

“Ejemplos de la utilización de la Arqueología Informática en temas de hardware y software”

Área de Conocimiento: Computación Educativa

Isaías Pérez Pérez¹, Citlali Anahí Monzalvo López²

¹ Universidad Autónoma del Estado de Hidalgo.
Instituto de Ciencias Básicas e Ingeniería
Carr. a Tulancingo s/n. Mineral de la Reforma, Hidalgo. México
e-mail: isaiasp@uaeh.edu.mx, isaiaspp7@hotmail.com

² Universidad Autónoma del Estado de Hidalgo
Instituto de Ciencias Básicas e Ingeniería
Carr. a Tulancingo s/n. Mineral de la Reforma, Hidalgo. México
e-mail: lex_any@hotmail.com

Resumen: *La presente investigación muestra algunos de los resultados conceptuales generados por la utilización del método de la Arqueología Informática, sobre algunos temas conocidos de hardware y software. Esta naciente disciplina, que surgió inicialmente de actividades de coleccionismo y divulgación de ciertos grupos de usuarios, sobre equipos de cómputo y software antiguo, ha empezado a demostrar su potencial implícito, ya que el conocimiento que rescata del pasado ha servido como inspiración para el desarrollo de hardware y software actual.*

Palabras clave: *Arqueología Informática o Computacional, Retroinformática, Método.*

Introducción

Dentro del enfoque histórico, el pasado presenta los personajes, sus contribuciones y los hechos que acontecieron y como todos estos fueron conformando el panorama del presente. Pero la influencia del pasado no termina ahí; a su vez, tanto el pasado como el presente determinan las directrices de los eventos que se darán en el futuro; en el mundo de las computadoras, esta regla aplica de la misma manera.

Para abordar el estudio del pasado computacional, ha ido surgiendo una naciente disciplina denominada “Arqueología Informática” (*también conocida como Arqueología Computacional o Retroinformática*), que es el estudio del hardware (computadoras, microprocesadores, periféricos, etc.), software (lenguajes de programación, sistemas operativos, etc.), tecnología de comunicaciones y redes de equipos de cómputo y todo lo relacionado con estos tres elementos: creadores, época en que se desarrollo o construyo el equipo o software, razones de su surgimiento, ventajas y desventajas que ofrecían, entre otras muchas cosas (Pérez Pérez y Monzalvo López, 2011).

Estado del arte y problemática presentada

Aunque se utiliza predominantemente una descripción histórica para abordar el estudio de un objeto de hardware o software específico antiguo, la Arqueología Informática no trata solo del simple relato de la historia, busca recopilar, analizar y entender el conocimiento que encerraban los diseños y su funcionamiento, disponibles o no hoy en día, de equipos de cómputo, software y redes de computadoras que existieron, con el fin de proveer la mayor cantidad de información al respecto; de esta manera se puede lograr reconstruir el conocimiento que encerraban estos diseños.

La Arqueología Informática busca las respuestas a las preguntas: ¿por qué?, ¿para qué?, ¿cómo o con qué?, ¿quién?, ¿cuándo y dónde?, fue creado el hardware o software en estudio. Al recabar esta información, ésta se convierte en un bagaje cultural extra para los interesados, además de que sirve como una forma de enseñanza de los temas tratados, generando esto una extensión del conocimiento que se posee sobre el tema. Además, al obtener las respuestas a las anteriores preguntas, es posible explicar y entender las motivaciones que se tuvieron para diseñar y construir el hardware y/o software en cuestión, dando paso a la posibilidad de generar posibles actualizaciones y hasta concebir nuevos diseños inspirados en dicha información (Pérez Pérez y Monzalvo López, 2011).

Para ejemplificar la utilidad de la Retroinformática, en la presente investigación se aplicará el método relativo a ésta, para generar un reporte escrito de tipo histórico-teórico, sobre los temas siguientes: 1) la evolución de la computadora mecánica a la electrónica; 2) la evolución de los lenguajes Pascal, Ada, Modula-2 y C; y 3) la evolución de la idea de usar el sistema binario en las computadoras actuales; dicho reporte puede ser considerado como un documento que aumentara, inicialmente, la cultura informática del lector.

Método y técnicas utilizadas

El método a utilizar es el que propone Pérez Pérez y Monzalvo López (2011), para llevar a cabo la investigación, el cual consiste en cuatro pasos:

- 1) Definir el objeto de estudio lo más concretamente posible (equipo de cómputo o software, como sistemas operativos, utilerías, lenguajes de programación o aplicaciones específicas).
- 2) Buscar, recopilar, analizar y estructurar, toda la información relacionada con el objeto de estudio seleccionado: su historia, creadores o desarrolladores, especificaciones de su diseño y funcionamiento, elementos de hardware o

software relacionados con él; en general, toda la información que pueda proporcionar referencias sobre el objeto de estudio.

- 3) Definir la interrelación y secuencia de los datos, obtenidos de la información recabada sobre el diseño, construcción, estructura y/o funcionamiento del objeto de estudio en cuestión.
- 4) Redactar el documento correspondiente y/o los diagramas correspondientes, para su posterior uso como fuente de información de consulta, que entre otros usos, podrá servir como soporte conceptual para desarrollos tecnológicos actuales.

Resultados experimentales

De cada uno de los temas seleccionados, en primer lugar, es necesario definir el objeto de estudio. A continuación se llevaron a cabo los pasos 2 y 3 del método de la Arqueología Informática de Pérez Pérez y Monzalvo López (2011). Finalmente, el resultado del cuarto paso, dio como producto los siguientes tres reportes:

a) Arqueología Informática en el hardware: la evolución de la computadora mecánica a la electrónica

Un concepto clave de la computación moderna es definitivamente la “Máquina Analítica” (*Analytical Engine*) de Charles Babbage, la cual hubiera sido la primera computadora de uso general de la historia si se hubiera construido. Babbage empezó a trabajar en este proyecto en 1834 y continuó haciéndolo durante toda su vida. Su modelo fue refinado muchas veces, y a lo largo de este proceso, Babbage tuvo muchas ideas visionarias sobre las computadoras (Quantum Networks, 2011).

¿Era realmente una computadora la Máquina Analítica?. En la actualidad, el concepto de computadora se define como una máquina programable para procesar información integrada de unidad de entrada, unidad de salida, unidad de control, procesador y memoria. La máquina analítica de Babbage era una máquina para procesar información constituida por las mismas partes que ahora forman una computadora. Sin embargo, no era capaz de almacenar un programa y ejecutarlo sin la intervención de un operario, característica fundamental en el actual concepto de computadora. Fue necesario que transcurrieran varias décadas para que, con el invento de la electricidad y la electrónica, se propiciaran las facilidades técnicas que permitieron a la computadora tener un programa almacenado (Nuncio Limón, 1991).

La Máquina Analítica fue concebida como una calculadora capaz de realizar cualquier tipo de cálculo de manera digital, pero no pudo ser concluida debido a que su tecnología era muy adelantada para la época y nunca se pudieron construir las sofisticadas piezas que se diseñaban para ella. Si la tecnología de esa época hubiera

estado al nivel que requería el diseño, posiblemente desde entonces se contaría con la primera computadora digital (Ferreira Cortés, 2006).

El proyecto de Babbage se adelantó considerablemente a su época y la idea en que se sustentó tenía plena validez. Además de crear una máquina, el científico inglés pensó la forma de hacerla funcionar mediante un programa o conjunto de instrucciones que pudieran modificarse para lograr que la máquina no sólo sirviera para un propósito, sino que ejecutara y resolviera cualquier problema. Ésa sigue siendo en esencia, la idea fundamental de la computadora y del software (Nuncio Limón, 1991).

Definitivamente, la Máquina Analítica de Babbage fue una idea que se anticipó 100 años a su época. Sus conceptos quedaron ocultos con el polvo de los años y los científicos que un siglo después lograron construir y hacer funcionar la computadora, no conocieron con detalle estas ideas y sólo unos pocos se inspiraron en ellas para realizar sus proyectos (Nuncio Limón, 1991); tal es el caso de Howard H. Aiken.

En 1937, Aiken, en la Universidad de Harvard, desarrolla junto con un equipo de científicos de su departamento e ingenieros de IBM, la IBM ASCC (*Automatic Sequence Controlled Calculator / Calculadora Automática de Secuencia Controlada*) o Harvard Mark-I, la cual se basaba en los planes originales de Charles Babbage para la máquina analítica de cien años atrás, y la propuesta trataba de construir el diseño de Babbage, usando relés en lugar de engranajes. (Helmut, 2010). El resultado fue la construcción de una calculadora numérica basada en el uso de relés electromagnéticos, ruedas dentadas y embragues electromecánicos, configurando la primera computadora electromecánica. Esta computadora se terminó de construir en 1944.

Esta máquina fue la primera computadora electromecánica que se construyó y que funcionó, aunque se utilizó poco tiempo, pues la aparición de las computadoras electrónicas hizo que las de este tipo cayeran rápidamente en desuso (Alcalde y García, 1994). Podemos decir que el Mark-I sirvió de transición en la tecnología de las computadoras, entre las mecánicas (Babbage) y las enteramente electrónicas (ENIAC) (Garner, 1984).

Por otra parte, en 1941, en Inglaterra, se construyó el ACE (*Automatic Computing Engine / Motor de Cálculo Automático*) que fue la primera computadora electrónica de programa almacenado, diseñada por Alan M. Turing, por la invitación de John R. Womersley, superintendente de la División de Matemáticas del National Physical Laboratory (NPL). El uso de la palabra *motor* fue en homenaje a Charles Babbage y su máquina diferencial y la máquina analítica. El diseño técnico de Turing de su propuesta de "*calculadora electrónica*" fue el producto de su trabajo teórico en 1936, titulado "Sobre los números computables" y su experiencia durante la guerra en Bletchley Park, donde el ordenador Colossus había tenido éxito en romper los códigos militares alemanes. En su artículo de 1936, Turing describió su idea como una "máquina de computación universal", pero que ahora se conoce como la máquina universal de Turing (WIKIPEDIA, 2011). Turing fue el primero en comprender la naturaleza universal de una computadora digital, alcanzando y aún superando las ideas de Charles Babbage (Nuncio Limón, 1991). Hoy se sabe que los ordenadores actuales son realizaciones de las ideas de John Von Neumann y de Alan Turing (Castelazo, 2010).

Charles Babbage murió en 1871, pero sus aportaciones serían realmente apreciadas sólo hasta que las primeras computadoras digitales fueron construidas, a mediados del siglo XX. Sus experimentos dejarían huella profunda en el trabajo sobre autómatas del español Leonardo Torres de Quevedo a principios de este siglo; posiblemente la idea de Herman Hollerith de usar tarjetas perforadas fue derivada por la sugerencia de Babbage, y se ha llegado a especular que la Máquina Analítica pudo haber sido incluso la fuente principal de inspiración del modelo teórico de la computadora moderna, desarrollado por el matemático Alan M. Turing y que hoy se conoce como "máquina de Turing". Con tan convincente evidencia de la importancia de sus ideas, tal vez no importe tanto que Babbage no haya logrado construir sus máquinas después de todo, pues sus aportaciones resultaron igualmente significativas de cualquier forma, dándole un sitio privilegiado en la historia de la informática, como el padre de la computación moderna (Quantum Networks, 2011).

b) Arqueología Informática en el software: el surgimiento de los lenguajes Pascal, Ada, Modula-2 y C.

Desde el momento de su nacimiento, la informática ha experimentado una continua evolución, y naturalmente los lenguajes de programación no han quedado atrás (Biblioteca de informática, 1993). Algunos ejemplos son la evolución de dos lenguajes de programación muy famosos: Pascal y el lenguaje C.

El Pascal fue diseñado a principios de los 70's como un lenguaje para enseñar los principios de la ciencia de las computadoras y el diseño de algoritmos. Rápidamente Pascal se convirtió en el principal vehículo de enseñanza en la mayoría de los libros de texto de la época sobre ciencia de las computadoras. Fue moderadamente usado como un lenguaje de programación práctico en aplicaciones matemáticas, de procesamiento práctico en aplicaciones matemáticas, de procesamiento de datos y de inteligencia artificial (Tucker, 1987).

La historia del Pascal estaría incompleta sin trazar primero la historia del Algol (Algorithmic Language / Lenguaje Algorítmico), del cual Pascal fue su evolución. La historia del Algol comenzó en 1958, cuando un comité de representantes del GAMM (organización europea de científicos en informática) y de la ACM (organización norteamericana análoga), se reunieron en Zurich, produciéndose un informe preliminar del IAL (International Algebraic Language / Lenguaje Algebraico Internacional), el cual se conoció más tarde como Algol 58. Este lenguaje atrajo mucho interés y fue implementado sobre varias computadoras. Los representantes europeos y estadounidenses se reunieron de nuevo en París en 1960, para considerar una versión completamente nueva de este lenguaje, la cual fue conocida como Algol 60. En 1962, en una reunión en Roma, los representantes revisaron y clarificaron la definición del Algol 60 y se escribió el "Revised Report on the Algorithmic Language Algol 60". Este informe fue aceptado como la definición del Algol para sus distintas implementaciones a lo largo de los años 60's (Tucker, 1987). Este lenguaje era más sofisticado e influyó en gran medida el diseño de los futuros lenguajes de programación. Sus autores pusieron mucha atención en la regularidad de la sintaxis, estructura modular y

otras características asociadas con los lenguajes estructurados de alto nivel (Pappas y Murray, 1994). Su definición rigurosa marcó nuevos estándares para el diseño e implementación de lenguajes de programación (Tucker, 1987).

Durante este periodo, Algol fue extremadamente popular entre los científicos informáticos, particularmente en Europa. Lamentablemente, el Algol 60 nunca cuajó realmente en Estados Unidos, debido quizás a su abstracción y generalidad (Pappas y Murray, 1994). Aunque la mayoría de los fabricantes americanos dieron sólo un soporte limitado al Algol, el interés en este lenguaje prosperó a pesar de todo. Algol se convirtió en un lenguaje universal para la definición de algoritmos publicados en revistas, en aquella época (Tucker, 1987).

La principal aportación de Algol fue la de poner en práctica la noción de programación estructurada. Su principio es la compartimentación del texto de los programas en subconjuntos que corresponden a una división lógica del problema tratado. Así, el programa se divide en bloques, cada uno de los cuales puede contener otros. Toda variable o todo procedimiento debe declararse a la entrada de un bloque y sólo puede asignarse en el interior de éste. Al dividir así en compartimentos los programas, se disminuyen las posibilidades de que un error en una región del programa repercuta en otras partes (Biblioteca de informática, 1993).

Más adelante, el propio grupo de trabajo de Algol desarrolló una extensión más rica y elaborada, conocida como Algol 68. Aunque las primeras versiones del Algol fueron lenguajes de programación de tipo científico, Algol 68 fue intencionadamente un lenguaje de propósito general con aplicaciones en un amplio rango de intereses. El "Report on the Algorithmic Language Algol 68" fue un documento largo y complejo, que describía un lenguaje enorme. Los implementadores tuvieron dificultad en desarrollar compiladores para el Algol 68 y hubo discrepancia entre los miembros del grupo de trabajo, sobre la posibilidad de que este nuevo lenguaje fuera demasiado ambicioso para ser práctico (Tucker, 1987).

Por su parte, Niklaus Wirth, un miembro original del comité de trabajo del proyecto Algol, desarrolló un sucesor para Algol 60, llamado Algol-W. Durante los siguientes cuatro años, con la ayuda de tres asistentes, desarrolló un sucesor para ese lenguaje, que llegó a ser conocido como Pascal, llamado así en honor a Blaise Pascal. Pascal es en muchos sentidos una versión elegante de Algol 60; es al mismo tiempo hermoso y práctico (Appleby y Vandekopple, 1998).

A pesar de sus fuertes mejoras sobre Algol (especialmente en el área de entrada-salida, archivos, registros, gestión dinámica de memoria y estructuras de control), Pascal también ha sido cuestionado por sus deficiencias. Debido a esto se propusieron sucesores en los lenguajes Ada y Modula-2. El último, diseñado por el propio Wirth, apareció como una promesa considerable en el área de la programación de sistemas.

En conclusión, se puede decir que así como Algol inspiró el diseño e implementación de muchos lenguajes de programación durante los 60's; también el Pascal lo hizo en los

años 70's con el lenguaje Ada y en los 80's con Modula-2, que fueron intentos de extensiones y generalizaciones del lenguaje Pascal (Tucker, 1987).

Por otra parte, la historia del lenguaje C comenzó también con Algol 60. Mas adelante, en 1963, en la Universidad de Cambridge como en la Universidad de Londres, se desarrollo el lenguaje CPL (Lenguaje de Programación Combinado). Los inventores de CPL intentaron acercar el elevado nivel de Algol a las realidades de una computadora de esa época; pero como el Algol, el CPL era muy grande. Esto hacía que el lenguaje fuera duro de aprender y difícil de implementar y esto explicaría su caída eventual.

Tratando de aferrarse a lo mejor que podía ofrecer el CPL, Martin Richards de la Universidad de Cambridge, en 1967, desarrollo el BCPL (Lenguaje de Programación Combinado Básico), que intentaba ser una versión reducida de CPL con sus características básicas. Poco tiempo después, en 1970, Ken Thompson, de los Laboratorios Bell, estaba intentando simplificar más el lenguaje BCPL. Tuvo éxito en crear un lenguaje muy sucinto que resultaba muy apropiado para utilizarlo en el hardware disponible. Lo denominó el lenguaje B, diseñado y desarrollado para una implementación inicial del sistema operativo UNIX. Sin embargo, puede que tanto BCPL como B llevaron demasiado lejos su intento de perfeccionamiento. Se convirtieron en lenguajes limitados, útiles solamente para ciertas clases de problemas. Por esta razón, en 1971 se comenzó a trabajar en un sucesor para el B, apropiadamente llamado lenguaje C.

En 1972, Dennis Ritchie, de los Laboratorios Bell, está acreditado como el creador del lenguaje C, que restauró parte de la generalidad perdida en BCPL y B. Lo logró con un uso perspicaz de los tipos de datos, mientras mantenía la simplicidad y el contacto con la computadora, que fueron los objetivos originales del diseño de CPL.

El lenguaje C está enfocado al diseño de software de sistemas. C es un lenguaje relativamente de bajo nivel, que permite especificar cada detalle en la lógica de un algoritmo para lograr la máxima eficiencia de la computadora. Pero C es también un lenguaje de alto nivel que puede ocultar los detalles de la arquitectura de la computadora, incrementando así la eficiencia de la programación (Pappas y Murray, 1994).

c) Arqueología Informática de hardware y software: computadoras decimales vs. computadoras binarias

Todas las computadoras electrónicas disponibles hoy en día utilizan el sistema binario, como forma de representación de los datos que procesan y la información que generan; pero en el comienzo de las computadoras, éstas se concibieron con el uso del sistema decimal; así lo podemos ver en la Máquina Analítica de Babbage.

El primer diseño de un computador controlado por programa fue realizado por Charles Babbage a mediados del siglo XIX. Este diseño no pudo llevarse a cabo en aquellos momentos, supuestamente porque era decimal y no binario, y por tanto muy complicado (WIKIPEDIA, 2011).

Una matemática aficionada y amiga de Babbage, Augusta Ada (1815-1852), Condesa de Lovelace e hija del poeta Lord Byron, se interesó mucho en la Máquina Analítica y trabajó junto con él. Está considerada como la primera programadora pues escribió secuencias de instrucciones en tarjetas perforadas, inventó métodos de programación como la subrutina e introdujo en sus programas las iteraciones y el salto condicional, lo que abre ya la posibilidad de tomar decisiones automáticamente. Otra de sus notables contribuciones es que propuso utilizar el sistema binario en lugar del decimal que utilizaba Babbage, para la codificación de los programas en tarjetas perforadas (Ferreyra Cortés, 2006).

A pesar de esto, tuvo que pasar más de un siglo para que el sistema binario se utilizara como forma de representación de los datos e información en una computadora. En 1941, un ingeniero alemán llamado Konrad Zuse, creó su computadora Z3, que fue la primera máquina programable de tecnología electromecánica y completamente automática, características usadas para definir a un computador. El éxito de Zuse suele ser atribuido al su uso del sistema binario, ya que el Z3 realizaba los cálculos con aritmética en coma flotante puramente binaria.

El sistema binario fue inventado hace unos 300 años, por el matemático alemán Gottfried Leibnitz (WIKIPEDIA, 2011), pero el interés de Leibnitz era religioso y no científico; intentaba construir una prueba de la existencia de Dios. Su sueño era desarrollar un “lenguaje simbólico generalizado”, y un álgebra como instrumento, de forma que se pudiera determinar la verdad de cualquier proposición en cualquier campo de la investigación humana, mediante un simple cálculo. Según sus propias palabras “no habría ya mayor necesidad de discusión entre dos filósofos que entre dos contables; les bastaría tomar sus lápices, sentarse ante el tablero y decir: Calculemos” (Biblioteca de Informática, 1993).

Posteriormente, a finales del siglo XIX, el sistema binario es utilizado por George Boole para desarrollar su álgebra booleana. En 1937, Claude E. Shannon, en su tesis de licenciatura, describió la utilización de la lógica simbólica y los números binarios en el diseño de los calculadores eléctricos (Biblioteca de Informática, 1993). Shannon fue el primero que introdujo la idea de implementar el álgebra booleana mediante relés electrónicos. Sin embargo, fue Zuse el que unió todo esto e hizo que funcionara (WIKIPEDIA, 2011).

Conclusiones y trabajos futuros

Se puede observar en los ejemplos presentados, que las antiguas ideas pueden refrescar los nuevos desarrollos, tanto en el hardware como en el software; de eso se trata la Arqueología Informática: desenterrar los elementos del pasado que pueden explicar el porqué de los diseños, con posibilidad de nutrir con nuevas ideas, los desarrollos de hoy en día.

La Arqueología Informática también muestra que ciertas ideas surgieron de otros contextos y con otros propósitos, pero debido a la influencia de ciertos personajes y diferentes sucesos, estas ideas tuvieron distinta aplicación, permitiendo avances tecnológicos importantes, como fue el caso del sistema binario. Esto muestra que las ciencias computacionales de hoy en día han sido producto, entre muchas otras cosas, de la convergencia de diversas ideas, de orígenes y contextos muy diferentes al mundo de las computadoras. Es un hecho que para crear cosas nuevas dentro de la ciencia informática, es necesario asirse de ideas tan frescas como variadas.

En conclusión, se puede decir que la Arqueología Informática es una herramienta teórica que puede utilizarse con dos fines principales: 1) como una enorme fuente de investigaciones históricas sobre el desarrollo de las computadoras y sus diversos componentes, con el objetivo de nutrir de manera sustancial el bagaje actual de la cultura informática de los diversos usuarios de computadoras; y 2) derivado de lo anterior, como una herramienta de conocimiento para el diseño y construcción de emuladores de dispositivos y/o la creación de nuevos equipos de cómputo y programas de software (Pérez Pérez y Monzalvo López, 2011).

Referencias

Alcalde, E.; García, M. (1994). *“Informática básica”*. Segunda edición. Editorial McGraw-Hill. México. pp. 25.

Appleby, D.; Vandekopple, J.J. (1998). *“Lenguajes de programación. Paradigma y práctica”*. Editorial Mc Graw-Hill. México. pp. 125.

Biblioteca de informática. (1993). Vol. 1. pp. 8, 16 y 72; Vol. 2. pp. 361, 364, 365, 368 y 369; Vol. 5. pp. 1479 y 1480. Editorial Limusa Noriega Editores. Primera edición. México.

Castelazo, Y. (2010). *“Alan M. Turing”*. Universidad Autónoma Metropolitana Iztapalapa. Disponible en: http://turing.izt.uam.mx/html/index.php/page/Alan_M_Turing.html. México. (Consultado: Septiembre, 2011).

Ferreya Cortés, G. (2006). *“Informática para cursos de Bachillerato”*. Segunda Edición. Editorial Alfaomega. México. pp. 23 y 24.

Garner, C. (1984). *“Conocer la computación. La informática al alcance de todos”*. Primera edición. Editorial ELASA. México. pp. 19.

Helmut, T. (2010). *“Historia de la computación”*. HELMUT SY VU. Disponible en: http://helmuty.homestead.com/files/computacion/Historia/Historia_Computadores_3.htm Venezuela. (Consultado: septiembre, 2011).

Nuncio Limón, R. (1991). *“Historia y perspectivas de la programación. Fundamentos de Informática”*. Editorial Trillas. Primera edición. México. pp. 1, 17-19, 30.

Pappas, C. H.; Murray, W. H. (1994). *“Manual de Borland C++ 4.0”*. Primera edición. Editorial McGraw-Hill. España. pp. 85 a 87.

Pérez Pérez, I., Monzalvo López, C. A. (2011). *“Introducción a la Arqueología Informática”*. En prensa. Área Académica de Computación. ICBI-UAEH. México.

Quantum Networks. *“Charles Babbage: El Padre de la Computación Moderna”*. Disponible en: <http://www.quantum-networks.com/articulos/charles-babbage--el-padre-de-la-computacion-moderna.html>. E.E.U.U. (Consultado: Septiembre, 2011).

Schmidhuber, J. (2009). *“Zuse’s Thesis: The universe is a computer”*. Jürgen Schmidhuber's. Disponible en: <http://www.idsia.ch/~juergen/>. México. (Consultado: Septiembre, 2011).

Tucker, A.B. (1987). *“Lenguajes de programación”*. Editorial McGraw-Hill. Segunda edición. España. pp. 18 y 19.

WIKIPEDIA. *“Editing Automatic Computing Engine”*. Disponible en: http://en.wikipedia.org/wiki/Automatic_Computing_Engine. (Consultado: marzo, 2011).

WIKIPEDIA. *“Z3”*. Disponible en: <http://es.wikipedia.org/wiki/Z3>. (Consultado: marzo, 2011).