Academia Mexicana de Computación
Boletín: 04 Año: 2022 Fecha: 20 de diciembre de 2022

Efemérides del trimestre

Jeffrey David Ullman.
Nació el 22 de noviembre de 1942.
Ganador del *ACM Turing Award* en 2020.
Realizó contribuciones teóricas y desarrolló algoritmos muy importantes en torno a la implementación de lenguajes de programación, y escribió numerosos libros que han servido para educar a múltiples generaciones de estudiantes de computación en todo el mundo.

Martin Hellman.
Nació en octubre de 1945.
Ganador del *ACM Turing Award* en 2015.
Inventó (con Diffie Whitfield) la criptografía asimétrica de llave pública y desarrolló aplicaciones de esta a firmas digitales.

Michael Stonebraker.
Nació el 11 de octubre de 1943.
Ganador del *ACM Turing Award* en 2014.
Realizó contribuciones seminales a los conceptos y el uso de los sistemas de bases de datos modernas.

Silvio Micali.
Nació el 13 de octubre de 1954.
Ganador del *ACM Turing Award* en 2012.
Junto con Shafi Goldwasser, estableció las bases teóricas para la criptografía, y en el proceso, desarrolló nuevos métodos eficientes de verificación de pruebas matemáticas en teoría de la complejidad.

Joseph Sifaker.
Nació el 26 de diciembre de 1946.
Ganador del *ACM Turing Award* en 2007.
Junto con Edmund Clarke y E. Allen Emerson, desarrolló una herramienta muy efectiva de verificación, usada en ingeniería de software, la cual se conoce como "Chequeo de Modelos".

Saludo Editorial

Es un gran placer darles la bienvenida al cuarto y último número de nuestro Boletín, que inicia con un resumen del artículo "The Computer Programs of Charles Babbage" que publicó el Dr. Raúl Rojas en el *IEEE Annals of the History of Computing*, en el cual se describe la arquitectura de programación de la "Máquina Analítica" del llamado "padre de la computación".

La Dra. Rocío Aldeco Pérez y el Dr. Sergio Rajsbaum nos proporcionan un artículo muy ameno sobre "*Blockchain*", en el cual hablan sobre su significado, sus orígenes y su funcionamiento. En la parte final del artículo hablan brevemente sobre la participación de la UNAM en un Centro de Excelencia Algorand, relacionado con blockchain.

Las Mujeres en Computación de la Amexcomp presentan una breve nota en torno al galardón que recibieron en la categoría de *Sororidad* en la edición 2022 del *Dev Day 4 Women* (DD4W) que organiza el equipo *SG4 Women de Software Guru*. ¡Muchas felicidades por este merecido reconocimiento!

El Dr. Ramón F. Brena Pinero nos proporciona un interesante artículo titulado "¿Reemplazará la IA a los programadores?" en el cual habla sobre la programación automática y la forma en la que la investigación en torno a este tema ha evolucionado a lo largo de los años.

La Dra. Karina Mariela Figueroa Mora y el Dr. Efrén Mezura Montes nos proporcionan una reseña sobre la *IEEE International Mexican Conference on Computer Science* (ENC 2022) que se llevó a cabo del 24 al 26 de agosto pasados de forma virtual.

La Dra. María del Pilar Gómez Gil y el Dr. Hugo Terashima Marín nos proporcionan un reporte sobre la reunión anual de coordinadores de doctorado que organizamos el pasado 12 de octubre de forma híbrida.

De mi parte, contribuí a este boletín con un reporte de la reunión anual de la AMEXCOMP que se llevó a cabo el pasado 13 de octubre en forma híbrida.

Finalmente, en nuestra columna titulada "Recordando a...", hablamos en esta ocasión de Butler Lampson, que ha realizado contribuciones relevantes en diversas áreas de la computación, incluyendo la arquitectura de computadoras, los sistemas operativos, los lenguajes de programación y el cómputo tolerante a fallas, entre muchas otras.

Un cordial saludo,
Dr. Carlos Artemio Coello Coello.
Presidente de la AMexComp

Los Programas de Charles Babbage

Por Dr. Raúl Rojas.
Freie Universität Berlin.

Con este breve resumen quisiera llamar la atención de la comunidad AMEXCOMP hacia un artículo que publiqué en la revista *IEEE Annals of the History of Computing*, con el título "The Computer Programs of Charles Babbage" (enero de 2021). El artículo describe lo que hoy llamamos la arquitectura de programación de la "Máquina Analítica", un aparato de cálculo diseñado por Babbage que, de haber sido completado, hubiera sido la primera computadora del mundo. La arquitectura de programación es la descripción abstracta de cómo se escriben y ejecutan programas sin considerar el tipo específico de hardware de la máquina, en este caso, engranes mecánicos.

Charles Babbage diseñó la Máquina Analítica (MA) a partir de 1833. Después de haber trabajado en la Máquina Diferencial, una máquina estructurada como un *pipeline* de acumuladores para la adición y substracción, Babbage se dio cuenta de que podía construir una máquina más poderosa y capaz de realizar cálculos "analíticos" (que hoy llamaríamos generales o universales). Estuvo trabajando en la MA durante décadas. Fruto de ese trabajo es un conjunto de 27 programas escritos de 1836 a 1840 que se pueden consultar hoy en día en el archivo digital del Museo de Ciencias de Londres. El artículo que resumo describe el área de aplicación de cada programa y deriva de su estudio la arquitectura de la MA.

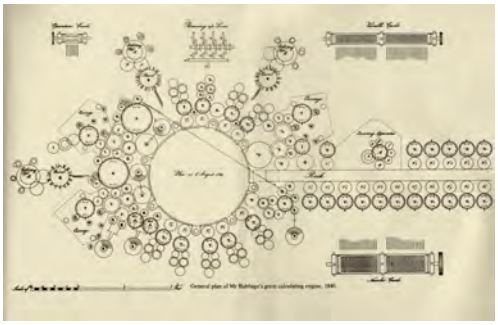

Figura 1. Máquina Analítica.

La MA tenía un procesador capaz de ejecutar las cuatro operaciones aritméticas elementales, utilizando dos argumentos. La memoria podría alojar hasta mil números de varios dígitos decimales (Babbage llegó a soñar con una memoria con 40 dígitos por localidad). La memoria podía enviar dos argumentos al procesador. Este podía realizar un cálculo y enviar el resultado de regreso a la memoria. Lo curioso de la MA es que se programaba con tarjetas perforadas encadenadas. Había una cadena para el procesador y otra, separada, para la memoria. Esto quiere decir que el procesador leía una tarjeta de su lectora y se ponía, por ejemplo, en modo de "adición", mientras que la memoria leía las dos localidades de memoria que había que enviar hacia el procesador. El procesador no sabe, en esta arquitectura, qué localidades de memoria han sido enviadas para ser sumadas, y la memoria no sabe qué operación está siendo ejecutada. Esta separación le permitía a Babbage ordenarle al procesador realizar 100 adiciones (con un contador) y se requería una sola tarjeta de adición para el procesador, mientras que las direcciones de los argumentos para la memoria eran leídas tarjeta tras tarjeta.

Esperamos sus contribuciones y avisos al correo del boletín:

boletin@amexcomp.org.mx

las cuales son muy importantes para mantener vivo el boletín.

La separación estricta entre operaciones y localidades de memoria implica que no se puede escribir código estático: solo se puede ejecutar el programa y se anota lo que va haciendo. Es decir, los programas de Babbage son más bien "traces", es decir, la historia de lo que se fue calculando. Es difícil leerlos porque hay que estar checando cuáles tarjetas para el procesador corresponden a qué tarjetas para la memoria.

La MA tenía la posibilidad de checar un número y reenrollar las tarjetas para el procesador y la memoria a una posición anterior. Es decir, había un "salto condicional" que posibilitaba escribir ciclos (*loops*) de programación, además de algo como "*if-then-else*". El hecho de que las tarjetas de memoria se desplazaran independientemente de las operaciones, le daba a la MA una especie de "direccionamiento indexado", tan necesario en ciclos de programación para poder acceder tablas de datos.

En el artículo describo todas estas posibilidades de la MA y especialmente un programa para la multiplicación de los coeficientes de polinomios. Este programa es importante porque muestra cómo se podrían realizar cálculos algebraicos en la MA. Es la joya de la corona del código escrito por el profesor de Cambridge.

Desde mi punto de vista Babbage no pudo terminar la MA por dos razones: la primera es que la complejidad de la máquina rebasaba las posibilidades de precisión mecánica de la época, la segunda es que Babbage rediseñó la máquina varias veces y nunca llegó a un conjunto de planos definitivo. Esto es lo que concluyó un grupo de trabajo que intentó reconstruir la MA hace varios años. Aún así, mi artículo muestra que la arquitectura de programación de la MA es tan universal como la de las computadoras modernas, y corrige simulaciones simplistas de la MA, disponibles en Internet, que no implementan la estricta separación entre las tarjetas de instrucciones para el procesador y para la memoria.

Claramente Charles Babbage fue el primer programador de la historia. Luigi Menabrea publicó un folleto en francés en 1842, con la primera descripción de algunos programas para la MA. Ada Lovelace tradujo el folleto de Menabrea del francés al inglés, con anotaciones propias, en 1843, es decir, tres años después del último programa escrito por Babbage. El programa para calcular los números de Bernoulli, contenido en las anotaciones de Lady Lovelace, le fue entregado por Babbage, como este reporta en sus memorias de 1864. Pero de todo esto da cuenta el artículo que está disponible en el repositorio digital de IEEE.

Anuncio importante

El costo de la membresía anual de los miembros de la Academia Mexicana de Computación será de $1200.00 pesos a partir del 1 de enero de 2023. Se hace notar que quienes deseen cubrir el pago de membresía de años anteriores al 2023, la cuota seguirá siendo de $1000.00 pesos.
El pago de la membresía debe hacerse en el transcurso del año mediante depósito a:

TITULAR: ACADEMIA MEXICANA DE COMPUTACION AC
BANCO: BBVA
No. Cta: 0198653992
CLABE: 012180001986539926

Mucho le agradeceremos anotar su nombre completo, dirección y RFC en la referencia del depósito y enviar copia al correo:

administracion@amexcomp.org.mx

Las contribuciones de nuestros miembros son esenciales para las diferentes actividades de la Academia.

Referencias:

[1]. R. Rojas, "The Computer Programs of Charles Babbage", in: *IEEE Annals of the History of Computing*, V. 43, N. 1, p. 6-18, Enero-Marzo 2021.

[2]. R. Rojas, "Charles Babbage, Pionero de la Computación", Confabulario, El Universal, Mayo 8, 2021,

https://confabulario.eluniversal.com.mx/charles-babbage-maquina-analitica/.

¿Qué es Blockchain?

Por:
Dra. Rocío Aldeco-Pérez, Facultad de Ingeniería (UNAM),
Dr. Sergio Rajsbaum, Instituto de Matemáticas (UNAM).

Blockchain, traducido al español como cadena de bloques, es una base de datos descentralizada, consensuada y distribuida entre múltiples nodos pertenecientes a una red que permite almacenar información de forma inmutable, ordenada y transparente. La palabra "bloque" se refiere a un conjunto de transacciones que son propuestas y verificadas por los otros nodos y que finalmente se añaden a la base de datos. La palabra "cadena" se refiere al hecho de que cada bloque de transacciones también contiene una prueba (un hash criptográfico) de lo que hay en el bloque anterior. Este patrón de captura de los datos del bloque anterior en el bloque actual continúa hasta el inicio de la cadena, donde se encuentra el bloque génesis, creando un registro públicamente verificable y a prueba de manipulaciones de todas las transacciones.

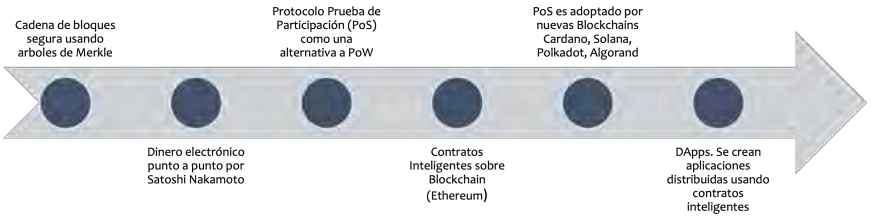

Figura 2. Historia del Blockchain.

Los orígenes de la blockchain se remontan a 1991 cuando Stuart Haber y W. Scott Stornetta propusieron crear una cadena de bloques criptográficamente segura en la que nadie podía manipular las marcas de tiempo de los documentos [1]. Posteriormente, en 1992, mejoraron esta idea al incorporar árboles de Merkle incrementando su eficiencia y permitiendo así la recopilación de más documentos en un solo bloque [2].

No fue sino hasta 2008 que la Blockchain como la conocemos hoy fue conceptualizada por la persona o grupo llamado "Satoshi Nakamoto" [3] donde se presenta un sistema de dinero electrónico punto a punto que permite la generación de "monedas digitales" así como la transferencia y resguardo de estas. Estas monedas son lo que conocemos como Bitcoin, la primera criptomoneda. Bitcoin permite crear transacciones que son firmadas digitalmente por sus creadores y que no requieren de una autoridad central para ser aprobadas. Esto se logra gracias al uso del algoritmo Hashcash [4] y el protocolo de consenso Proof of Work (PoW). En este protocolo los nodos participantes compiten para resolver un problema criptográfico muy difícil y presentar su solución junto con una nueva propuesta de bloque (esto es lo que se llama "minería" y a los nodos "mineros"). El ganador es recompensado con parte de la moneda subyacente del sistema y su bloque pasa a formar parte de la cadena. Es así que el resolver un problema de consenso mediante PoW permite crear un acuerdo entre todos los nodos, de cuál es el siguiente bloque en la cadena.

Además, como consecuencia del uso de llaves públicas y privadas creadas por un algoritmo de firma digital, las transacciones son anónimas. Esto quiere decir que se sabe que la transacción fue realizada por una llave pública (anónima) pero se desconoce al dueño de esta. Sin embargo, el protocolo de consenso de PoW requiere de alto poder computacional para resolver el problema criptográfico dado y es necesaria la participación de la mitad de los nodos en dicho consenso. Esto hace de Bitcoin una solución lenta, poco escalable y con un enorme consumo energético.

En 2013, preocupado por las limitaciones de Bitcoin, Vitalik Buterin propone una blockchain maleable que es capaz

de realizar varias funciones más allá de sólo transferencias de dinero electrónico de una cuenta a otra. Así surge Ethereum [5], una plataforma de blockchain descentralizada que permite a las personas registrar otros activos (no sólo criptomonedas) así como la ejecución de los llamados "contratos inteligentes". Estos contratos son programas auto-verificables, auto-ejecutables y resistentes a la manipulación que consisten de un conjunto de transacciones que se ejecutan en la Blockchain sin necesidad de una autoridad central de confianza. Las transacciones se envían desde cuentas de Ethereum que son llaves públicas anónimas. El remitente debe firmar las transacciones y gastar Ether (la criptomoneda nativa de Ethereum) como costo del procesamiento de las transacciones en la red. Todo esto hace que en Ethereum sea posible desarrollar aplicaciones descentralizadas. A pesar de las evidentes mejoras que ofrece Ethereum sobre Bitcoin, aún persiste el uso de PoW como protocolo de consenso lo que no resuelve los problemas centrales de Bitcoin.

Así en 2012, surge un nuevo protocolo de consenso llamado Proof of Stake (PoS) [6] propuesto por el desarrollador cuyo seudónimo es Sunny King. PoS es una manera más ecológica de llegar a un consenso, ya que no requiere de alto procesamiento y no utiliza grandes cantidades de electricidad. PoS da a los nodos que tienen más participación, o sea, los que tienen más de la moneda subyacente en el sistema, más influencia en la propuesta y validación de nuevos bloques, normalmente a través de algún tipo de mecanismo de votación. Esto crea blockchains más escalables, eficientes y ecológicas. Como resultado de esto, surgen nuevas redes que usan PoS para su consenso. Entre ellas podemos mencionar Ethereum 2.0, Cardano [7], Polkadot [8], Solana [9] y Algorand [10].

A pesar de que estas blockchains tienen sus propias redes, tecnologías y criptomonedas, todas estas comparten la misma estructura y funcionamiento general. Esta estructura se presenta en la Figura 3, donde cada bloque de la blockchain está ligado al bloque anterior usando sus hashes criptográficos, de tal forma que corromper el bloque actual o los previos es computacionalmente muy difícil.

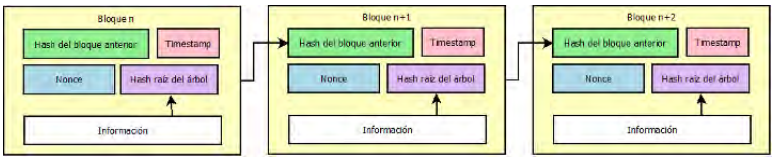

Figura 3. Estructura de bloques en Blockchain.

Nuevos bloques pueden ser añadidos si existe un acuerdo entre la mayoría de las partes. Este acuerdo se realiza a través de los protocolos de consenso (PoW o PoS) que garantizan con alta probabilidad que no se agreguen a la cadena dos bloques concurrentemente que ocasionen una bifurcación en la cadena, con transacciones inconsistentes. La información de un bloque no puede ser modificada, debido a las herramientas de criptografía usadas (hashes y firmas digitales), aunque un bloque podría estar en la cadena tentativamente, si hubiera una bifurcación. Sin embargo, transcurrido cierto tiempo de haber agregado un bloque, se puede asumir que este quedará en la cadena de forma permanente. Es en este sentido que la información almacenada en la cadena ya no podrá ser modificada (inmutabilidad). De esta manera es posible generar confianza entre los participantes sin que necesariamente estos se conozcan entre sí o exista una entidad tercera que respalde esta confianza, ya que, además cualquier participante puede verificar que los bloques están bien formados y concatenados correctamente.

En resumen, el blockchain une varios elementos importantes en su funcionamiento: (1) una base de datos distribuida entre todos los participantes sin necesidad de una autoridad central, (2) un protocolo de consenso que garantiza que la información en esta base de datos sea idéntica para todos, y verificable por todos y (3) el uso de algoritmos criptográficos para garantizar la integridad e inmutabilidad de la información dentro de esta base de datos.

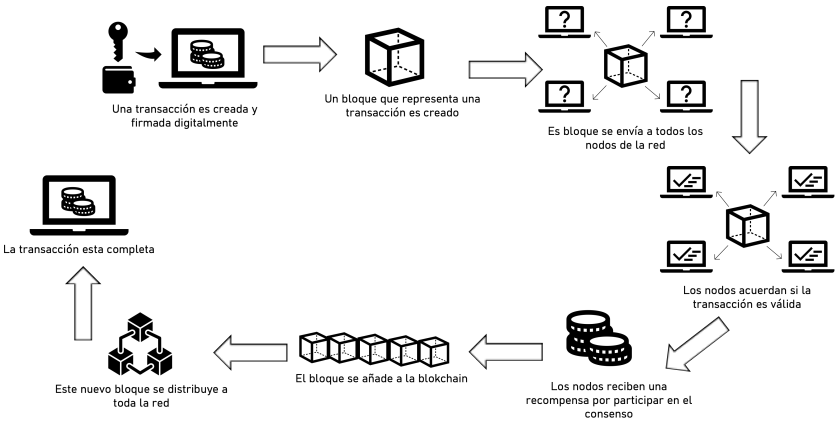

Figura 4. Funcionamiento de Blockchain.

Como puede verse en la Figura 4, la información que se desea almacenar en la Blockchain se guarda en forma de transacción para ser firmada digitalmente. Esta transacción se representa como un bloque que posteriormente es enviado a todos los nodos de la red. En ese momento los nodos ejecutan el protocolo de consenso elegido para decidir si esta transacción es válida. Los nodos que sean parte de este protocolo recibirán una recompensa por su participación (usualmente en la moneda nativa). Posteriormente este bloque se vuelve válido y se añade a la Blockchain para después ser distribuido al resto de los nodos en la red.

Los promotores de blockchain afirman desde su creación que esta tecnología revolucionaría diversos sectores de la sociedad, entre ellos el mundo financiero o la democracia. La realidad es que es aún difícil de entender el alcance de los blockchains debido a su complejidad y los cientos de blockchains existentes todas soportadas por tecnologías drásticamente diferentes. Al mismo tiempo hay diversas criptomonedas que se ejecutan en estas blockchains, cada una con diferentes fortalezas y debilidades. El entusiasmo por la tecnología de blockchains es sin embargo enorme. Muestra de ello es la creación de múltiples compañías de blockchains, el desarrollo de soluciones basadas en blockchains por parte de grandes empresas de software y la creación de centros de investigación en blockchains en las universidades más importantes del mundo. Por poner solo un ejemplo, la prestigiosa Universidad de Yale en EUA, creó un centro de investigación multidisciplinario de blockchains financiado con 5.75 millones de dólares por cinco años [11]. En México, la UNAM participa en un Centro de Excelencia Algorand financiado por Algorand Foundation [12]. El proyecto, denominado MEGA-ACE financiado con 8 millones de dólares durante 3 años, tiene como objetivo el constituirse en centro impulsor de la investigación sobre blockchain, un foro para el intercambio de ideas en el área, un centro para la formación de estudiantes en países que actualmente no ofrecen este tipo de formación y un facilitador de la movilidad para el personal (profesional y de estudiantes) [13].

Referencias:

[1]. Haber, S., Stornetta, W.S. (1991). How to Time-Stamp a Digital Document. In: Menezes, A.J., Vanstone, S.A. (eds) Advances in Cryptology-CRYPTO'90. CRYPTO 1990. Lecture Notes in Computer Science, vol 537. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-38424-3_32

[2]. Bayer, D., Haber, S., Stornetta, W.S. (1993). Improving the Efficiency and Reliability of Digital Time-Stamping. In: Capocelli, R., De Santis, A., Vaccaro, U. (eds) Sequences II. Springer, New York, NY. https://doi.org/10.1007/978-1-4613-9323-8_24

[3]. Nakamoto, S. (2009). Bitcoin: A Peer-to-Peer Electronic Cash System. www.bitcoin.org

[4]. A. Back, "Hashcash - a denial of service counter-measure", http://www.hashcash.org/papers/hashcash.pdf, 2002.

[5]. Buterin, V. (2014). A next-generation smart contract and decentralized application platform. Etherum, (January), 1–36. Retrieved from http://buyxpr.com/build/pdfs/EthereumWhitePaper.pdf

[6]. King, S., & Nadal, S. (2012). PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake,

https://decred.org/research/king2012.pdf

[7]. David, B., Gaži, P., Kiayias, A., Russell, A. (2018). Ouroboros Praos: An Adaptively-Secure, Semi-synchronous Proof-of-Stake Blockchain. In: Nielsen, J., Rijmen, V. (eds) Advances in Cryptology – EUROCRYPT 2018 . EUROCRYPT 2018. Lecture Notes in Computer Science(), vol 10821. Springer, Cham. https://doi.org/10.1007/978-3-319-78375-8_3

[8]. Wood, G. (2017). Polkadot: Vision for a Heterogeneous Multi-Chain Framework. White Paper, Draft 1, 1–21. Retrieved from https://polkadot.network/PolkaDotPaper.pdf

[9]. Solana [9]

[10]. Chen J., Mical S. i(2019). Algorand: A secure and efficient distributed ledger.

[11]. Theoretical Computer Science, Volume 777, Pages 155-183, ISSN 0304-3975,

https://doi.org/10.1016/j.tcs.2019.02.001

[12]. With $5.75M Grant, Yale Leads Multidisciplinary Blockchain Center | Yale School of Engineering & Applied Science, https://seas.yale.edu/news-events/news/575m-grant-yale-leads-multidisciplinary-blockchain-center, Accessed: 2022-11-30

[13]. MEGA-ACE (Multi-disciplinary Educational Global Alliance for Algorand Centre of Excellence) is an unique academic endeavour composed of 17 institutions across 6 continents, forming a powerhouse of multi-disciplinary blockchain research and education, https://www.algorand.foundation/ace-university/purdue-university, Accessed: 2022-11-30

[14]. UNAM, única universidad mexicana que participa en un Centro de Excelencia Algorand,

https://www.dgcs.unam.mx/boletin/bdboletin/2022_753.html, Accessed: 2022-11-30

Mujeres en Computación del AmexComp

Estamos muy orgullosos de compartirles que el pasado 28 de octubre de este año el grupo de Mujeres en Computación del AmexComp fue galardonado en la categoría de Sororidad en la edición 2022 del Dev Day 4 Women (DD4W) que organiza el equipo SG 4 Women de Software Guru. Nuestras representantes en esta ocasión fueron la Dra. Alejandra Silva Trujillo de la Universidad Autónoma de San Luis Potosí y la Dra. Belém Priego Sánchez de la Universidad Autónoma Metropolitana unidad Azcapotzalco.

Cabe señalar que el DD4W es un evento que tiene como objetivo apoyar a las mujeres involucradas en la creación de software. Este año fue la quinta edición del evento y se ha convertido en una de las conferencias más importantes para mujeres en Latinoamérica, la cual, a partir del 2020, se ha llevado a cabo en formato virtual.

Agradecemos a los organizadores la invitación a participar en este evento y, sobre todo, por el reconocimiento a la labor que realiza el grupo de Mujeres en Computación.

Figura 5. Reconocimiento otorgado al grupo de Mujeres en Computación del AmexComp.

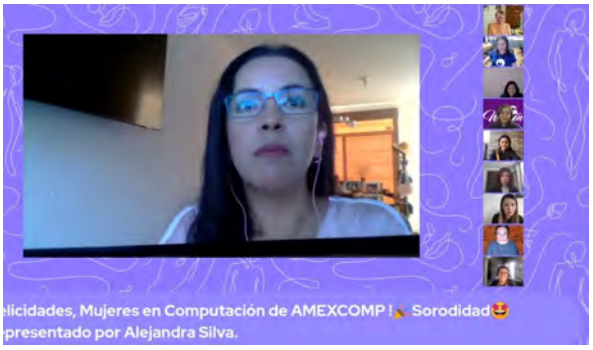

Figura 6. Dev Day 4 Women (DD4W).

¿Reemplazará la IA a los programadores?

Por Dr. Ramón F. Brena Pinero.
Tecnológico de Monterrey.

¿Qué es la "programación automática" a fin de cuentas?

Existe cierto temor entre los programadores profesionales de que algún día, tarde o temprano, la IA se haga cargo de sus trabajos (te estoy observando, GPT-3).

¿Qué van a hacer?

Síntesis de programas

Mi viaje de investigación en IA comenzó hace casi 40 años, tratando de resolver con precisión el problema de la programación "automática" de la IA.

En la década de los 80s estaba en Francia terminando mi doctorado. La tesis era sobre "Síntesis de Programas", que era como solíamos llamar a la "programación automática" hecha por IA. Conocía a todos los investigadores líderes en la comunidad internacional de Síntesis de programas.

Aunque obtuve algunos resultados preliminares (¡entre ellos, terminar mi doctorado!), todo el asunto de la síntesis de programas resultó ser un problema mucho más difícil de lo que parecía. Pasó de moda en los años siguientes.

Más tarde, me encontré con algunos de los investigadores de síntesis de programas, quienes me dijeron que estaban haciendo otra cosa.

Y yo también.

La misma historia que con la traducción automática

La triste historia anterior no significa que la búsqueda de la programación automática haya terminado.

Algo similar sucedió también con la traducción automática en los años sesenta (sí, 1960). Algunos investigadores estadounidenses y rusos se reunieron con la esperanza de obtener traducciones basadas en computadora en pocos años. Por supuesto, fracasaron miserablemente y pasaron a otros temas.

Pero después de 2010, casi 50 años después, los resultados innovadores en la traducción finalmente llevaron a resultados prácticos.

En 2012 conversé con Franz Och, entonces jefe de traducción de Google, y tuvo la amabilidad de informarme sobre las limitaciones de la traducción estadística en Google.

Finalmente, en la década actual, estamos comenzando a tener herramientas de traducción aceptables disponibles para el público en general.

Las tres muertes

Safi Bahcall, el autor del libro "Loonshots", argumenta que las innovaciones verdaderamente revolucionarias tienen que soportar tres muertes. Eso le pasó al radar, a medicamentos muy exitosos rechazados varias veces antes de alcanzar el éxito... y a la traducción automática.

Creo que pasará lo mismo con la programación automática, pero no muy pronto.

¿A qué nos referimos con "programación automática"?

Antes que nada, tenemos que preguntarnos qué significa "programar una computadora" para que podamos juzgar si la tarea de automatizarla se ha cumplido o no.

En los primeros días de las computadoras, la programación se hacía a nivel de instrucción de CPU o en lenguaje ensamblador, que era lo mismo pero con nombres en lugar de códigos binarios.

El punto aquí es que la programación solía significar construir secuencias de códigos de máquina.

Entonces, cuando John Backus y su equipo desarrollaron FORTRAN, con instrucciones de "alto nivel" que luego se traducían a varias instrucciones de máquina, algunas personas lo llamaron "programación automática". Y en cierto modo tenía sentido porque uno no escribía en códigos de máquina; más bien le indicaba a la computadora qué hacer... en FORTRAN.

El lector (de menos de 60 años) no lo va a creer, pero en ese entonces era incluso controversial usar cualquier cosa que no fuera el lenguaje ensamblador.

Donald Knuth, el autor de "El arte de programar computadoras" (la Biblia de la programación para muchos), aconsejó enfáticamente apegarse al ensamblador. Según él, "Al comprender un lenguaje orientado a máquina, el programador tenderá a utilizar código mucho más eficiente; está mucho más cerca de la realidad".

¿Es la programación de alto nivel "programación automática"?

En un lenguaje de programación de "alto nivel", las instrucciones de la máquina se dan con el menor detalle posible. Cuando decimos "alto nivel", decimos "más abstracto".

La búsqueda por ofrecer lenguajes de programación de mayor y mayor nivel ha continuado con lenguajes como Haskell (programación funcional), Prolog (programación lógica) y muchos otros.

La programación orientada a objetos también trajo algunos mecanismos de abstracción, como encapsulación y herencia. Pero casi nadie afirma que hacen la programación "automática".

Ni siquiera los lenguajes de programación populares actuales como Javascript, Swift, Python, C# y Go son suficientes.

¿Y entonces qué? ¿Algo calificará como "programación automática"?

¿Qué es la programación automática en realidad?

Aquí voy con mi definición "oficial" de programación automatizada:

"En la programación automática, el ser humano simplemente dice lo que debe hacer el programa y la máquina crea un código ejecutable para hacer exactamente eso".

Dicho de otra forma, el programador recibe el qué y entrega el cómo. Tan sencillo como eso.

Entonces, en la programación automática, *la programación la realiza una máquina*.

Probemos nuestra definición de "programación automática" con programación sin código:

¿Está familiarizado con plataformas sin código como Bubble, Airtable y otras? Sus anuncios dicen que no tienes que escribir una sola línea de código. Eso debería ser programación automática, ¿verdad?

No según la definición anterior.

Para un programa Bubble específico, en ninguna parte se puede escribir lo que se supone que debe hacer el programa (excepto en los comentarios, por supuesto). La conversión del "qué" al "cómo" es enteramente responsabilidad del programador.

En Bubble, el programador también debe ocuparse de los condicionales, las estructuras de datos y las excepciones... Tomé un curso intensivo de Bubble de dos semanas y apenas pasamos la superficie.

¿La codificación usando GPT-3 es programación automatizada?

Existen algunas herramientas de vanguardia para ayudar a los programadores usando tecnologías de aprendizaje profundo, en particular, GPT-3 de Open AI.

En los ejemplos que yo he visto, el sistema de IA no genera código a partir de una especificación "qué", sino que completa el código a partir de fragmentos parciales o el encabezado del procedimiento y algunos comentarios.

Esto se adapta muy bien a las tareas de "llenar los espacios en blanco" para las que se desarrolló GPT-3. Como hay millones y millones de ejemplos de código, es posible entrenar un sistema para descubrir cómo completar un fragmento de código.

Sin embargo, esto *no es programación automática*: no le dices a la computadora lo que debe hacer.

A duras penas podría admitir que los comentarios que preceden a un procedimiento podrían ser una forma de "qué", pero demasiado informales y vagos para ser considerados un requisito de software genuino.

Reflexiones finales

En conclusión, los programadores pueden estar seguros de que tienen un trabajo seguro por ahora. Las máquinas no los pondrán en desempleo en un futuro previsible.

Incluso suponiendo que la "programación automática real" fuera una realidad, las cosas no serían tan malas para los programadores.

En efecto, construir unos buenos requerimientos de software (el "qué") para un programa de software *no es una tarea trivial*. Hay chistes sobre esto en la comunidad de Ingeniería de Software, diciendo que el vendedor promete un palacio, el gerente imagina un condominio y los programadores terminan implementando una choza.

La realidad es que es difícil acertar con los requerimientos.

Incluso si los programadores solo escriben requerimientos, eso también es un trabajo y no es trivial – ni mal pagado tampoco.

Recursos:

[1]. Síntesis de Programas: Conocimiento y Deducción en los Dominios de Aplicación (Tesis en Francés).

[2]. Open AI GPT-3.

[3]. Safi Bahcall, "Loonshots".

[4]. Donald Knuth, "The Art of Computer Programming".

Reseña de la IEEE International Mexican Conference on Computer Science (ENC 2022)

Por:
Dra. Karina Mariela Figueroa Mora (Universidad Michoacana de San Nicolás de Hidalgo)
karina.figueroa@umich.mx,
Dr. Efrén Mezura-Montes (Universidad Veracruzana)
emezura@uv.mx.

Del 24 al 26 de agosto de 2022 se llevó a cabo, en formato virtual, la *IEEE Mexican International Conference on Computer Science* (ENC 2022), organizado por la Sociedad Mexicana de Ciencia de la Computación A.C. y co-organizado por la Facultad de Ciencias Físico-Matemáticas de la Universidad Michoacana de San Nicolás de Hidalgo, el Instituto de Investigaciones en Inteligencia Artificial de la Universidad Veracruzana, que fue la sede virtual del evento, el Tecnológico Nacional de México - Instituto Tecnológico de Culiacán, CENIDET, el Centro de Investigación en Matemáticas y la IEEE Sección Veracruz.

Cada año, el ENC busca ser un foro donde estudiantes, investigadores y personas de la industria presentan sus trabajos más recientes e intercambian ideas sobre los temas actuales y relevantes en lo referente a la Ciencia de la Computación y áreas relacionadas.

La edición 2022 del ENC se organizó en siete talleres:

- Inteligencia Artificial: Métodos, Algoritmos y Aplicaciones,
- Seguridad Computacional y de la Información: métodos y aplicaciones,
- Tecnologías Emergentes en Educación,
- Ambientes Inteligentes Inclusivos y Equitativos,
- Informática Médica para la Salud y el Bienestar,
- Cómputo Científico Aplicado a Problemas Geoespaciales,
- Ingeniería de Software.

Se recibieron 86 artículos de 73 instituciones nacionales e internacionales. Mediante un proceso de revisión doble-ciega con al menos tres revisores nacionales e internacionales, seleccionados por su trayectoria académica, se aceptaron 49 artículos para presentación oral (57 % de aceptación).

Tres reconocidos investigadores impartieron conferencias magistrales: la Dra. Pilar Gómez del INAOE nos habló sobre

Eventos

EIR 2022/2023
Escuela de Invierno de Robótica
09-13 de Enero, 2022: Evento.

IEEE ICC 2023
IEEE International Conference on Communications
May 28th - June 01st, 2023: Event.

ICT4S 2023
International Conference on Information and Communications Technology for Sustainability
June 5th, 2023: Pre-conference workshops,
June 6th-8th, 2023: Main conference,
June 6th-8th, 2023: Post-conference workshops.

SIPECO 2023
3er Seminario Iberoamericano de Pensamiento Computacional
26-28 de Junio, 2023: Evento.
Antioquia, Colombia.

ética e Inteligencia Artificial, el Dr. Víctor González y González de Sperientia [studio+lab]®, presentó un panorama sobre el reto de ser académicos emprendedores, y el Dr. Edgar Duéñez Guzmán, de DeepMind, comentó sobre el uso de la Inteligencia Artificial para estudiar fenómenos sociales.

El programa se complementó con un coloquio de estudiantes donde se presentaron trabajos de tesis en desarrollo, los cuales recibieron retroalimentación por parte de investigadores. Por otro lado, los asistentes pudieron elegir entre doce tutoriales sobre temas actuales en Ciencia de la Computación. Finalmente, se llevaron a cabo tres paneles, uno sobre Seguridad Informática, otro sobre Ingeniería de Software y el último sobre Interacción Humano-Computadora, donde expertos discutieron sobre estos temas.

Con el paso de los años, el ENC se ha convertido en un referente para la presentación de trabajos de investigación en el área de Ciencia de la Computación, que ofrece a los estudiantes tutoriales sobre temas actuales en los que pueden incursionar, además de ser un punto de encuentro para la comunidad que discute y planea sobre los retos futuros del área a nivel nacional e internacional.

Reunión Anual de coordinadores de doctorado - AMEXCOMP

Por: Dra. María del Pilar Gómez Gil (Instituto Nacional de Astrofísica, Óptica y Electrónica)
pgomez@inaoep.mx,
Dr. Hugo Terashima Marín (Tecnológico de Monterrey)
terashima@tec.mx.

El pasado 12 de Octubre, previo al evento anual de la Amexcomp, se llevó a cabo la ya tradicional reunión de coordinadores de doctorado, realizándose de forma híbrida. En esta ocasión también fueron invitados a participar por vía remota (o por vía presencial cubriendo sus propios gastos) los representantes de programas de maestría que estuvieran interesados(as). Contamos con la participación de 13 coordinadores y coordinadoras, 8 de forma presencial y 5 de forma remota.

La reunión inició con las palabras de bienvenida del Dr. Carlos Coello Coello, presidente de la Academia, seguido de un breve resumen de las reuniones de coordinadores anteriores, éste a cargo de la Dra. María del Pilar Gómez Gil, secretaria de la Academia. Posteriormente, cada uno de los participantes presentó durante 5 minutos sus respectivos programas.

La conferencia plenaria estuvo a cargo del Dr. Jesús Favela Vara del Departamento de Computación del CICESE, con el tema "Algunas reflexiones sobre los programas de doctorado en CC en México". El Dr. Favela, basado en su experiencia de muchos años como gestor y maestro de programas de posgrado en México, nos presentó una reflexión sobre varios temas fundamentales en este tópico, incluyendo la importancia de las becas para el mantenimiento de los posgrados y de la investigación en México, así como los retos que enfrentan actualmente los programas de posgrado. El Dr. Favela inició su plática mostrando gráficas del crecimiento que tuvieron los posgrados del 2000 al 2010, poniéndonos en contexto de la situación de la investigación en dicho periodo. Enseguida comentó algunas ideas sobre las oportunidades y retos actuales del posgrado, en cuanto al ingreso y egreso de estudiantes. Posteriormente el Dr. Favela comparó la situación actual con las condiciones anteriores e inició una discusión sobre qué estrategia podemos seguir para aprovechar el aprendizaje acumulado y las oportunidades actuales, a fin de promover la matrícula e investigación. Entre otras estrategias, se propuso trabajar revisando la factibilidad de crear programas de conversión doctorales, fomentar y buscar fuentes de financiamiento para las estancias de investigación, diseñar un taller de escritura y promover en los estudiantes de doctorado la participación temprana en la docencia.

Durante la reunión de coordinadores(as), también tuvimos un panel de discusión, titulado "Problemas fundamentales actuales de posgrados doctorales en México y estrategias de solución" a cargo de los secretarios de la Academia, el Dr. Hugo Terashima Marín y la Dra. María del Pilar Gómez Gil. Se inició la discusión con un resumen de las principales fortalezas y debilidades de los posgrados que fueron recabadas en un formulario previo al evento. Entre las principales fortalezas se nombraron:

- El buen nivel de madurez de varios de los programas, promovido en varios casos por la existencia del anterior Programa Nacional de Posgrados de Calidad (PNPC) de Conacyt.
- La existencia de programas de corte especial en algunas regiones del país, algunos con enfoques multidisciplinarios.
- La disponibilidad de becas otorgadas en base al talento académico en algunos posgrados.
- La posibilidad de algunos programas de contar con proyectos y colaboraciones internacionales.

Entre las principales debilidades y retos a vencer se nombraron:

- La necesidad de incrementar la matrícula; en algunos casos de forma particular se requieren estudiantes que tengan sus licenciaturas obtenidas de instituciones diferentes a la del posgrado.
- El desarrollar estrategias exitosas para poder elegir a las y los estudiantes que mejor se adapten al perfil del programa en cuestión.
- El fortalecer la vinculación con los diversos sectores de la sociedad, con proyectos orientados a las PRONACES y promover el impacto social de la investigación.

- El fortalecer los núcleos académicos atrayendo a investigadores(as) con el perfil requerido por el posgrado.
- El resolver la falta de recursos humanos, en la mayoría de los posgrados, para mantener una difusión constante.

Durante este panel, se presentaron algunas ideas sobre cómo conseguir y promover estancias y programas duales con universidades nacionales y extranjeras, lo que implica fomentar a los estudiantes en su preparación del idioma de inglés, así como talleres para escribir artículos, y algunas sugerencias para promover la participación de los estudiantes en la docencia.

Para finalizar, el Dr. Coello hizo una breve presentación del libro “La ruta de un doctorado exitoso”, coordinado por el Dr. Ramón Brena y editado por la Amexcomp. En la escritura del libro también participaron los doctores Jesús Favela, Francisco Cantú, Carlos Coello y Pablo Noriega. La revisión fue realizada por la Dra. Hanna Oktaba y un equipo de colaboradoras.

Figura 7. Participantes presenciales y en línea durante la reunión de coordinadores de posgrado AMEXCOMP.

Reunión Anual de la AMEXCOMP

Por Dr. Carlos A. Coello Coello.

Durante el pasado 13 de octubre se llevó a cabo la Reunión Anual de la Academia Mexicana de Computación (AMEXCOMP), de manera presencial en el Hotel Hacienda Vista Hermosa, ubicado en Puente de Ixtla, Morelos, y se transmitió vía Zoom a los miembros que se registraron al mismo. Al evento asistieron 118 miembros (73 presenciales y 45 virtuales). Como ya es costumbre, el día 12 de octubre por la tarde se llevó a cabo la Reunión de Mujeres de la AMEXCOMP, también en modo híbrido.

Figura 8. Reunión Anual de la AMEXCOMP.

Durante la reunión se llevó a cabo la Asamblea anual de la AMEXCOMP, en la cual se proporcionó el informe de labores de la Presidencia y se tomaron decisiones importantes, tales como la elección de nuevos miembros para la Comisión de Membresía y para la Comisión de Premios. Los nuevos miembros de la Comisión de Membresía son: el Dr. Giner Alor Hernández, la Dra. Yolanda Margarita Fernández Ordóñez y la Dra. María Yasmín Hernández Pérez. Los nuevos miembros de la Comisión de Premios son: la Dra. María Lucía Barrón Estrada, el Dr. Carlos Alberto Reyes García y el Dr. Sergio Rajsbaum.

Figura 9. Ceremonia de Inauguración del año académico 2022.

Durante la Ceremonia de Inauguración del año académico 2022, el invitado de honor fue el Dr. Ramsés Mena Chávez, quien actualmente se desempeña como director del Instituto de Investigaciones en Matemáticas Aplicadas y Sistemas de la Universidad Nacional Autónoma de México (IIMAS-UNAM). En esta misma ceremonia se entregaron los diplomas a los nuevos miembros de la AMEXCOMP.

Dentro de las actividades de la Reunión Anual, se realizaron las tradicionales presentaciones de 30 segundos de los miembros que asistieron presencialmente. Además, se llevó a cabo la entrega del Premio Nacional de Computación al Dr. Sergio Rajsbaum, del Instituto de Matemáticas de la Universidad Nacional Autónoma de México. El Dr. Rajsbaum impartió también la conferencia titulada “El teorema de imposibilidad de Arrow, una perspectiva desde la computación distribuida”.

Figura 10. Algunas de las actividades de la Reunión Anual.

También se contó con la conferencia magistral de la Dra. Bernadette Bouchon-Meunier, quien es directora de Investigación Emérita en el Centro Nacional de Investigación Científica con sede en la Universidad de la Sorbonne, en París, Francia. Su conferencia se tituló "Fuzzy Artificial Intelligence for Explainable Artificial Intelligence".

Figura 11. Conferencia Magistral de la Dra. Bernadette Bouchon-Meunier.

De entre las actividades informadas durante la Reunión Anual, destaca el hecho de que este año ingresaron 29 nuevos miembros (20 miembros adherentes y 9 miembros regulares).

Además, se concluyó y publicó el libro "Matemáticas Básicas: de lo intuitivo y concreto a lo abstracto", Segunda Edición, escrito por Leonardo Romero Muñoz y Moisés García Villanueva. Esta segunda edición corrige algunos pequeños errores de la primera edición (publicada también por la AMEXCOMP), y además incluye los capítulos de sistemas de ecuaciones, polinomios y desigualdades. Se agregó también un prefacio a la segunda edición explicando en detalle los cambios. Los capítulos nuevos son temas estándar en libros de aritmética y álgebra, como antecedentes para abordar el cálculo diferencial e integral y el álgebra lineal. De esta manera, el libro contiene de manera más completa los temas fundamentales de matemáticas, desde la primaria hasta el inicio de la universidad.

También se habló sobre el libro "La ruta a un doctorado exitoso en computación", escrito por varios investigadores, coordinados por el Dr. Ramón Brena, el cual se encuentra actualmente en proceso de revisión para su posible publicación por la AMEXCOMP. Adicionalmente, la Presidencia habló de los planes para 2023.

Figura 12. Asistentes de la Reunión Anual de la AMEXCOMP.

En general, el evento fue muy ameno y permitió restablecer la convivencia entre los miembros de la AMEXCOMP (a la fecha, contamos con 310 miembros) que nos honraron con su valiosa presencia.

Recordando a...

Por Dr. Carlos A. Coello Coello.

Figura 13. Butler W. Lampson.

Butler W. Lampson nació en Washington, DC y asistió a la *Lawrenceville School*, que es una escuela de élite ubicada a unos 10 km de Princeton, Nueva Jersey. En 2009, Lampson recibió el *Aldo Leopold Award*, que es la distinción más importante que esta escuela otorga a sus egresados.

El primer contacto de Lampson con una computadora ocurrió durante la preparatoria cuando uno de sus compañeros de clase consiguió permiso de usar una IBM 650 subutilizada en Princeton que tenía una memoria de 2000 palabras de 10 dígitos decimales cada una y cuyo único sistema de entrada/salida era una lectora de tarjetas perforadas.

Mientras cursaba la licenciatura en física en la Universidad de Harvard (de donde se graduó *magna cum laude* en 1964), Lampson aprendió a programar en APL, un lenguaje de programación que diseñó Kenneth Iverson, quien había estado en Harvard en un receso sabático. En esa época, no había una implementación de este lenguaje de programación y se decía que Iverson no quería una, a fin de no comprometer su estructura original. Mientras era todavía un estudiante de licenciatura, Lampson programó una PDP-1 para analizar fotografías del Acelerador de Electrones de Cambridge y escribió un editor para la pantalla del tipo un-punto-a-la-vez de la PDP-1.

Lampson ingresó en 1964 a la Universidad de California en Berkeley para cursar un doctorado en física. Sin embargo, en la *Fall Joint Computer Conference* realizada ese año en San Francisco, se topó con Steve Russell del Instituto Tecnológico de Massachusetts (MIT por sus siglas en inglés), quien le contó sobre el proyecto **Genie**, que se encontraba oculto tras una puerta sin ningún tipo de señalamiento en Berkeley. Lampson se interesó en este proyecto y no tardó en incorporarse a él. Ahí conoció a Peter Deutsch y Mel Pirtle (el Responsable Técnico de Genie). Poco tiempo después, Lampson decidió abandonar la física para incribirse en el doctorado en ingeniería eléctrica y ciencias de la computación de Berkeley, del que se graduaría en 1967.

El proyecto Genie modificó una minicomputadora SDS 930 de *Scientific Data Systems* (SDS), para crear el primer sistema de tiempo compartido con una interacción caracter por caracter. Tiempo después, SDS comercializó esta computadora como la SDS 940, que se convertiría en la primera computadora de tiempo compartido de uso general disponible comercialmente. Lampson escribió partes del sistema operativo y varios lenguajes de programación para esta computadora. De los desarrollos de Lampson destacan Cal, que era un lenguaje interactivo para cálculo numérico, derivado del lenguaje Joss, desarrollado por Cliff Shaw, así como en lenguaje de sistemas QSPL que desarrolló junto con Peter Deutsch.

En Berkeley, Lampson diseñó también el sistema de tiempo compartido Cal para una CDC 6400 junto con Jim Gray, Charles Simonyi, Howard Sturgis y Bruce Lindsay. Curiosamente, todos ellos se volvieron famosos años después. Este fue el primer sistema basado en capacidades en contar con una comunidad de usuarios reales. Este sistema mostró que los sistemas basados en capacidades no son una buena base para desarrollar sistemas de seguridad de largo plazo.

Los investigadores que trabajaron en Genie no pudieron determinar cómo construir un segundo sistema mucho más grandioso en Berkeley, y en 1969 crearon la *Berkeley Computer Corporation* (BCC) para ese fin. Dos años y cuatro millones de dólares después, la BCC solo contaba con un sistema funcional antes de irse a la quiebra. Lampson diseñó y codificó la mayor parte del microcódigo de este sistema operativo y trabajó en el lenguaje de programación de sistemas SPL. En esa época, también propuso el modelo de matriz de acceso para la seguridad de los sistemas de cómputo, unificando las ideas de los sistemas basados en capacidades con las de las listas de control de acceso.

Por cuestiones azarosas, mientras BCC se encaminaba al precipicio, arrancaba el *Xerox Palo Alto Research Center* (PARC), y Bob Taylor encontró al núcleo de su Laboratorio de Ciencias de la Computación (CSL por sus siglas en inglés) en BCC: Chuck Thacker, Peter Deutsch, Charles Simonyi, Jim Mitchell y Butler Lampson. El objetivo de este equipo era inventar y desplegar la oficina del futuro. Nadie en el equipo sabía el significado real de este objetivo, pero se sentían muy motivados para desarrollar nuevas tecnologías.

En esa época, la computadora estándar de investigación era la DEC PDP-10 con su sistema operativo Tenex, pero como Xerox acababa de comprar SDS, que era el principal competidor de DEC, los miembros del CSL tuvieron que construir de cero una PDP-10 en 1971.

Dos años después, la misma tecnología y herramientas se transfirieron a la Alto, que fue la primera computadora personal moderna reconocible, y que estaba muy adelantada a su época. La Alto tenía una pantalla con una resolución de 600×800 pixeles (en blanco y negro, con solo un bit por pixel, pues ya así consumía la mitad de los 128 Kilobytes de la computadora), un disco duro, una tarjeta de red local Ethernet y una impresora láser.

Lampson diseñó parte de la computadora, escribió su sistema operativo y, junto con Ron Rider, diseñó la electrónica y el software para el prototipo de impresora láser que construyó Gary Starkweather, que se adaptó posteriormente para crear la impresora láser Xerox 9700. El software de la Alto incluía Bravo, que fue el primer editor de textos del tipo "*what you see is what you get*", el cual fue diseñado por Lampson y Charles Simonyi. Bravo inspiró el desarrollo del Word de Microsoft. Butler y Alan Key diseñaron el esquema de lenguaje máquina usado por Smalltalk, Mesa y las versiones posteriores del software de la Alto que se distribuyeron en universidades, la Casa Blanca y varios otros sitios de prueba.

Lampson también diseño buena parte del lenguaje de programación de sistemas Mesa, particularmente el mecanismo de procesos, que posteriormente se convertiría en la base para la mayoría de los sistemas modernos de hilos (*threads*). Los módulos de Mesa condujeron al trabajo inicial que Lampson desarrolló con Eric Schmidt en torno a sistemas para construir sistemas de software de gran tamaño. Otro derivado de este trabajo fue Euclid, el primer lenguaje diseñado específicamente para ser utilizado para la verificación de programas. Mesa condujo después a Cedar, que fue un intento solo parcialmente exitoso por combinar las virtudes de Mesa y LISP. Butler realizó algo del diseño de Cedar y escribió una descripción muy detallada del lenguaje. Todo este trabajo llevó a Lampson a escribir un artículo titulado "*Hints for computer system design*" (publicado en 1983), que se volvería una publicación sumamente influyente en años posteriores.

En PARC, Lampson se involucró en diversos proyectos. Por ejemplo, fue el líder del diseño de Dorado, la cual fue una computadora personal mucho más poderosa que la Alto. También fue el líder del diseño de *Wildflower*, que era una versión más económica de la Alto. *Wildflower* fue la base para el hardware de la computadora Star, que fue el primer sistema de automatización de oficinas de Xerox, el cual fue construido por un grupo de desarrollo surgido de PARC liderado por David Liddle. Butler también diseñó la electrónica de la Dover, que era una versión mucho más económica de la impresora láser, la cual fue utilizada en Xerox y en varias universidades.

Con Paul Rovner y otros, Lampson diseñó el Modula 2+, que era una extensión del lenguaje de programación Modula 2, diseñado por Niklaus Wirth, el cual proporciona liberación automática de almacenamiento, resolución de tipos en tiempo real, excepciones y concurrencia. Este lenguaje de programación se ha utilizado para escribir más de un millón de líneas de código. Lampson trabajó con Rod Burstall en lenguajes de programación usados para describir la forma en que se construye un sistema a partir de sus componentes. Después de varios arranques en falso, este trabajo finalmente condujo al sistema Vesta, creado por Roy Levin, Yuan Yu y otros, el cual sigue siendo un referente en cuanto a la construcción eficiente y confiable de grandes sistemas de software con muchos desarrolladores y muchas versiones.

Hacia 1983, el grupo más influyente dentro de PARC comenzaba a desintegrarse y muchos de sus mejores investigadores, incluyendo a Lampson, se fueron a trabajar al *Systems Research Center* (SRC) de *Digital Equipment Corporation* (DEC). En esa época, Lampson vivía en Filadelfia, donde su esposa Lois daba clases en la Universidad

de Pensilvania. Sus intereses de aquel entonces se centraron en el desarrollo de sistemas de correo electrónico. El trabajo realizado previamente por Mike Schroeder, Roger Needham, Andrew Birrell y otros en el sistema de correo electrónico distribuido *Grapevine*, condujo a un diseño y una especificación formal de un servicio de nombres distribuido, descentralizado y tolerante a fallas, el cual se conocería después como "consistencia eventual" y sería muy utilizado. Las preocupaciones en torno a cómo dotar de seguridad al correo electrónico, condujeron al desarrollo del primer esquema para autenticación distribuida. Otro proyecto importante de SRC en el que Lampson se vio involucrado fue el de dos redes de área local de alta velocidad con una combinación sin precedentes de velocidad, disponibilidad y seguridad. Su trabajo en autenticación lo llevó a publicar una serie de artículos que sentaron las bases de la seguridad en los sistemas distriubidos.

En 1987, Lampson se mudó a Cambridge, Massachusetts, porque su esposa se volvió profesora de la Escuela de Medicina de la Universidad de Harvard. Siguió trabajando para DEC, pero también se volvió Profesor Adjunto del Departamento de Ingeniería Eléctrica y Ciencias de la Computación del MIT. Ahí trabajó con Nancy Lynch en especificaciones formales y demostraciones matemáticas para el *TCP connection establishment* (también conocido como mensajes a-lo-mucho-una-vez) y para el protocolo Praxos de Leslie Lamport. También trabajó para persuadir a la comunidad de sistemas que Praxos y las máquinas de estados replicados de Lamport eran la forma correcta de construir sistemas tolerantes a fallas, y el tiempo le dio la razón, pues este sistema se comenzó a adoptar ampliamente. También se interesó en estudiar cómo evoluciona la computación, analizando cómo los componentes de software han fallado y han tenido éxito a lo largo del tiempo. Con David Tennenhouse, analizó la combinación de factores técnicos y económicos que conducen o impiden la evolución de las telecomunicaciones.

En 1995, Butler se incorporó a *Microsoft Research*, donde trabajó en el software para la tableta PC, en seguridad y en programación de usuario final para diversas aplicaciones.

La carrera de Lampson ha sido muy prolífica, pues ha realizado contribuciones importantes a la arquitectura de computadoras, a las redes de área local, a los lenguajes descriptores de páginas, a los sistemas operativos, a los lenguajes de programación y su semántica, al cómputo tolerante a fallas, al procesamiento de transacciones y a la seguridad en los sistemas de cómputo, entre muchas otras áreas.

Butler Lampson recibió el *ACM Turing Award* en 1992. Adicionalmente, ha recibido muchos otros premios, de entre los que destacan el *ACM Software Systems Award* (por su trabajo en la Alto), el *IEEE Computer Pioneer Award*, la Medalla *IEEE von Neumann*, el *Computer History Museum Fellows Award* y el Premio *Draper* de la Academia Nacional de Ingeniería de Estados Unidos. También es miembro tanto de la Academia Nacional de Ciencias como de la Academia Nacional de Ingeniería de Estados Unidos, y tiene Doctorados Honoris Causa de la Universidad de Boloña y de la *Eidgenössische Technische Hochschule*, Zürich. Tres de sus artículos han ganado el *SIGOPS* (*ACM Special Interest Group on Operating Systems*) *Hall of Fame Award*.