco y hasta por almacenar sus tarjetas y cintas magnéticas.

Un centro donde se sublima esta inquietud, es la bahía de San Francisco en California, en donde se acuña la consigna: “producir mi propia computadora” (aunque no sirva). En este contexto, aparecen personalidades como Ted Nelson, un fuerte y a veces irracional impugnador de las políticas de servicio de IBM, Steve Wozniak, Steven Jobs y Lee Felsenstein, entre otros que se enamoran de esta idea y trabajaron para lograrla.

Este propósito no se hubiese consumado sin el acceso a los novedosos y relativamente baratos microprocesadores.

Un incidente tecnológico-comercial catalizó la disponibilidad del 8080. La historia refiere que la Datapoint, una empresa fabricante de equipo periférico para computadoras, solicitó a Intel y a la TI la elaboración de circuitos de propósito específico con el afán de mejorar y abaratar la fabricación de sus terminales de video. Por otra parte, una empresa japonesa productora de calculadoras electrónicas, suscribió una petición similar, también con el propósito de simplificar su línea de producción.

La TI no atendió la solicitud, pero por su parte, Intel elaboró, con el diseño de Ted Hoff, el 8080, que resultó ser un circuito de propósito general demasiado flexible y elegante, que desde el punto de vista de la lógica y las matemáticas, resultó completo.

El 8080 resultó ser una arquitectura clásica como lo fueron las computadoras PDP-8 y PDP-11, fue demasiado general para las necesidades específicas de Datapoint y paradójicamente, demasiado lento: la velocidad de operación no resultó apropiada para realizar el despliegue de la pantalla de acuerdo con los estándares comerciales y por ésto, fue rechazado por Datapoint.

Este excelente circuito no se comercializó con los propósitos iniciales por lo que Intel lo promovió al mejor postor. Como el circuito no fue resultado de un proyecto militar específico, en este ámbito tampoco llamó la atención sobre sus posibilidades. El mercado que mayor impacto recibió por sus bajos costos, fue el civil.

El inesperado éxito en las ventas permitió a Intel reducir el precio original de 600 dólares a 100 dólares y posteriormente a 25 dólares. Sobra decir que quienes lo compraron estaban en la posibilidad de construir su propia computadora. Aquí inciden los anhelos y la creatividad de aquellas personalidades inquietas, que de inmediato percibieron las posibilidades y estimulados más por éstas que por los primeros resultados, trabajaron en forma decidida hasta consumir su ambición.

3.3 LA PRIMERA OLA

El éxito coronó los esfuerzos de numerosos diseñadores, y con ello se produjo una inquietud por comercializar sus productos. Rápidamente aparecen en el mercado varios proveedores de microcomputadoras organizados en pequeñas compañías. Se recuerda a algunas porque hicieron aportaciones relevantes, a otras, por sus notables deficiencias:

- Micro Instrumentation and Telemetry Systems (MITS) con su Altair 8800, promovido en el número de enero de 1975 de Popular Electronics, introdujo una versión inicial del bus S-100 y con él, el concepto de modularidad.
- Industrial Micro Systems (IMS), con su IMSAI 8080, aportó un panel frontal más elaborado con luces y teclas que reconcilió su equipo con la imagen cinematográfica que se tenía de las computadoras. También es recordada por la introducción del sistema operativo CP/M V1.3, elaborado por Gary Kildall, que es uno de los de mayor uso en la actualidad.
- CROMEMCO, que se caracterizó por el control de calidad de sus productos.
- Processor Technology con su SOL, que insinuó la portabilidad de los equipos e introdujo un acabado en madera.
- PolyMorphic con su tamaño reducido, eliminó el panel frontal y lo reemplazó con la imagen de video y su programa de control para el desarrollo y depuración de programas.
- North Star, que introdujo el primer disco flexible de 5 pulgadas y su sistema operativo DOS.
- Vector Graphics por sus módulos de memoria acoplados a la pantalla de video.
- Digital Group por sus excelentes diseños y su obstinación por promover su propio bus.
- Sphere, por ser uno de los primeros.
- SWTPC, por haber elegido al 6800 de Motorola y por sus reducidas posibilidades de expansión.
- Godbout, por sus excelentes diseños y gran diversidad de accesorios.



POLY 88, una de las primeras microcomputadoras. Emplea el 8080 y fue contruida por PolyMorphic Systems.

Aunque la calidad y las facilidades de los equipos fueron variables, los diseños electrónicos de los módulos fueron similares.

También Intel y Motorola diseñaron microcomputadoras a partir de sus circuitos integrados. Definieron sus propios protocolos de interconexión y produjeron equipos que principalmente se orientaron a clientes industriales, con diseños y costos también industriales. Para el servicio de los particulares, los costos resultaban prohibitivos. En consecuencia, no impactaron este mercado.

En poco tiempo, además de las ya existentes, aparecieron varias firmas que ofrecieron múltiples accesorios para estos equipos. Esto dió lugar a productos tales como:

- Módulos de memoria
- Controles de video
- Generadores de tonos musicales
- Generadores de voz
- Reconocedores de voz
- Unidades y controles para discos flexibles y rígidos
- Convertidores analógicos digitales
- Relojes y contadores de tiempo
- Graficadores en papel
- Unidades y controles para cinta de cassette de audio
- Digitalizadores
- Impresoras

Todos ellos en una gran variedad de modelos con diferentes niveles de elaboración y calidad. Algunos resultaron verdaderos juguetes, otros, permitieron realizar trabajos serios.

No faltó el apoyo de los medios de comunicación. Participaron, en este entusiasmo viejas publicaciones como: Radio Electronics, Computer Design y Popular Electronics. También surgieron nuevas como: Infoworld, Byte, Kilobaud, Interface, Dr. Dobbs Journal y muchas otras más. Mención especial merece la personalidad de Adam Osborne, quién fue encargado de elaborar la descripción de las características técnicas del 8080 en Intel. Posteriormente estableció su editorial a través de la cual difundió las características de los microprocesadores. Se propuso y lo cumplió, elaborar descripciones para cada tipo de procesador interesante. También participó en la difusión y popularización del software.

En estos días, la gente esperaba y leía las revistas con la esperanza de encontrar el anuncio de un nuevo módulo, de un nuevo control o de un equipo periférico más novedoso.

Los equipos que se caracterizaron por que su costo inferior a los mil dólares, aunque emplearon el recurso de la fragmentación para que cada parte tuviera un costo inferior. No se percibía un límite real para el desarrollo; la tecnología avanzaba con rapidez y la gente que participaba en el desarrollo experimentó una verdadera obsesión.

En el terreno del software -aunque menos acelerado que en el hardware- ocurrió un importante desarrollo, que se caracterizó por las siguientes aportaciones: prominencia de Basic como el lenguaje de programación, con una gran diversidad de versiones. Sobresalen el Livermore Basic, Ebasic-CBasic de Gordon Eubanks y G. Kildall y el Microsoft Basic de Bill Gates. La revista Dr. Dobb's dedicó sus primeros tres números a la publicación de un interpretador para el lenguaje Basic.

Entre los sistemas operativos destaca CP/M desarrollado por G. Kildall, para Intel, que por cierto, lo rechazó y posteriormente IMS lo popularizó a tal grado que jugó un papel determinante, quienes lo ignoraron, no prosperaron, entre otros, North Star y PolyMorphic Systems.

3.4 LA SEGUNDA OLA

Con el aumento de la demanda de estos productos, algunas empresas ya establecidas notaron el éxito comercial, en tanto que las originales, enfrentaron el problema del crecimiento.

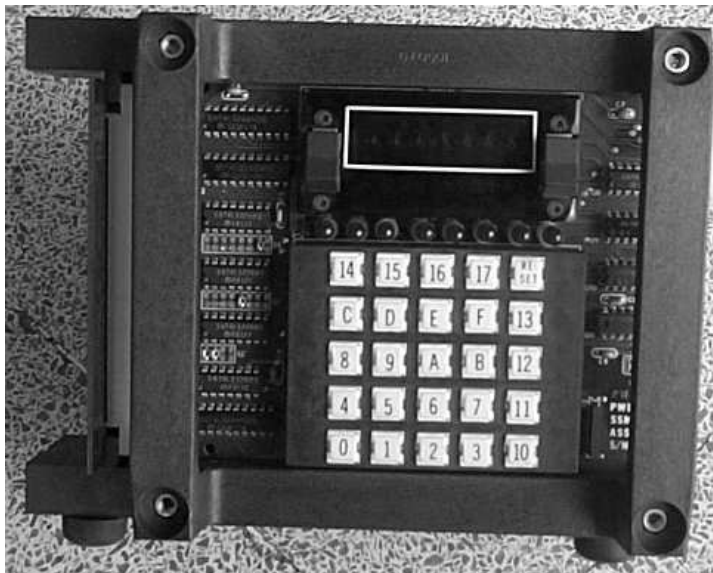
No tardó en identificarse la condición para promover con éxito comercial un diseño electrónico brillante: el apoyo financiero y la infraestructura. La barrera del millón de dólares en las ventas anuales, señala algo más que una cifra. Evidencia la urgente necesidad de promover un cambio radical en las políticas de administración de las empresas. Hay casos que trasponen este umbral con éxito cuando se transforman, otros que deliberadamente se mantienen por debajo para no afrontar la transformación y la mayoría que abandona el mercado.

En esta dinámica, las empresas que prevalecen en esta segunda ola, son:

- Radio Shack, que introdujo diferentes versiones de sus microcomputadoras. Para el efecto, aprovechó su cadena comercial dedicada a la venta de equipos electrónicos económicos orientados a un público juvenil: radios CB, grabadoras, radios de transistores, tocadiscos, etc. La imagen de su calidad es regular pero no extraordinaria. Esta imagen se reveló incompatible para comercializar equipos de alta tecnología. Sus equipos TRS-80 se designan jocosamente como "TRASH-80" (basura-80) en el

ambiente comercial. Paulatinamente la calidad de sus equipos le ha otorgado un lugar en el mercado.

- Apple, uno de los pioneros que sobrevivió y el único que creció considerablemente, fue fundada por Steve Wozniak y Steve Jobs; para su crecimiento, fue determinante la participación de Mike Makkula, ingeniero jubilado de Intel. Makkula se había propuesto promover una compañía desde sus inicios hasta insertarla entre las 100 mayores de Fortune en un período no mayor de 5 años. Probablemente Makkula aportó la experiencia necesaria para que Apple cruzara la barrera del millón de dólares.
- Commodore, con diez años de experiencia en el mercado de las calculadoras electrónicas, contaba ya con la planta y organización necesarias para la operación en gran escala. Su experiencia y capacidad le permitieron la integración vertical en la empresa. Como una medida estratégica, adquirió a MOS Technology. Jack Tramiel tuvo una participación destacada en esta empresa.



Microcomputadora construida con base en el Bus STD, con diseño y calidad industrial. Se emplea en algunos cursos introductorios en la UAP.

3.5 LA TERCERA OLA

Y el orden volvió a imperar en el mundo. Los pocos se hicieron aún menos.

Con mayor crecimiento en el mercado, la IBM se dió cuenta de que debía aprovecharlo y con el cambio en la administración de la empresa, se operó un cambio en sus políticas de ventas. No solamente abandonó su práctica de arrendamiento, sino que se insertó de lleno en el mercado de las microcomputadoras del que se había marginado. Con la introducción de su PC computer, en un par de años acaparó la mayor parte del mercado, gracias al reconocido prestigio que gozan sus servicios en el ámbito de los negocios.

Apple permanece en el mercado ocupando ahora un segundo plano. Diversifica sus equipos con las nuevas versiones Lisa y Macintosh.

Radio Shack pasó a ocupar el tercer lugar.

En esta tercera etapa, los costos continúan bajando y las capacidades de los equipos continúan en aumento. Sin embargo, las configuraciones prácticamente se han uniformado.

Contrariamente, el software se diversifica, pues el crecimiento de la industria da una nueva dimensión al uso y a la producción de programas. Es tan fuerte el impulso, que surgen programas de aplicación que no se hubieran producido en el ámbito de las computadoras tradicionales.

Son característicos de esta generación los programas:

Word Star, T/Maker, d/Base II, Visicalc y otros.

El desarrollo de nuevos lenguajes también se acelera, aparecen entre otros:

Pascal, C y ADA.

Entre los sistemas operativos destacan:

CP/M, AMOS y UNIX.

Como se puede observar, el nacimiento, desarrollo y establecimiento de la industria de las microcomputadoras, registra un paralelismo con otras ramas industriales, entre las que se pueden citar a la automotriz, la del radio y a la cinematográfica. Al principio muchos diseñan y fabrican, pocos pueden crecer y al final unos cuantos dominan el mercado.

El paralelismo es más amplio, así como estas industrias transformaron nuestros hábitos cotidianos, las microcomputadoras transformarán seguramente nuestra conducta. Las potencialidades de estos accesorios, no radican tanto en el equipo, sino en el desarrollo de la programación de control.

¿Experimentará este último una evolución similar?.

LA HISTORIA DEL SOFTWARE

4.1 ¿QUE ES EL SOFTWARE?

Software, hardware y computación, términos que se asocian con la modernidad son en realidad nombres nuevos de prácticas ancestrales.

Todas las culturas antiguas desarrollaron técnicas de cómputo, con sus correspondientes instrumentos y notaciones. Se conservan pocos vestigios de esto.

A los instrumentos, hoy se les conoce como hardware y a las técnicas o procedimientos -la parte inmaterial intangible- como software.

Haciendo una analogía, en la música los instrumentos conforman el hardware, las partituras y la habilidad para crearlas e interpretarlas constituyen el software. Un buen instrumento es importante, pero más importante es un buen músico, entre éste y aquél, media una notación musical que define la calidad del producto.

4.2 UN PRODUCTO DE HISTORIA

El software como se ha visto, no surge con los equipos electrónicos, -aunque es con ellos que adopta el nombre- está presente desde el empleo de ábacos o sumadoras mecánicas. Sin embargo, en estos casos, el software no se encuentra incorporado en el equipo. Es aportado por el operario. La máquina analítica de Charles Babbage, incidentalmente, tuvo su software, y fue una amiga de éste, la legendaria lady Lovelace, quien aportó el software que no se llegó a usar, dado que la máquina nunca se completó.

En el ENIAC el control de las operaciones estaba parcialmente integrado en el equipo. Dicho control era realizado por un circuito que requería un alambrado específico para cada aplicación. Imaginemos lo engorroso que resultaba realambrar el circuito cada vez que cambiaba el uso del ENIAC.

Hasta este momento, no se percibía una diferencia sustancial entre el equipo y el control de las operaciones. El concepto de programa de control almacenado en memoria, aportación popularmente atribuida a John von Neumann, precipitó el desarrollo de software. En éste se perfilaron dos tendencias de desarrollo: los programas de aplicación y los de servicio. Estos últimos tenían como propósito facilitar el desarrollo de programas a partir de programas.

Algunos programas de servicio fueron simples cargadores que permitieron emplear notaciones como el octal o hexadecimal más compactas que el binario. Otros como los ensambladores simplificaron más el proceso al reemplazar las notaciones numéricas con los símbolos mnemónicos que aportaron para describir a cada instrucción de la máquina. El siguiente paso significativo fue la traducción de fórmulas, que permitió la descripción de los algoritmos con el empleo de expresiones algebraicas. Dicha traducción se realiza con programas que se denominan compiladores, generan programas que al ejecutarse producen los resultados.

DIFERENTES NIVELES DE LENGUAJE			
Binario	Hexadecimal	Ensamblador	Lenguaje Algebraico
001000010000001000100000	210220	LXI H,DJ	I = J + K
01001110	4E	MOV C,M	
00100011	23	INX H	
01000110	46	MOV B,M	
11000101	C5	PUSH B	
00100001000000100000100000	210420	LXI H,DK	
01001110	4E	MOV C,M	
00100011	23	INX H	
01000110	46	MOV B,M	
11100001	E1	POP H	
00001001	09	DAD B	
00100010000000000001000000	220020	SHLD DI	
	2000:	DI: DW 0	
	2002:	DJ: DW 0	
	2004:	DJ: DW 0	

Es importante destacar que en tanto los programas de aplicación saturaron los recursos de los equipos, imponiendo sus requerimientos en cuanto a velocidad, precisión en la aritmética y capacidad en los almacenamientos; los programas de servicio repercutieron en la evolución de la arquitectura de los equipos (hardware). Entre las aportaciones más notables, podemos citar el empleo de pilas y el reemplazo de referencias físicas por lógicas.

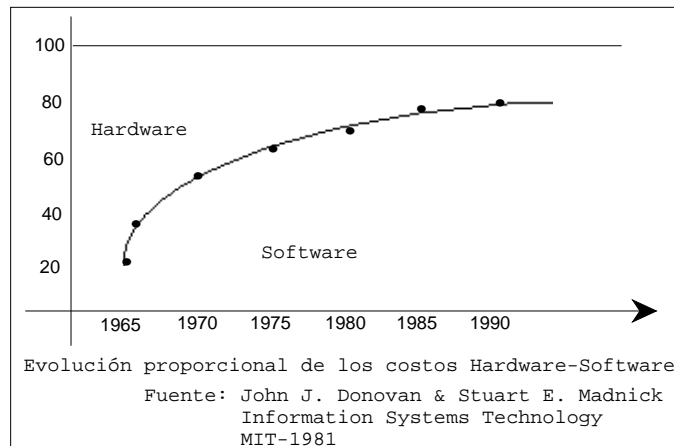
Con la pila (Push Down List), se da lugar al manejo recursivo de los procesos. Por ejemplo, esto ocurre en una oficina administrativa, cuando se pospone la solución de un problema para resolver otro de mayor exigencia. El problema original se suspende y se aborda nuevamente cuando el de mayor exigencia ya ha sido resuelto.

Con el reemplazo de referencias físicas por lógicas, se obtuvo un incremento más real que virtual de los recursos disponibles. Almacenamientos secundarios, registros operacionales, memoria virtual, memoria cache e hizo traslapes (overlay), son algunas de las técnicas que emplean este concepto. El efecto es similar al de las operaciones bancarias nominales con que las instituciones de crédito prestan varias veces su capital.

Los elementos aportados por los programas de servicio, al interrelacionarse configuran el sistema operativo con el cual se administran los recursos disponibles en las computadoras y se establecen líneas de producción para el proceso de programas con una mínima participación del operario: la automatización de la automatización. Al principio, los sistemas operativos brotan como extensiones de los lenguajes. Posteriormente, el fenómeno se invierte de modo que los sistemas operativos configuren el ambiente en el que se desempeñan las aplicaciones y los programas de servicio.

4.3 UNA CLASIFICACIÓN DEL SOFTWARE

El software paulatinamente adquirió mayor importancia que el hardware. En un principio, la proporción favorecía al equipo físico, pero progresivamente, el componente inmaterial adquirió una mayor relevancia hasta hacerse el más importante. Para facilitar su análisis, el software se clasifica generalmente en tres grandes rubros: sistemas operativos, lenguajes de programación y programas de aplicación.



4.4 LOS SISTEMAS OPERATIVOS

Como ya se anotó, surgen como extensiones de los lenguajes de programación, tal es el sistema FAP elaborado para la IBM 709 que orbitaba en torno al lenguaje Fortran, con una mínima participación del operador, estableció un flujo controlado automáticamente para el proceso de lotes de aplicaciones (batch).

Con el aumento tanto de los equipos como de los programas de servicio, las funciones y características de los sistemas operativos se ampliaron y sofisticaron, al grado de adquirir su propio dialecto o lenguaje, tal es el caso del OS/360 para la serie 360 de la IBM, con su JCL (Job Control Language) como lenguaje de control. Algunos sistemas aportaron el concepto de la multiprogramación como una medida para incrementar la productividad de la computadora como el OS/360, otros, la interacción con el usuario representa esfuerzos intelectuales de consideración, el sistema operativo para la serie 370 de la IBM, tuvo un costo de 5 millones de horas hombre.

La interacción con el usuario dió una nueva dimensión al uso de los equipos y con ello, produjo el concepto de tiempo compartido. Entre los sistemas que aplicaron esta filosofía de operación destacan:

- CTSS (Compatible Time Sharing System) desarrollado en el MIT para una IBM 7090.
- TOPS-10 para la PDP-10 de la Digital Equipment Corporation (DEC)

- MULTICS, elaborado también en el MIT para equipos de General Electric y Honeywell.

En las minicomputadoras destacan el OS para la PDP-8 desarrollado por la DEC y UNIX, un sistema operativo para la PDP-11 de DEC, desarrollado por dos investigadores en los laboratorios de la Bell Telephone.

4.5 LENGUAJES DE PROGRAMACIÓN

Dependiendo del criterio que uno admite para calificarlos, existen cientos o miles de lenguajes de programación. Las formas de operación de los programas que emplean a los lenguajes de programación amplían la variedad, pero a la vez ofrecen una clasificación en ensambladores, interpretadores y compiladores.

Entre los programas que emplean lenguajes de programación, los más elementales son los ensambladores, ya que ofrecen al programador un modelo riguroso de la máquina. En ellos se emplean mnemónicos, abreviaturas simbólicas, para cada una de las instrucciones de la máquina y desde ellos, están disponibles para el programador todos los recursos del equipo en su nivel más elemental. Los programas que se encargan de la traducción, se limitan a realizar una correspondencia uno a uno entre los mnemónicos y las instrucciones de máquina.

Los interpretadores utilizan lenguajes, también simbólicos, pero en este caso, cada símbolo tiene una función asociada por el interprete con esta modalidad, el interpretador puede prever las consecuencias de la operación antes de realizarla y emitir, en su caso, diagnósticos para beneficio del usuario. Las operaciones asociadas a cada símbolo no están confinadas a las operaciones disponibles en la máquina, pueden ser instrucciones de otra máquina y de cualquier nivel de complejidad y elaboración.

Los compiladores realizan la traducción de programas desde un lenguaje hasta otro. Las naturalezas de los lenguajes fuente y destino son en principio diferentes. Generalmente, el lenguaje destino es el lenguaje de la máquina o el ensamblador. Es el caso de los compiladores de Fortran que traducen los programas que están descritos por el programador en términos de un lenguaje algebraico y con el empleo de matrices. Rigurosamente, Fortran no fue el primero de los esfuerzos en esta dirección, hubo antes otros como el MAD (Michigan Algebraic Decoder), fue el esfuerzo unificador de la IBM que condujo al Fortran.

Con el empleo de los lenguajes de programación, los usuarios no requieren conocer en detalle las características de la computadora para hacer sus computaciones en gran escala. En la actualidad hay tantos lenguajes de programación que nos limitaremos a hacer algunos comentarios sobre los más conocidos.

Fortran, Algol, Pascal, Ada y Modula, representan una cadena de lenguajes con un origen común, permitir la descripción de computaciones numéricas con el empleo de notaciones algebraicas y matriciales, cada eslabón de la cadena aporta nuevos elementos, la mayor de las veces triviales, pero en la cadena se observa una tendencia muy señalada consistente en que cada vez es más rigurosa la especificación de los tipos de datos que se emplean en los programas.

Algol pretendió ser más un lenguaje matemático para describir algoritmos, que un lenguaje de programación. Introdujo los principios de la estructuración, descomponiendo

el programa en procedimientos y poniendo énfasis en las formas de transmisión de argumentos del programa a los procedimientos secundarios. Para obtener independencia de las particularidades de la entrada y salida de cada equipo, no estableciendo especificaciones sobre estos aspectos.

PASCAL por su parte, acentuó la estructuración de los programas, para ello prescribió el uso de las transferencias incondicionales, limitando las formas de control a las estructuras DO-WHILE e IF-THEN-ELSE, con lo que obligó al usuario a organizar sus programas, simplificando su desarrollo, depuración y mantenimiento.

Fortran, Basic, Pascal, en esta serie de lenguajes. También con un origen común, el énfasis que se observa consiste en ofrecer una mayor independencia en las especificaciones de entrada y salida, en mejores facilidades para el manejo de textos en los programas, un empleo interactivo del lenguaje, así como una acusada simplificación en la presentación de los resultados.

Específicamente, Basic eliminó las tediosas especificaciones de formato, indispensables en Fortran, para la presentación de los resultados.

También hubo lenguajes que enfatizaron tanto la especificación de los datos, que llegaron al extremo de no solo ofrecer datos numéricos y textos, sino también de integrarles una estructura.

Tal es el caso de "C", Alpha-Basic (una variante de Basic para los equipos Alpha-Micro), e inclusive COBOL, que se orientó más a ambientes administrativos que a los de la ingeniería, por lo que su aritmética fue muy limitada.

Algunos de los lenguajes, introdujeron características muy especializadas, tal es el caso de COBOL, que con su enfoque administrativo, incorporó una excelente colección de subrutinas para el manejo y la organización de fuertes volúmenes de datos alfanuméricos, entre ellas el ISAM (Indexed Sequential Access Method) que establecieron los fundamentos para los manejadores de bases de datos.

Y también el de "C", que pretendió utilizar el lenguaje de máquina de la PDP-11 de DEC, como lenguaje de alto nivel.

Al margen de toda esta colección de lenguajes en los que el cálculo numérico juega el papel central, surgen otros lenguajes, los clasificados como no numéricos, que se caracterizaron por su énfasis en el manejo de cadenas de caracteres, la edición de textos, el reconocimientos de patrones y la propiedad de que el programa fuente es de la misma naturaleza que los datos que manejan.

Por su enfoque diferente, este tipo de lenguajes se utilizó en aplicaciones menos convencionales, como lo son la compilación de lenguajes y para los estudios y desarrollos realizados en el terreno de la inteligencia artificial. Sobresalen entre ellos LISP para el manejo de listas; COMIT y SNOBOL en el reconocimiento de patrones. Más adelante, se agregan IPL V y PROLOG.

Por su naturaleza, los lenguajes no numéricos enfrentaron en sus inicios problemas con la velocidad de los equipos y las capacidades de sus memorias. Por esto, su desarrollo registró una pausa aunque en fechas recientes, con el apoyo de los nuevos equipos, han recibido un nuevo impulso.

Si bien los primeros resultados en este campo fueron poco convincentes: programas de juego decepcionantes, sistemas rígidos de traducción, y demostraciones de teoremas

triviales; el nuevo período ha ofrecido resultados sumamente interesantes, entre los que podemos citar a los sistemas expertos.

Estos sistemas expertos, además de organizar considerables volúmenes de datos y reducirlos a través de operaciones lógicas, realizan ya inferencias elementales. Orientados a la geología, la medicina y al diagnóstico en general, se perfilan como consejeros capaces de emitir juicios.

Uno de los éxitos más sonados y que mayor controversia ha despertado, es la inferencia que realizó un sistema experto en 1982, el Prospector que permitió identificar un filón de molibdeno, en un sitio en donde nadie lo habría imaginado.

4.6 LOS PROGRAMAS DE APLICACIÓN

Los programas de aplicación justificaron la construcción de los equipos, inicialmente consistieron de aplicaciones militares y científicas como es el caso del cálculo de la trayectoria de proyectiles, de las simulaciones de efectos termodinámicos y del análisis del espacio aéreo. Con la comercialización de las computadoras, sus aplicaciones se orientaron también al apoyo de la administración pública y de las grandes empresas: sistemas de nóminas, sistemas contables, controles de inventarios, producción de parámetros, estadísticas y aplicación científica.

Algunos fueron de uso exclusivo, como los proyectos APOLLO y SAGE. Este último acoplaba un sistema de radar al computador, para controlar el espacio aéreo. Inició sus operaciones en 1958, después de 5 años de trabajos con un esfuerzo equivalente a 3.6 millones de horas hombre. Otras aplicaciones tuvieron un empleo también limitado, es el caso del sistema de reservaciones para vuelos de la TWA, con una IBM 370/168 y un sistema de teleproceso que integra más de 3000 terminales en América y Europa, realiza dos millones de transacciones por día. El conjunto de programas que soporta esta aplicación contiene más de 1.5 millones de líneas en sus códigos fuente y representa un esfuerzo equivalente a 100 mil horas hombre.

Otros, de uso más común, experimentaron generalizaciones a través de la estructuración, estandarización y parametrización de los datos que procesan. Es el caso de los paquetes estadísticos como SPSS y BMD, de simulación como GASP y GPSS, científicos como STATPAK, EISPACK y FUNPAK, y aquellos para la optimización y el modelaje económico.

Con la incorporación de las nuevas computadoras a la producción industrial, se dio origen a nuevos campos, el CAD/CAM (Computer Aided Design y Computer Aided Manufacturing). La Texas Instruments y la Intel fueron unas de las primeras en incorporar el apoyo de las computadoras para el diseño de sus circuitos integrados. La tendencia actual se orienta en el sentido de desarrollar las aplicaciones en forma integral con el apoyo de un manejador de bases de datos para organizar la información.



Centro de Cálculo de la Universidad Autónoma de Puebla.

HISTORIA DEL SOFTWARE

(Segunda parte)

5.1 EL SOFTWARE PARA LAS MICROCOMPUTADORAS

Fue tan importante el papel de las letras minúsculas en el desarrollo del software para las microcomputadoras, que se identificaron como una de sus principales características. A propósito de esto, circuló en el ambiente un eufemismo: las microcomputadoras emplean letras chicas (minúsculas) y las computadoras, letras grandes (mayúsculas). Esta ironía encierra algo de verdad, el mismo desarrollo tecnológico que dió origen a las microcomputadoras, permitió un amplio uso con costos económicos de letras minúsculas en el ambiente de la computación.

El uso de las letras minúsculas en la computación no son ciertamente una consecuencia de las microcomputadoras, de hecho ya se empleaban esporádicamente en algunas computadoras, inclusive sus códigos se habían estandarizado en 1967 (documento USASCII X3.4-1967 y USASI X3.4/428-1968); pero su empleo generalizado jugó un papel determinante en el desarrollo del software para las microcomputadoras. Seguramente algunas de sus principales aplicaciones, entre ellas la edición de documentos, no tendrían lugar sin el concurso de las minúsculas. Uno de los principales aspectos referente a la calidad de las terminales e impresoras fue la estética de sus letras minúsculas.

Aunque la evolución del software para las microcomputadoras presenta paralelismos en casi todos los aspectos con la evolución que registró el software de las primeras computadoras, hubo un detalle de considerable valor. Los primeros programas en las grandes computadoras que existían. Estas disponían de sistemas de desarrollo que consistían fundamentalmente en ensambladores de paso (cross assemblers) y simuladores, programas que emulaban la ejecución del programa en el equipo de desarrollo.

Generalmente, estos sistemas provocaron el desarrollo para el software; se operaban en enormes computadoras a través de los sistemas de tiempo compartido, en algunos de los casos, por medio de redes telefónicas. Inclusive, algunos de estos sistemas fueron proporcionados como un servicio adicional para promover la venta de sus productos, por los fabricantes de los circuitos integrados.

Tal es el caso de TYMNET, para el desarrolló de programas para los circuitos de Intel y NCSS para los de National, disponibles en sistemas de tiempo compartido en

computadoras General Electric. Tampoco faltaron estas facilidades para los clientes de Motorola.

Algunas universidades desarrollaron programas en sus computadoras para facilitar el desarrollo de programas para las microcomputadoras. Con empleo de los ensambladores y simuladores, se desarrollaba el programa para la microcomputadora que posteriormente se deposita en memorias ROM (Read Only Memory) para su uso final.

Las primeras aplicaciones consistieron en programas de control para conmutadores telefónicos, semáforos, máquinas de escribir, máquinas y herramientas industriales, instrumental de laboratorio y equipos periféricos para computadoras: terminales de video, impresoras, graficadores, controles de disco, controles de cinta y otros.

Al aparecer la primera microcomputadora popular, la Altair 8800, la programación se realizó por medio de los interruptores y switches que incluía su panel frontal. Con todo el entusiasmo que la efervescencia producía, algunos fanáticos en Berkeley y en otras universidades desarrollaron -con el solo recurso de los switches y luces- pequeños monitores interactivos para el desarrollo de programas con el uso de teletipos y terminales de video y con el empleo de notaciones más compactas que el binario. La experiencia no resultó vana, el uso de los switches les permitió una mejor comprensión de las capacidades y potencialidades de los equipos.

Estos primeros programas utilizaron inicialmente las cintas de papel de los teletipos para almacenar los datos y programas y posteriormente, los cassettes de audio. De manera natural, aparecen numerosas versiones de programas editores y ensambladores -para uso en las microcomputadoras- que se distribuían por medio de la cinta de cassette.

Con la participación de William Gates en MITS, la Altair 8800 incorporó en poco tiempo un interpretador de Basic entre su repertorio de programas de servicio. Al parecer, el Basic de Gates fue desarrollado con el empleo de un sistema de desarrollo disponible en alguna microcomputadora, como parte de un proyecto académico y con el modelo de las versiones (de Basic) disponibles en otros equipos.

Había otros lenguajes de programación como candidatos para usarse en las microcomputadoras además del Basic, se especulaba con Forth, APL y Fortran. Es verdad que algunos de ellos resultaron demasiado concisos y taquigráficos, otros carecían del aspecto interactivo. Sobra decir que las especificaciones del formato de Fortran representaron un verdadero obstáculo para su elección.

A pesar de que Ed Roberts, fundador de MITS, se atribuye (y por ello ha recibido múltiples regaños y algunos reproches) la elección de Basic como el lenguaje de programación para las microcomputadoras, muy probablemente esta elección -por los usuarios y no por Roberts- fue consecuencia de sus características interactivas, de la simplicidad con que permite la descripción y edición de resultados y particularmente, por la amplia difusión que tenía este lenguaje en los medios universitarios a finales de los 60's.

Basic fue elegido en forma unánime, a pesar de los esfuerzos que algunas empresas y sectores realizaron para popularizar otros lenguajes.

En uno de los primeros números de la revista BYTE realizó un esfuerzo por popularizar a Pascal. Posteriormente, los esfuerzos se orientaron en dirección de otros lenguajes como Modula y Prolog. Finalmente, Pascal adquirió importancia, sin llegar a desplazar al Basic.

El Basic de Gates fue la única versión disponible, la revista Dr. Dobb's publicó en sus primeros tres números -con un formato de tabloide- otra versión del lenguaje. Por su parte, la revista BYTE distribuyó otra versión en un disco como una forma de promoción. El disco se colocaba en una tornamesa conectada a los controles del cassette de audio, para la lectura del programa. Otras versiones de Basic surgen como proyectos académicos, es el caso del trabajo de Gordon Eubanks y Gary Kildall, que dió como resultado EBasic, cuya evolución produjo una versión comercial denominada CBasic.

Posteriormente, Gates se independizó de MITS y estableció su propia empresa: Microsoft Inc., para la venta y distribución de software. Esta empresa dió origen a la versión más popular del lenguaje, el Basic de Microsoft . Motivado quizás por el amplio uso que tenían los lenguajes Fortran y Cobol en las computadoras tradicionales, esta compañía también produjo, al cabo de dos años, compiladores de estos lenguajes para el 8080.

Con el recurso de los ensambladores y de las versiones de Basic da principio un desarrollo de programas de aplicación, en el que se empleó de manera preponderante la cinta de cassette como medio de intercambio y distribución. Los primeros programas de aplicación consistieron de juegos electrónicos del tipo de Star-Trek, Mammouth Cave y algunos programas de servicio para la edición de documentos, entre ellos el programa Electric Pencil.

```

        VB = DB-EB
        JJ = JJ+1
        U(JJ) = XX-EB*((EX-XX)/(DB-EB))
42      IF (JJ.EQ.0) GOTO 44
        DO 43 KK=1,JJ
        IF ((KK.EQ.1).AND.(JJ.EQ.1)) XI = U1
        IF ((KK.EQ.1).AND.(JJ.EQ.2)) XI = S*AMIN1(S*U1,S*U2)
        IF (KK.EQ.2) XI = S*AMAX1(S*U1,S*U2)
        F = (XI-XX)/(EX-XX)
        YI = YY+F*(WY-YY)
        CALL PL (XI,YI,((KK.EQ.1).AND.VV))
        IF ((ABS(XX-XI).LE.EP2).OR.(ABS(XI-EX).LE.EP2)) GOTO 43
        IF ((KK.EQ.2).AND.(ABS(U1-U2).LE.EP2)) GOTO 43
        J1 = II(J1)
        X1(J1) = XI
        T1(J1) = TT+F*(TO-TT)
        B1(J1) = BB+F*(BO-BB)
43      CONTINUE
44      IF ((J1.LT.2).OR.(J1.GT.M-1)) GOTO 46
        IF (VISSL(EX,TE,X1,T1,J1+K).GT.EP1) GOTO 46
        IF (VISSL(EX,BE,X1,B1,J1+K).LT.EP1) GOTO 48
46      J1 = II(J1)
48      X1(J1) = EX
        T1(J1) = TE
        B1(J1) = BE
        VV = ((VT.GT.EP).OR.( -VB.GT.EP)).AND.P(J)
        CALL PL (EX,WY,VV)
        L = .FALSE.
        ET = DT
        EB = DB
        XX = EX
        YY = WY
        TT = TO
        BB = BO
        GO TO 30

```

Segmento de la subrutina VISBO, escrita en Fortran. La subrutina VISBO, es la parte medular del paquete PLOT que permite graficar superficies tridimensionales opacas con la técnica de líneas escondidas.

En materia de sistemas operativos, el panorama es menos variado. Al principio, las

microcomputadoras carecieron completamente de sistema operativo. Con la aparición de Basic y de los discos flexibles (North Star introdujo el primero de 5 pulgadas), surgen las primeras versiones, de los sistemas operativos. Aparece el OS de North Star y el CUTS de Processor Technology, como extensiones de Basic.

El más importante de los sistemas operativos fue CP/M, desarrollado por Gary Kildall para el 8080. Inicialmente, Kildall fue contratado por Intel para el desarrollo de un compilador de PL/M (PL/M es la versión para microcomputadoras del lenguaje PL/I de la IBM) para el 8080. Una vez concluido el proyecto, continuó sus trabajos en la dirección de un sistema operativo, posiblemente inspirado en algunos sistemas operativos de los equipos de DEC específicamente en el OS de la PDP-8.

El sistema operativo CP/M consiste principalmente de un monitor residente, ensamblador, editor, depurador de programas y accesorios para organizar archivos de información de disco, intercambiar los datos entre los equipos periféricos y para configurar versiones apropiadas a las características del equipo.

Kildall propuso un sistema operativo a Intel y ésta lo rechazó, quizás porque contaba ya con su sistema operativo ISIS II para el 8080, por lo que Kildall decidió comercializar la versión 1.3 de su CP/M y para el efecto, fundó la empresa Digital Research.

Rápidamente Cromemco adoptó el CP/M bajo su designación CDOS, y lo mismo hizo IMS para su IMSAI con el nombre original. Fue tan evidente su aceptación que estableció un estándar.

Cuando el éxito de la Apple empezó a ser evidente, Microsoft puso en venta su Soft-Card, accesorio que permitió usar el software que había desarrollado para el 8080, en las microcomputadoras Apple, con la marginación del microprocesador original de la Apple, un 6502 de MOS Technology, en favor de un Z-80 de Zilog, procesador que tuvo como virtud su compatibilidad con el 8080 de Intel.

Con base en el CP/M de Digital Research y en el Basic de Microsoft, se inicia un impetuoso desarrollo de programas de aplicación. Aparecen varias empresas dedicadas a la distribución de software, pero ahora en discos flexibles. Surgen así VisiCorp, Software Arts, Peachtree Software, Ashton Tate, Human Engineered Software, Sorcim y Micropro, entre otros.

Se produce una gran cantidad de programas de aplicación, enfocados a facilitar algunas tareas administrativas específicas: preparación de documentos, procesamiento de textos, preparación de presupuestos y balances, y para la organización de fichas bibliográficas y expedientes.

Inicialmente los programas de aplicación se elaboraron en Basic o en otros lenguajes de programación, y posteriormente se optimizaron programando algunas de sus partes más importantes en lenguaje de máquina para mejorar su rendimiento. Entre los programas de mayor éxito podemos citar a: VisiCalc (auxiliar contable), WordStar (editor de textos), MailMerge (procesador de textos), dBase II (manejador de fichas) y a T/Maker (edición y manejo de tablas).

Específicamente, VisiCalc fue declarado el programa del año en 1982, por el “hit parade” que mantiene la revista Infoworld para productos de software. Su éxito es indiscutible, pues en 1983 VisiCorp vendió 400,000 ejemplares de este programa, actualmente es el programa de mayor venta con una cifra récord de más de 700,000 ejemplares. En

Ventas anuales de los últimos 3 años pronósticos para 1985							
EX		1982	1983	1984	Prom.	Razón de crecimiento	1985 (Est.)
		9999.9	9999.9	9999.9	9999.9	999.9	9999.9
AC1		+	+	+	avr		
AC2		-	+	+		grw	
UC3				+		prj	+
Ventas anuales (miles)							
++	A	12.1	13.9	14.9	13.6	11.0	16.5
++	B	18.1	19.2	16.5	17.9	-4.5	15.8
+ +	C	5.8	9.2	13.1	9.4	50.3	19.7
===	Total	36.0	42.3	44.5	40.9	11.2	52.0
Porcentajes							
%	A	33.6	32.9	33.5	33.3	-0.2	31.8
%	B	50.3	45.4	37.1	44.2	-14.1	30.3
%	C	16.1	21.7	29.4	22.4	35.2	37.9

Ejemplo de los resultados que se obtienen con el programa de aplicación T/Maker. Este programa se emplea como auxiliar contable en la preparación de presupuestos y balances. Con el cuadro interior se indican los datos básicos, las demás cantidades se obtienen a partir de las ecuaciones aritméticas (primer renglón y columna) con el comando compute.

opinión de Steve Wozniak, cofundador de Apple, en una buena medida el éxito comercial de la Apple II se debió a la aparición de este programa. Mucha gente compró el Apple para poder usar VisiCalc.

Nada es permanente, mientras el mundo se dedicaba al desarrollo del Software para el 8080 para explotar sus características, la tecnología de los semiconductores trabajaba en dirección de nuevos productos: los microprocesadores de 16 bits. En 1979 Intel anuncia su nuevo microprocesador 8086, aunque tarda más de un año en ponerlo en el mercado. También Motorola introduce su microprocesador de 16 bits, el 68000. No tardaron en aparecer diseños a partir de los nuevos circuitos, Seattle Computer es la primera en anunciar su módulo con procesador 8086 para el bus S-100 y Godboud, por su parte, introduce un módulo con doble procesador, un 8085 y un 8088 (compatibles con el 8080 y 8086, respectivamente).

También aparecen diseños construidos a partir del 68000 de Motorola. Dual Systems introduce un módulo para el bus S-100 y Apple anuncia las nuevas versiones de sus equipos: Lisa y Macintosh.

Con los nuevos equipos que incorporaron al 8086, se inicia la producción del software, en un principio con el simple traslado de aplicaciones y programas de servicio, que en una buena medida fue posible gracias a la compatibilidad, que deliberadamente Intel conservó en su 8086 con relación al 8080, aunque el traslado del software exigió un esfuerzo de consideración. Esto mismo introdujo una fuerte resistencia al cambio, todo mundo producía programas para el 8080 y todo mundo también trataba de evitar a los nuevos circuitos.

También como un traslado de software, aparece el sistema operativo UNIX, particu-

laramente en las microcomputadoras que contienen un microprocesador con arquitectura compatible con la PDP-11. Específicamente, aparece en los equipos que emplean el 68000 de Motorola. Las posibilidades de UNIX para dominar el mercado son limitadas ya que las microcomputadoras no poseen las mismas características de la PDP-11 que permitieron el éxito de UNIX en su versión original: la memoria virtual, la memoria cache y la capacidad y velocidad de los periféricos.

La inercia rompe cuando la IBM decide participar en el mercado de las microcomputadoras, ante esta amenaza, todos enfocan su atención en el 8086, después de que la IBM lo señaló al perfilar su producto de entrada al mercado, la IBM PC (Personal Computer).

Para realizar sus propósitos por el lado del hardware, la IBM adquirió parte de las acciones de Intel. Por el lado del software, entró en pláticas con Digital Research para negociar el empleo del CP/M, y con Microsoft para el empleo de sus programas de servicio y compiladores de lenguajes. Al parecer, la IBM no consigue un acuerdo favorable con Digital Research, por lo que establece otro contrato con Microsoft, esta vez para el desarrollo de otro sistema operativo para el 8086. Para atender esta nueva petición, Microsoft contrata a personal que contaba ya con la experiencia adquirida con los primeros esfuerzos de Seattle Computer.

Con lo anterior, surgen dos sistemas operativos para el 8086: CPM/86 desarrollado por Digital Research y MS-DOS por Microsoft. El producto de Digital Research correspondió exclusivamente al traslado del CP/M original del 8080. El de Microsoft se caracterizó, porque asimiló algunos de los aspectos convenientes y atractivos que ofrece el sistema operativo UNIX.

```
[main program]
{
  {letter} ((and,<1>,(or,(IVL,A,Z),(IVL,a,z),,$))A
  {ddigit} ((and,<1>,(IVL,0,9)))B
  {hdigit} ((and,<1>,(or,(IVL,0,9),(IVL,A,f),(IVL,a,f,)))C
  {dstring} (<:B:>(ITR,<:B:>))D
  {white} ((or, (^I))(ITR,(or, (^I))))E
  {anum} (<:A:>(ITR,(or,<:A:>,<:B:>)))P
  {fconst} ((ITR,<:B:>).(ITR,<:B:>)(or,(or,E,e)(or,+,-)<:E:>))G
  {bconst} ((or,.true,.false.))H
  {aconst} ((or,<'>(ITR,(or,<:A:>,<:B:>)<'>,nHxxx))I
  {const} ((or,<:G:>,<:D:>,<:H:>,<(><:G:><,><:G:><)>))J
  {prim} ((or,<:J:>,<(><:S:><)>,<:F:><(><:S:>(ITR,<,><:S:><)>,<:F:>))K
  {exponent} ((or,<:K:>(or,**,^)<:K:>,<:K:>))L
  {prod quo} (<:L:>(ITR,(or,*,/)<:L:>))M
  {sum diff} ((or,-,.)<:M:>(ITR,(or,+,-)<:M:>))N
  {vbool} ((or,.not.,.<:N:>))O
  {relexp} (<:O:>(or,(<:T:>.<:O:>))P
  {and} (<:P:>(ITR,.and.<:P:>))Q
  {exprn} (<:Q:>(ITR,(or,.or,..xor.)<:Q:>))S
  {relation} ((r,eq,ne,lt,le,gt,ge))T
  {index} ((or,(or,<:D:>+,<:F:>(or,(or,+,-)<:D:>,<:D:>))U
  {lvalue} (<:F:>(or,<(><:U:>(ITR,<,><:U:><)>))V
  {data type} ((or,character,integer,logical,real,complex,external,dimension))W
  {i do tail} (<:F:>=<:S:><,><:S:>(or,<,><:S:>,>))X
  }
  { ($R,<9>.FOR))R
  }
  (9)
}

Una parte de un compilador para el lenguaje Fortran. El
compilador está escrito en CNVRT. Este lenguaje, orientado
a la transformación simbólica, utiliza el principio del
reconocimiento de patrones y el remplazo de esqueletos.
```

Recientemente se anunciaron nuevos productos de Intel, el 8087, el 80286 y el 80386, a sólo tres años de distancia del anterior. Transcurrieron ya tres años desde la aparición del 8086 y todavía no se dispone del software adecuado para que se utilicen todas sus

capacidades. Sobre todo, porque no se han identificado aplicaciones que verdaderamente requieran y exploten las características de los microprocesadores de 16 bits. Para la preparación de documentos y los apoyos contables, resultan innecesarios los nuevos equipos. Se ha hecho evidente la falta de entusiasmo e interés que caracterizó al desarrollo del software para los primeros equipos.

Parece, pues, que asistimos a un período de transición; los nuevos productos sólo se han perfilado, aún no se han consolidado; sin embargo, los anteriores ya empezaron a declinar y están por desaparecer. Por si fuera poco, recientemente se anunció la bancarrota o inminente bancarrota de las principales casas de software. Y VisiCorp no iba a ser la excepción, al parecer, esto resulta como consecuencia de una saturación del mercado.

Sin embargo, el desarrollo de la industria continuará indudablemente en las líneas ya establecidas y en otras que ya se perfilan como innovaciones interesantes.

5.2 CONCLUSIONES

1. El software tiene mayor vigencia temporal que el hardware. El componente inmaterial e intangible persiste sobre los equipos. Las computadoras surgen y desaparecen, los programas perduran sobreviviendo a los equipos, emigrando y adaptándose de un equipo a otro. Dos conceptos y realizaciones, entre más abstractos, mayor vigencia temporal tienen.
2. El primero que llega al mercado con un producto aceptable, aunque no sea el óptimo, domina el mercado. El primero siempre fija la imagen, que los demás competidores intentarán reproducir, incluidos todos sus inconvenientes.
3. Existen pocos motivos legítimos para comprar una computadora. Aún cuando se podrían citar cientos y quizás miles de razones para que la gente adquiera una computadora: prestigio, comodidad, curiosidad, novedad. La utilidad y el sentido práctico terminan por imponerse y son pocos los motivos que justifican la mayoría de las adquisiciones. Edición de documentos (Wordstar, Vedit), elaboración de presupuestos y balances (VisiCalc, LOTUS 1-2-3), manejadores de datos (dBase II, dBase III, Fox, T/Maker) y párele de contar.
4. La costumbre se apodera rápidamente de los usuarios. Quienes adquieren un producto, de inmediato desarrollan una rutina cotidiana. Es tan considerable el esfuerzo necesario para aprender a usar un programa de aplicación, que surge una resistencia al cambio, a tal grado que el usuario se siente confortable con los caprichos del programa. La resistencia resulta mayor cuando los nuevos productos sólo aportan mejoras insignificantes y triviales.
5. La industria de las microcomputadoras registra una evolución gradual. Las empresas involucradas en el ambiente, tienen su ritmo. La aparición de nuevos circuitos y equipo periféricos, demandan mejoras e incluso cambios de filosofía en los sistemas operativos y las aplicaciones. Entre la aparición del hardware y su plena utilización con el software apropiado, transcurre un período medio de 5 años hasta este momento.

Podemos concluir continuando con la analogía musical, que la gran mayoría de los usuarios se limitan a usar el instrumento al nivel de una pianola, a la que se dota con un rollo y se hace funcionar con los pies.

El verdadero impacto innovador de las microcomputadoras se producirá cuando los usuarios se decidan a crear sus propias aplicaciones, recreando y potenciando su trabajo; cuando compongan sus propias melodías. En tanto esto no ocurre, la situación será equivalente a comprar un piano Steinway o un violín Stradivarius para tocar una sola canción de cuna.

EL CÁLCULO NUMÉRICO Y EL LENGUAJE FORTRAN

“Estudie programación Fortran en 3 semanas y entre al mundo de la computación electrónica” o bien “Dedique un poco de su tiempo libre y en tres semanas usted será un experto en computación”. Estos son algunos de los slogans más socorridos en la propaganda de las academias que ofrecen capacitación técnica en temas relacionados con la computación.

Por su parte, la IBM y sus competidores también utilizó propaganda del mismo género cuando, en la mitad de la década de los 50's, promovió y difundió el uso del lenguaje Fortran. Si bien agregó el requisito de la familiaridad con las notaciones algebraicas y matriciales, también redujo el tiempo de la capacitación a tres horas,

No resulta difícil comprender lo exagerado de estas afirmaciones, varias generaciones de egresados de la licenciatura en Ciencias de la Computación que ofrece la UAP pueden constatarlo con su propia experiencia.

No basta con una computadora para resolver un problema específico, además del equipo, se requiere de un programa que controle la actividad de la computadora. El programa de control juega un papel muy importante, dado que establece en cada momento las operaciones que la computadora debe realizar para dar solución al problema.

Cada problema requiere de un programa de control diferente, que se conforma con las instrucciones que en su nivel elemental dispone la computadora. Como estas instrucciones son muy simples, la elaboración de los programas resulta ser una labor de considerable complejidad que consume grandes cantidades de recursos.

Por esta razón se han desarrollado diversos accesorios que facilitan la elaboración de los programas, entre los que destacan, de manera señalada los lenguajes de programación. El primer lenguaje de programación que gozó de amplia difusión fue Fortran (Formula Translator). Su diseño se orientó a las aplicaciones numéricas que plantearon la ingeniería y la ciencia, en donde las soluciones se describen de manera natural con el empleo de expresiones algebraicas y matrices.

El programa que realiza la traducción del programa se designa compilador.

Fortran incorporó en su inicio pocos elementos, además de las fórmulas algebraicas y el empleo de matrices, integró algunas instancias de control. Transferencias incondicionales, elección de opciones y la repetición de ciclos iterativos permitieron describir el control del

flujo del lenguaje. Para establecer comunicación con el mundo exterior, contó con proposiciones, sensiblemente influenciadas por las tarjetas perforadas y las cintas magnéticas en virtud de que fueron los medios disponibles.

Incorporó también la posibilidad de definir y usar subrutinas. Con esto el programador pudo organizar mejor su programa y definir sus propios procedimientos compuestos asignándoles el nombre, con el cual les hacía referencia cada que los deseaba utilizar. La definición y uso de subrutinas permitió desde un principio el uso de argumentos lo cual planteó el problema del intercambio de datos.

Otros aspectos que determinaron las características del lenguaje fueron las primeras aplicaciones, específicamente, a través de los requerimientos que plantearon en cuanto a la precisión de los datos, tamaño del programa y la velocidad del proceso.

Fortran en sus inicios no incorporó algunas facilidades como los índices flotantes y los incrementos flotantes o negativos, no tanto porque no se habían identificado, sino porque explícitamente fueron excluidas en su definición. Con la evolución de los equipos, el lenguaje posteriormente los incorporó.

Se da el caso en que la versatilidad y abundancia de recursos en los lenguajes se revierten en contra del usuario. No es este el caso de Fortran porque continuó su evolución sin proponerse incorporar accesorios innecesarios. Otros lenguajes han integrado recursos que nunca se emplean; en estos casos no sólo existe desperdicio sino que éste, introduce elementos de perturbación para el usuario.

La presencia de Fortran facilitó el desarrollo de aplicaciones y potenció el cálculo numérico en la ingeniería y las ciencias. Con su utilización se desarrollaron aplicaciones numéricas de gran escala. Su empleo redujo el tiempo en preparación del programa en al menos un orden de magnitud.

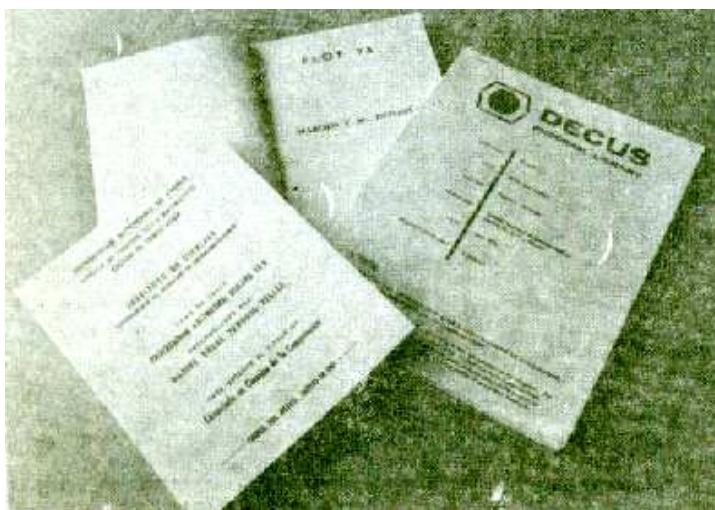
6.1 CULTURA GENERAL Y USO DEL LENGUAJE

Un conocimiento superficial del lenguaje -saber qué es y para qué sirve- satisface las necesidades culturales de un gran número de personas. Pero los usuarios de las computadoras, aquellos que las emplean en la solución de problemas, requieren conocer con mayor detalle y profundidad las características y ventajas que ofrecen los lenguajes de programación.

Para este propósito se cuenta con una amplia bibliografía que describe con cierto detalle los aspectos generales. Las características particulares de cada compilador -que en una buena medida depende del equipo- se describen en los manuales que preparan los proveedores.

A pesar de que estos manuales representan un apoyo significativo, usualmente resultan incompletos, en vista de abordar de manera superficial algunas características del compilador y en otros casos las omiten o eluden su descripción. Algunas de las veces, las omisiones son deliberadas, ya que los proveedores desean mantener algunas de sus características como secretos comerciales.

Por lo regular, para el usuario el conocimiento del lenguaje es parcial e inexacto, por méritos propios y ajenos. Los vendedores, instructores y maestros adolecen de la misma deficiencia, toda vez que están imposibilitados para subsanar las deficiencias de los manuales e instructivos.



A manera de ejemplo, los libros que describen los movimientos de las piezas del ajedrez son útiles para comprender el juego. También son útiles las crónicas y análisis de las partidas más notorias de los grandes maestros, pero son insuficientes ya que ningún tratado puede contemplar y analizar todas las posibilidades que ofrece este juego.

Para un mismo lenguaje hay una gran variedad de compiladores originada por las diferencias en los equipos de las diferentes compañías. Es también notable la gran diversidad en las calidades de los compiladores. Estas diferencias impiden en buena medida producir aplicaciones independientes del equipo y versiones del lenguaje. Para mitigar esto, algunos lenguajes -Fortran incluido- están definidos a través de normas y estándares, elaborados por comités de proveedores y usuarios que desafortunadamente no resuelven el problema ya que en ellos no están previstas todas las posibilidades que se pueden presentar. Los compiladores son elaboraciones demasiado complejas como para que se puedan describir con fidelidad todas sus características.

Adicionalmente, los compiladores pueden contener errores o presentar interpretaciones equivocadas -accidentales o deliberadas- de los estándares.

Por otra parte, los estándares están descritos en términos más propios de los abogados que de especialistas en computación o matemáticos. Llevando las cosas al extremo, les podríamos describir como acuerdos entre piratas y bandoleros, dispuestos a suscribirlos pero no a cumplirlos y a regirse por ellos.

La mejor referencia para el compilador, la más fiel, la que no omita descripción de las características, ni permite interpretaciones equivocadas, es el listado del programa fuente del compilador.

Como en los ambientes jurídicos, aquí también priva la regla de la mejor evidencia: no se admite el testimonio de un testigo o un duplicado como sustituto del contrato o prueba original.

Desafortunadamente, es política de uso corriente de los proveedores de equipos y programas el ofrecer los programas en binario y nunca los programas fuente. Estos últimos

algunas veces también se pueden adquirir, pero su precio los hace prohibitivos, en ocasiones equivalentes al del equipo completo.

Las empresas que los producen, cotizan a los programas fuente de un compilador o de un sistema operativo en varias decenas de miles de dólares como una medida de evitar la competencia desleal de sus clientes.

Por esta razón, el Instituto de Ciencias de la UAP ha promovido a través de su participación en la Escuela de Ciencias Físico Matemáticas, que los usuarios -en este caso los estudiantes universitarios- conozcan todos los aspectos relacionados con los lenguajes, de manera íntima y profunda y no exclusivamente con su uso. Aunque es justo decir que en este empeño se ha obtenido un éxito relativamente parcial. De acuerdo con esta intención, en el Instituto se han analizado los compiladores de los equipos disponibles, entre otros, el F80 (Fortran para la 8080), FTN (Fortran para la PDP-8), Fortran de la IBM-1130, el compilador para "C" de BDS, y Alpha-Basic (Basic para la Alpha-Micro).

6.2 EL PUNTO DE VISTA DE LOS PROGRAMADORES DEL SISTEMA

Todos los conceptos que son visibles y accesibles al programador de Fortran tienen su representación y contraparte en términos de instrucciones de máquina y códigos en el programa que resulta de la traducción.

Algunos de ellos resultan familiares, más como producto del condicionamiento que aporta la educación elemental que por el conocimiento de los mecanismos de la traducción. Algunos otros conceptos -posiblemente los más relevantes- resultan desconocidos, no obstante que inciden con mayor intensidad en la eficiencia del programa de aplicación.

Para garantizar su propósito, los compiladores aplican procedimientos sistemáticos y generales para tratar cada aspecto del programa; lo mismo para los casos más complejos y elaborados que para los más simples. Como resultado, en los programas que genera, el compilador incluye muchas computaciones redundantes de las que el usuario no tiene la menor sospecha.

Estas situaciones se presentan lo mismo con el uso de matrices que con el intercambio de datos entre subrutinas o en los elementos del control del flujo del programa. Pero esto no es privativo de Fortran, también se presenta en otros lenguajes como Algol, "C" y Pascal. Lo anterior tiene como consecuencia que los programas producidos por los compiladores son muy grandes y también lentos, aunque la calidad del compilador puede agudizar el problema.

Por eso, los compiladores producen programas que realizan muchos cálculos redundantes, que efectúan un excesivo movimiento de datos entre las diferentes instancias de almacenamiento y que emplean estructuras de control complejas, que en los casos simples -que son la mayoría- producen códigos ineficientes, sin que el usuario tenga conciencia de ello.

6.3 LA NECESIDAD DE FORMAR UN MODELO DE LOS COMPILADORES

El científico no puede pretender que comprende algo, en tanto no se haya formado un modelo de su objeto de estudio, Seymour Papert, creador de LOGO, expone en sus trabajos una filosofía educativa que, al parecer, adquirió en su niñez gracias a la familiaridad que tuvo con el funcionamiento de los relojes. Su interés le llevó a destripar varios de ellos, y el funcionamiento de la maquinaria del reloj fue un problema que para Papert quedó resuelto hasta que concibió un modelo para el mecanismo, que le permitía en su imaginación, jugar personalmente el papel de cada uno de los componentes.

La elaboración de un modelo de esta naturaleza para un compilador, constituye una base sólida cuya utilidad se proyecta hacia trabajos futuros. A partir del modelo, se puede producir un producto de mejor calidad, con un mayor nivel de organización, más libre de errores y más eficiente en el desempeño de sus propósitos. Analizar un programa con el afán de mejorarlo sustancialmente en efecto corresponde con la idea de hacer ingeniería al revés.

Los altos costos, la necesidad de una referencia fiel para el lenguaje y el atractivo de disponer de una base adecuada para el desarrollo de nuevos trabajos, fueron motivos suficientes para realizar esfuerzos orientados a obtener un programa fuente para el compilador a partir de su binario. Y mediante su análisis, formular un modelo para la operación del programa, siempre con la perspectiva de producir un producto mejorado.

El problema de la originalidad del trabajo científico tiene su historia. Uno de los casos más sonados, surgió en torno a la solución de la ecuación cúbica. El conflicto se dió entre un profesor y sus alumnos. Uno de los alumnos reunió todos los elementos del problema que se encontraban aislados y aportó algunos de ellos, publicó el trabajo sin dar al profesor ni a los demás participantes los créditos que esperaban. La molestia, tanto del profesor como de los demás involucrados, fue manifiesta. Al alumno se le calificó de ingrato, él, por su parte, quedó convencido que había salvado aquel valioso material para la posteridad. Sin su trabajo, muchos de aquellos resultados posiblemente se habrían perdido.

El primer compilador de Fortran no se elaboró a partir de nada. En la definición del lenguaje y en la elaboración del compilador se aprovechó la experiencia acumulada por mucha gente en la traducción de fórmulas algebraicas, el indexamiento de elementos de vectores y matrices, en la aritmética de punto flotante y en la conversión de valores y edición de resultados.

Por otro lado, el obtener el programa fuente de un programa complicado -a partir del binario-, analizarlo y documentarlo, puede involucrar tanto o más trabajo del que fue necesario para producirlo. Para realizar el desensamble y el análisis de un programa de la complejidad de un compilador, es necesario el desarrollo de muchos y muy variados accesorios. A la vez que se resuelve el problema de obtener una referencia fiel del lenguaje, se produce también, una infraestructura de programas accesorios abundante y variada y se adquiere una considerable experiencia.

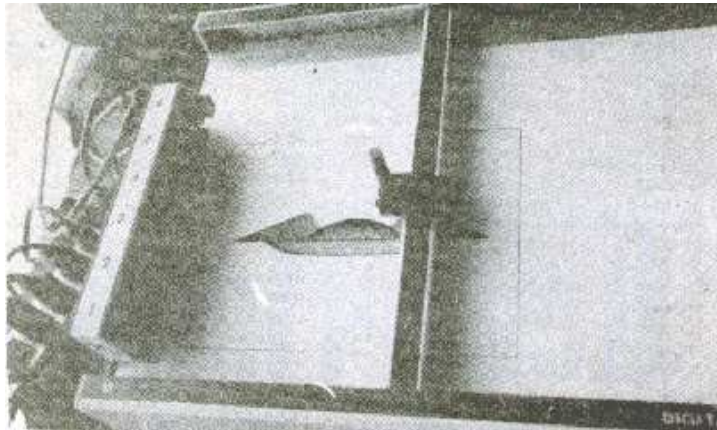
Para realizar un trabajo de esta naturaleza es necesaria una mezcla de capacidad, experiencia, entusiasmo y por encima de todo, un afán por involucrarse y participar del problema.

El resultado puede ser una copia o un original, depende del criterio. No hay muchas cosas nuevas en el mundo, pero algunas son más nuevas que otras. Un ejemplo de esto lo observamos cuando alguien estudia cálculo; comprende el material y produce su propio libro con su interpretación del tema. El contenido no es original, no puede serlo, los teoremas son los mismos que se describen en todos los demás libros, pero su valor se encuentra en el esfuerzo intelectual y en la interpretación que se aporta del tema.

6.4 RELACIÓN DEL COMPILADOR CON EL HARDWARE.

Las microcomputadoras ofrecieron una opción económica para el cálculo numérico, pero adolecieron de la falta de aritmética de punto flotante adecuada.

La evolución del software para estos equipos, específicamente la aparición de F80 -un compilador para Fortran en el 8080- y de las versiones de Basic que incorporaron aritmética de punto flotante parcialmente subsanaron esta deficiencia, aunque su aritmética de punto flotante resultó lenta.



La elaboración de gráficas requiere de una cantidad considerable de aritmética.

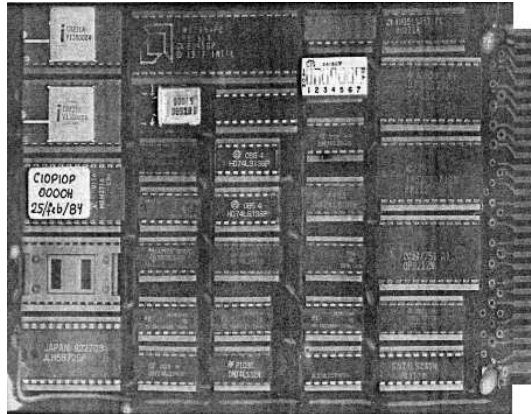
Esta lentitud fue relativa, notoria en contraste con la velocidad de la aritmética de otros equipos. Recordemos que la mayoría de las computadoras actuales incluyen operaciones aritméticas de punto flotante en su repertorio de instrucciones, inclusive algunas de ellas -como la VAX 11/780- ofrecen elementos aritméticos -aceleradores de punto flotante- para mejorar su rendimiento en el aspecto numérico.

Las microcomputadoras carecían de instrucciones para aritmética de punto flotante en sus repertorios de instrucciones, la aritmética de que disponían se realizaba a partir de sus operaciones elementales, que consistían en sumas y restas de números de 8 dígitos binarios. Por ello, a pesar de contar con este tipo de aritmética, su aprovechamiento en el aspecto numérico fue muy limitado.

En estas circunstancias surgieron en el mercado circuitos con el propósito específico de realizar la aritmética de punto flotante. De entre ellos, se destacó el circuito 9511

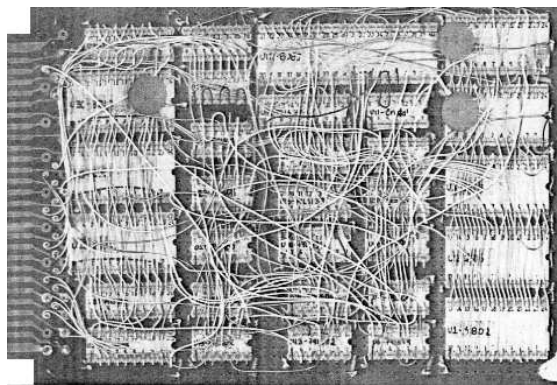
producido por Advanced Micro Devices, posteriormente Intel puso en el mercado su 8231, un circuito equivalente.

Con este tipo de circuitos, fue posible construir elementos para las microcomputadoras que se encargaron de la aritmética de punto flotante, con ellos se incrementó el rendimiento de las microcomputadoras. Estos módulos operaron en forma periférica y su capacidad real se aprovechó cuando los compiladores de los lenguajes numéricos se transformaron para producir programas que utilizarán las ventajas de los elementos aritméticos.



Elemento aritmético elaborado en la Universidad Autónoma de Puebla.

En el Departamento de Aplicación de Microcomputadoras del Instituto de Ciencias de la Universidad se diseñaron y contruyeron varios elementos aritméticos. El pleno aprovechamiento de sus capacidades de cálculo fue posible gracias a la disponibilidad de los programas fuentes para el compilador de Fortran. Las modificaciones de estos programas -generados en el propio Instituto- y el concurso de los elementos aritméticos mejoraron sensiblemente los tiempos de algunas de las aplicaciones. Es el caso del paquete PLOT, que para la preparación de gráficas tridimensionales opacas y mapas de contornos, emplea una cantidad considerable de aritmética de punto flotante.



Parte posterior del elemento aritmético.

Los elementos aritméticos se prepararon en forma modular para que su integración en las microcomputadoras de la Universidad fuera más simple. Se elaboraron elementos

para acoplarse tanto en el bus STD como en el S-100 y varios de ellos se encuentran en operación para el servicio de la Universidad desde hace varios años.

Algunos de los elementos aritméticos que se han diseñado en la Universidad no tienen equivalente en el mercado. Es el caso de los diseños del procesador aritmético que desarrolló el alumno Daniel Angel Tenorio Téllez como tema para su servicio social y trabajo de tesis profesional. Este elemento aritmético incorpora dos circuitos aritméticos 9511 y puede realizar operaciones de punto flotante con números complejos. El diseño del elemento aritmético de referencia incluye un procesador de control que le permite utilizar los dos circuitos 9511 para calcular simultáneamente las componentes real e imaginaria de las operaciones aritméticas complejas.

Con este tipo de elementos aritméticos, la velocidad de la aritmética de punto flotante de las microcomputadoras se ha incrementado en 4 y hasta en 12 veces (cuando se trata de aritmética compleja).

Con el trabajo realizado, la experiencia que se adquirió y a la luz de los resultados obtenidos, se abren amplias perspectivas para el desarrollo de nuevos proyectos, en la dirección de mejores compiladores, ahora también apoyados por los resultados y experiencias obtenidos en el desarrollo de otros proyectos y también en la dirección del diseño de nuevos elementos aritméticos, con los nuevos circuitos más veloces que ya han aparecido en el mercado.

COMPUTADORAS Y PROCESOS SIMBÓLICOS

7.1 ¿Las computadoras pueden pensar?

La interrogante acerca de si las computadoras pueden pensar, constituye uno de los temas más apasionantes de la ciencia cuyo interés crece en la medida en que se registran mayores avances en el perfeccionamiento de los instrumentos de la computación.



M. C. Escher.

Existe mucha polémica al respecto, en buena medida la dificultad para arribar a una conclusión depende del grado de claridad que se tenga en relación con los procesos del pensamiento. En la búsqueda de esta claridad, la humanidad desde tiempos inmemoriales ha destinado no pocos esfuerzos.

Con los trabajos de los griegos clásicos, no sólo se busca establecer los fundamentos de la geometría (Euclides), sino también poner de relieve las formas del pensamiento (Aristóteles y Platón).

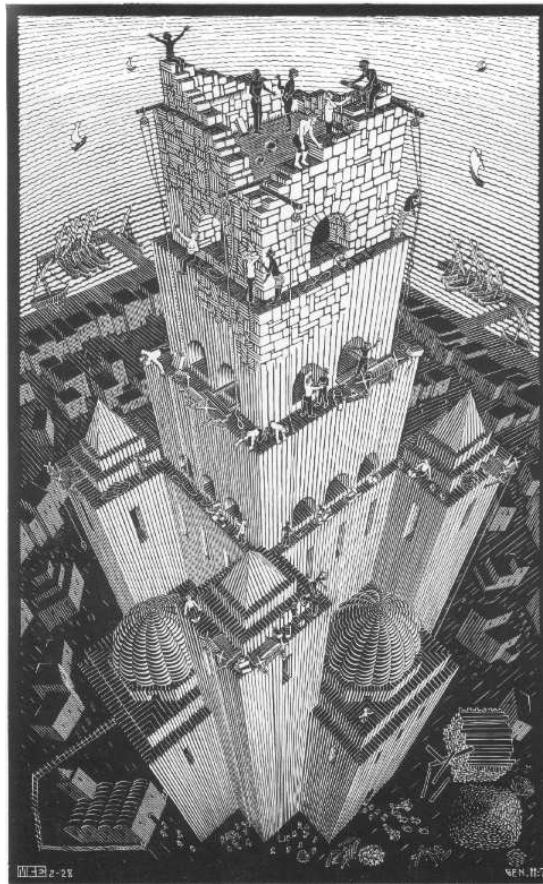
Las preocupaciones que sobre el pensamiento ocuparon a los antiguos, se extienden a lo largo de la historia hasta nuestros días.

El propio Rene Descartes y Gottfried W. Leibniz mostraron el poder del álgebra para simbolizar e incluso mecanizar la geometría.

Más tarde George Boole y Augustus DeMorgan en sendos trabajos aportaron nuevos simbolismos algebraicos, ahora aplicables a los procedimientos lógicos. En particular, Boole estaba convencido de que la simbolización del lenguaje vigorizaría la lógica.

Como contribución a esta empresa, otros pensadores intentaron unificar las formas del razonamiento con la lógica y los procesos computacionales, sin intentar hacer prevalecer a ninguno de ellos.

Así tenemos que Bertrand Russell y Alfred North Whitehead, en su obra monumental Principia Mathematica recopilaron y examinaron exhaustivamente las matemáticas en un intento por articularla bajo el principio del álgebra de Boole.



M. C. Escher.

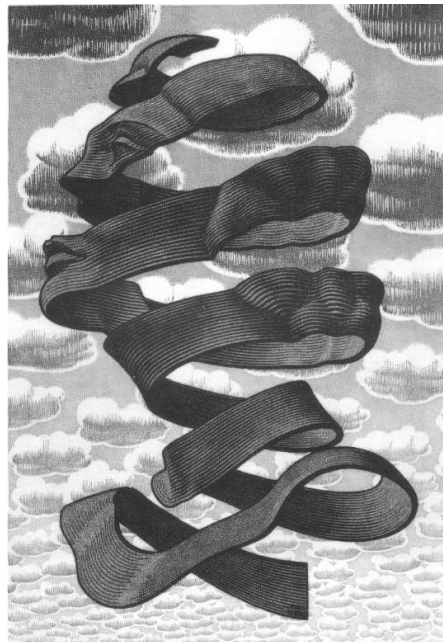
Así, como los griegos clásicos lo hicieron con la geometría, Russell y Whitehead intentaron sustentar la matemática sobre principios matemáticos, y de esta manera, imprimirle una estructura de su propia naturaleza. Aunque hoy sabemos que esto no es posible, sus trabajos permitieron identificar que una parte considerable de las matemáticas se pueden construir a partir del manejo mecánico de los símbolos.

Por su parte, David Hilbert -quién fue el promotor de la idea de buscar la coherencia de las matemáticas- en 1926 establecía que “en esencia, el objeto del pensamiento matemático, son los símbolos en sí mismos. Estos símbolos no pueden ser considerados más tiempo como simples idealizaciones de los objetos físicos”.

El continuo desarrollo de estas ideas permitió a A. Church, elaborar su teoría de las funciones recursivas, en la que, reformuló el concepto de algoritmo; a E. L. Post por su sistema de producciones. En tanto que Alan M. Turing elaboraba sus modelos matemáticos recurriendo a analogías mecánicas. Markov formulaba la teoría de los algoritmos que llevan su nombre, y Kurt Gödel su esquema aritmético acerca del problema de la coherencia de los sistemas formales.

Cada uno de ellos, no obstante haber escogido perspectivas diferentes en sus análisis, arribó a resultados equivalentes.

1. Definieron el concepto de Algoritmo y produjeron esquemas descriptivos de su naturaleza.
2. Contribuyeron en el esclarecimiento de la naturaleza de los procesos computables.
3. Establecieron que existe la imposibilidad de determinar por medio de un proceso, la viabilidad de otro.
4. Concluyeron que era posible la construcción de procedimientos de propósitos universales, aplicables a problemas específicos con la sola descripción apropiada.



M. C. Escher.

7.2 La Computadora y los símbolos

Al mismo tiempo que se elaboraban estas construcciones formales, se desarrollaba también la tecnología electrónica, de manera que al confluir ambos procesos, se posibilitó la construcción de las modernas computadoras.

Aunque originalmente las primeras computadoras, se construyeron para realizar operaciones aritméticas en gran escala, muy pronto fue evidente que también se podían utilizar para otro tipo de manipulación de símbolos, porque ciertamente, los números son también símbolos pero no son todos los símbolos; existen otros -quizá más importantes- de diferente naturaleza.

En efecto, muy rápidamente la computadora comenzó a ser utilizada en el manejo de información no numérica; principalmente en el texto escrito y los símbolos de la matemática no aritmética.

Desde las primeras aplicaciones no numéricas, fue posible obtener no sólo los resultados inicialmente esperados, sino también la consolidación de ciertos procedimientos de orden general. Tal es el caso de IPL, un conjunto de subrutinas elaboradas por Newell, Simon y Shaw en el Instituto Tecnológico de Carnegie orientadas a organizar datos con el empleo de estructuras distintas a las tablas, en las que las relaciones que guardan con los datos entre sí, son tan importantes como los datos mismos.

De igual forma surge FLPL (Fortran List Processing Language), como una colección de subrutinas -construidas en Fortran- que permitieron organizar información en forma de estructuras ramificadas. Aunque debe decirse que su éxito fue limitado.

No tardaron en surgir verdaderos lenguajes de programación, entre los cuales merecen mención especial COMIT, desarrollado por V. Yngre y Lisp (List Processor), desarrollado por John McCarty - y sus alumnos-, en el Instituto Tecnológico de Massachusetts al final de los 50's. Lisp se convertiría por antonomasia, en el lenguaje de las aplicaciones en inteligencia artificial.

Lisp reunió en su definición pocos conceptos, pero todos ellos de importancia capital: estructuras ramificadas (árboles binarios o listas), uso de texto escrito, recursividad y el concepto de máquina universal. Como puede apreciarse, resulta ser un lenguaje compacto, consistente consigo mismo.

Lisp incluyó la recursividad, posiblemente porque su autor fue estudiante de la Universidad de Princeton en donde las ideas de Church, Post y Turing tenían considerable arraigo y difusión.

Fue también el primer lenguaje que comprendía el concepto de función universal.

No obstante que fue el lenguaje que mejor potenció la computación simbólica, dado que su diseño reunió la elegancia, la flexibilidad y la sencillez, sufrió de algunas deficiencias de índole pragmática:

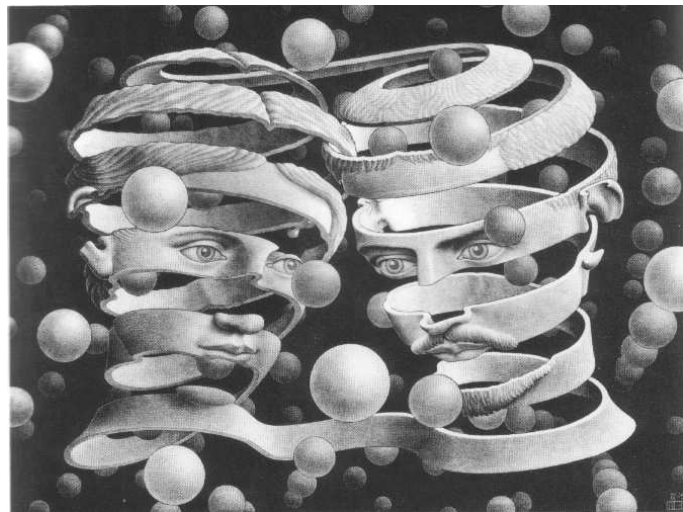
1. La recursividad inevitable en Lisp, ya que forma parte del lenguaje exige la conservación de resultados parciales aún cuando no se usen, por ello, los programas en Lisp requieren grandes capacidades de memoria y también prolongados tiempos de ejecución. No es propiamente una deficiencia de lenguaje, sino una característica de la naturaleza de los problemas a los que se aplica.

2. Sus deficiencias para cálculo numérico fueron tan agudas, porque carecía de aritmética elemental de cualquier tipo. Esta carencia es más imputable a la pereza de los autores que a una imposibilidad conceptual, aunque no faltó quién intentara -con poco éxito- subsanarla.

La programación involucrada no es difícil, pero la definición misma de un número es más compleja y profunda que la de un conjunto.

Para un manejo adecuado de la aritmética, los lenguajes deben distinguir a los números de entre los otros símbolos. La comprensión del concepto número, requiere de sólidos fundamentos en el lenguaje y su acercamiento conceptual, exige elaborados esquemas matemáticos para su construcción formal. Y esto lo deben ofrecer de manera que no importunen al usuario con formas grotescas para indicar la aritmética.

3. Otras quejas menores, producto de las características cosméticas del lenguaje, entre las que se pueden citar el empleo de la notación funcional y el abundante uso de paréntesis.



7.3 El lenguaje Convert, la programación a base de ejemplos. Una aportación nacional

Durante un curso que impartió el Dr. Harold V. McIntosh en el CeNaC (Centro Nacional de Cálculo del IPN) en 1965, definió el lenguaje de programación Convert y apoyado por algunos de sus alumnos escribió en Lisp el primer interpretador para el lenguaje; se utilizaron para el efecto, las facilidades disponibles por vía telefónica del sistema SDCQ32 (System Development Corporation) instalado en Santa Mónica, California. Las principales descripciones de los primeros resultados sobre Convert se publicaron en la tesis profesional de Adolfo Guzmán y en un artículo en la revista ACM (Association for Computing Machinery).

Desafortunadamente, las consecuencias del doble proceso de interpretación involucrados en el uso de Convert (ya que Lisp generalmente era interpretado) no se hicieron esperar. El tiempo de ejecución de las primeras aplicaciones de Convert se vieron multiplicados por un factor de mil. Para mejorar el rendimiento del lenguaje se planteó la necesidad de compilarlo para evitar el doble proceso e interpretación. Uno de estos intentos se esbozó y parcialmente desarrolló en la tesis profesional de Raymundo Segovia.

Los principios de programación del lenguaje Convert se pueden resumir en la programación descrita con ejemplos, esto es: “si ves algo de esta naturaleza, identifica los elementos que los conforman y sustitúyelos de esta otra manera”. En su definición concluyeron dos deseos:

1. Simplificar y sistematizar algunos programas para el reconocimiento de patrones previamente elaborados en Lisp, entre los que se encontraban aquellos para el cálculo de derivadas simbólicas de formas algebraicas.
2. Realizar en una forma adecuada las transformaciones de Post para simplificar los métodos de reducción y simplificación de las expresiones algebraicas.

Un programa en Convert es una colección de reglas que se confrontan con un texto hasta que alguna de ellas se le puede aplicar. Las reglas pueden ser terminales o iterativas, estas últimas, reinician la confrontación del texto.

Con patrones y esqueletos se forman las reglas de transformación en Convert. Una regla consiste de un patrón y su esqueleto. Al confrontar la regla con el texto, se intenta el ajuste del patrón, si la correspondencia se verifica, el texto se transforma de acuerdo con el esqueleto.

Los patrones se clasifican en simples o compuestos. Los patrones simples pueden ser constantes o variables. Los primeros son secuencias de símbolos que se indican textualmente, los variables deben marcarse para distinguirlos entre los constantes.

La manera de marcar los patrones variables, depende del medio en que se describe el programa. En forma manual se puede emplear letra cursiva, o el recurso del subrayado. Pero en una computadora es más conveniente el uso de símbolos especiales para este propósito. En Convert se emplean los paréntesis angulares para distinguir a los patrones variables de los constantes.

Los patrones compuestos se obtienen a partir de otros patrones por medio de la composición (concatenación), de las combinaciones lógicas y del uso de funciones, en este acto, con el empleo de una notación que utiliza paréntesis redondos para indicar el nombre de la función y sus argumentos.

Un texto corresponde con un patrón cuando los elementos de ajuste incluidos en el patrón permiten establecer la correspondencia del patrón con el texto. En este proceso, se utiliza la flexibilidad que aportan los elementos variables del patrón, ya que con los elementos constantes, la correspondencia debe ser literal.

Cuando se establece la correspondencia del texto con el patrón, los elementos variables del patrón adquieren una definición que se emplea para transformar el texto original, sustituyendo las definiciones de los elementos variables del patrón en el esqueleto asociado.

Un esqueleto puede ser simple o compuesto. Los esqueletos simples son constantes o variables, en este último caso, la referencia es al valor de la variable. Como en el caso de los patrones, los esqueletos compuestos, se obtienen a partir de otros por concatenación y con el uso de funciones que se denotan en la misma forma con el uso de paréntesis redondos.

En Convert, el usuario puede definir patrones y esqueletos complejos y referirlos por medio de un identificador para abreviar la descripción de las reglas. En realidad, un programa consiste de cuatro elementos, la definición de PATRONES, ESQUELETOS, VARIABLES y REGLAS.

La forma de notación para las diferentes instancias del lenguaje, variables, funciones y combinaciones lógicas son algunos de los elementos que le imprimen al programa en Convert su aspecto característico.

Los lenguajes orientados al manejo de símbolos tradicionalmente han adolecido de programas para realizar aritmética y para manejar arreglos, Convert no fue tampoco la excepción. No obstante ofrece algunas ventajas para la manipulación de símbolos. Algunas de sus principales aplicaciones se citan a continuación:

1. Modificación selectiva de un texto, particularmente en los casos en que la modificación es más complicada que una simple sustitución.
2. Búsqueda de información en archivos. Para la localización de palabras clave y para la elaboración de índices por referencias para las palabras clave.
3. Elaboración de ensambladores y compiladores. Y no sólo para los tipos convencionales, sino para la traducción de programas de un equipo a otro. También se ha utilizado con éxito en la elaboración de desensambladores y descompiladores.
4. Generación de programas complicados (en ensambladores, Fortran u otros lenguajes) a partir de protocolos descriptivos. Esta posibilidad ofrece una nueva dimensión para la elaboración de programas. Sin con Convert no se puede realizar un trabajo de manera eficiente, con él se pueden construir programas más eficientes y de esta manera reducir el tiempo que de otra manera resultaría ser desmedido.
5. Cálculo simbólico, tales como el cálculo de derivadas simbólicas y el análisis de conjuntos.

7.4 REC, Compilador de Expresiones Regulares

Este lenguaje, también diseñado por el Dr. McIntosh, se originó a partir de una solicitud muy explícita: elaborar una prueba de aceptación para una minicomputadora PDP-8 del Centro de Cálculo Electrónico de la UNAM.

El diseño de este lenguaje de programación fue motivado por el deseo de incorporar la iteración en Lisp en una manera axiomática, reformulando el lenguaje y rescatando sus características más importantes.

En la definición de este lenguaje, el autor aprovechó las experiencias adquiridas en los esfuerzos realizados para extender las características del lenguaje Lisp, particularmente los encaminados a dotarle de una aritmética adecuada y de funciones accesorias para manejo de textos.

Para la definición del lenguaje, el autor identificó que la gran mayoría de los procedimientos computacionales consisten de relativamente pocos elementos: una secuencia de operaciones, la selección con base en cálculos intermedios y la repetición de los procedimientos (iteración); y por ello, redujo los elementos de control a cuatro símbolos:

(paréntesis izquierdo
: dos puntos
; punto y coma
) paréntesis derecho.

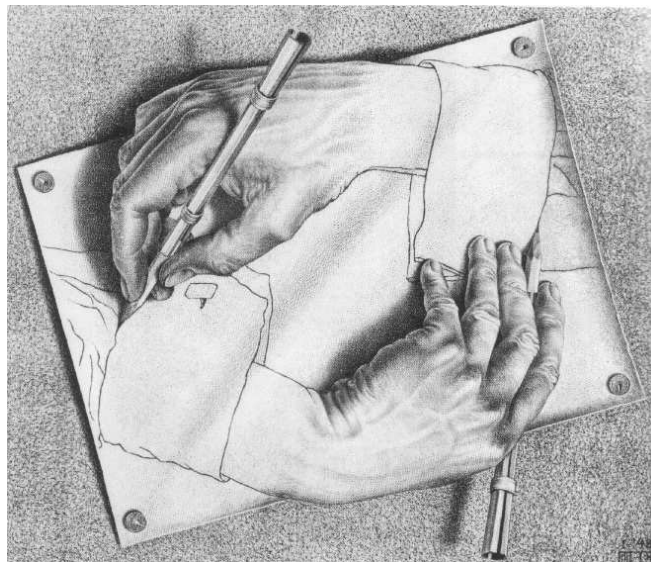
El lenguaje no incluye transferencias incondicionales -ni las tendrá-, no utiliza etiquetas y por ello, se anticipó en una buena medida al concepto de la programación estructurada.

En REC se destina un símbolo para denotar cualquier operación, y se le atribuye un valor lógico que interviene en el control del programa. La secuencia de operaciones se indica concatenando los símbolos correspondientes. Su sintaxis utiliza paréntesis redondos para agrupar los símbolos.

La toma de alternativas en REC, se realiza de acuerdo con el valor de la operación, el valor verdad se emplea para continuar la secuencia y el falso, para romperla y elegir otra opción. Con “;” se separan las secuencias alternativas y con “:” se indica la repetición de las operaciones desde el principio.

Adicionalmente, permite la definición y el uso de subrutinas. El uso de las subrutinas hacen de REC un lenguaje recursivo y su definición le otorga el carácter iterativo.

Su estructura de control es, como se puede apreciar, fundamental y económica.



M. C. Escher.

Las primeras aplicaciones matizaron el desarrollo del lenguaje enfílándolo en la dirección del manejo de símbolos, aunque otras direcciones son también posibles.

La versión del lenguaje para la edición de textos tuvo un amplio uso en diferentes aplicaciones, tanto administrativas como científicas y pedagógicas. Principalmente se utiliza para elaborar programas de servicio inclusive, sistemas operativos completos - y generadores de programas de aplicación-.

Después de 20 años de desarrollo del lenguaje, por parte de su autor y de una gran cantidad de sus alumnos entre los que destaca Gerardo Cisneros por sus aportaciones, REC encontró una nueva aplicación, que consistió en ofrecer una opción para compilar Convert.

De esta manera, la exasperante lentitud de los programas en Convert ya no presentó una barrera infranqueable para utilizarlo en múltiples aplicaciones no numéricas.

Y con respecto a la pregunta original, podemos concluir que si bien en los procesos del pensamiento los símbolos y su manejo juegan un papel de primer orden y que las computadoras participan del manejo simbólico desde hace tiempo y cada día lo hacen con mayor intensidad; también es cierto que -como ya lo había percibido la legendaria Lady Lovelace- en el siglo pasado “sólo pueden realizar aquello que se es capaz de describirles como hacerlo”.

Por lo tanto, cuando las computadoras piensen, -si acaso llegan algún día a hacerlo- será porque el hombre ha desentrañado los procesos del pensamiento y ha sido capaz de describirle a un mecanismo la manera de realizarlos.