

Efemérides del trimestre

Jack Dongarra

Nació el 18 de julio de 1950. Ganador del ACM Turing Award en 2021. Realizó contribuciones pioneras a los algoritmos numéricos y a las bibliotecas (como Netlib) que permitieron que el software para cómputo de alto desempeño pudiera estar a la altura de las mejoras exponenciales al hardware que han ocurrido en las últimas cuatro décadas.

Alfred Vaino Aho

Nació el 9 de agosto de 1941. Ganador del ACM Turing Award en 2020. Desarrolló algoritmos fundamentales y la teoría detrás de la implementación de varios lenguajes de programación. Además, publicó libros que tuvieron una enorme influencia en la educación de muchas generaciones de expertos en computación en todo el mundo.

Tim Berners-Lee

Nació el 8 de junio de 1955. Ganador del ACM Turing Award en 2016. Inventó la World Wide Web, el primer navegador web y los protocolos y algoritmos fundamentales que permitieron que escalara el internet.

Kristen Nygaard

Nació el 27 de agosto de 1926. Ganador del ACM Turing Award en 2001. Desarrolló, junto con Ole-Johan Dahl, las ideas fundamentales de la programación orientada a objetos, a través del diseño de los lenguajes de programación Simula I y Simula 67.

E. Allen Emerson

Nació el 2 de junio de 1954. Ganador del ACM Turing Award en 2007. Desarrolló, junto con Edmund Clarke y Joseph Sifakis, el Chequeo de Modelos (Model-Checking) que se convirtió en una tecnología muy efectiva de verificación que es ampliamente utilizada en la actualidad tanto en la industria del software como en la del hardware.

Saludo Editorial

Es un gran placer darles la bienvenida al tercer número de nuestro Boletín, que inicia con un artículo del Dr. José Alfonso Aguilar Calderón titulado “Evolución y Tendencia en el Desarrollo de Software”.

En una breve nota el Dr. Raúl Rojas habla sobre “El primer programa de computadora” de la historia y posteriormente, el Dr. Hugo Terashima Marín presenta una breve reseña sobre el 2023 *SIGEVO Outstanding Contribution Award* que tuvo el honor de recibir el 19 de julio pasado en Lisboa, Portugal.

Los Dres. Ramón Zatarain Cabada, Héctor Manuel Cárdenas López y Hugo Jair Escalante presentan una reseña de su libro titulado “*Multimodal Affective Computing, Technologies and Applications in Learning Environments*”, el cual fue publicado recientemente por Springer.

Rafael Gamboa Hiraes nos proporciona una breve reseña del libro “*Konrad Zuse’s Early Computers*”, escrito por Raúl Rojas González. Los Dres. Allan Antonio Flores Fuentes, Juan Fernando García Mejía y Everardo Efrén Granda Gutiérrez nos proporcionan una reseña del “Seminario de Divulgación Aportaciones de la Inteligencia Artificial en la Sociedad”, el cual se llevó a cabo el pasado 30 de mayo en Atlacomulco, Estado de México.

El Dr. Luis Enrique Sucar presenta una breve nota sobre el “Taller sobre Descubrimiento Causal” realizado del 19 al 21 de junio pasado en el INAOE.

La Dra. María del Pilar Gómez Gil habla sobre “Los 8 principios de la ACM para el desarrollo, implementación y uso de Inteligencia Artificial (IA) Generativa”, los cuales son el producto del trabajo de decenas de especialistas de Europa y Estados Unidos.

En nuestra columna titulada “Recordando a . . .”, hablamos en esta ocasión de Timothy John Berners-Lee, que es el inventor de la World Wide Web.

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

Evolución y Tendencia en el Desarrollo de Software.

Por:

Dr. José Alfonso Aguilar Calderón

ja.aguilar@uas.edu.mx

Es una realidad que la tecnología avanza a pasos agigantados. La humanidad ha evolucionado gracias a la tecnología, como todo en esta vida, con sus puntos a favor y sus puntos en contra. Sin embargo, se considera positivo este avance para el ser humano. La evolución tecnológica y de comunicaciones se debe en una buena parte gracias a la creación del Internet y al desarrollo de la World Wide Web (WWW). Recordemos aquel proyecto que nació en el CERN, el Centro Europeo de Física Nuclear, en Ginebra (Suiza), de la mano del ingeniero y físico británico Sir Tim Berners-Lee como un sistema de intercambio de datos entre los científicos que trabajaban ahí [1]. La WWW es un servicio que opera sobre Internet, como lo es el tan utilizado hoy en día correo electrónico, y ha permitido el desarrollo de software para ese entorno, así como el surgimiento de nuevos lenguajes de programación (o combinaciones de ellos) para desarrollar aplicaciones que operen en la web de una forma eficiente. Pero, antes de todo esto, una pieza clave fue el desarrollo de software.

La tecnología hoy en día, para funcionar y/o administrarse, necesita de un software. La Ingeniería de Software (IS) es una disciplina muy antigua, tanto así que se considera que Alan Turing fue el primero que desarrolló una teoría sobre el software en un ensayo de su autoría en el año de 1935-1936 sobre números computables titulado Computing Machinery and Intelligence [2]. El término Ingeniería de Software se utilizó por primera vez en la década de los 50s por John W. Tukey y en los inicios de los años 60s. En esa época, la OTAN (Organización del Tratado del Atlántico Norte), a través de su división de ciencias, fue el patrocinador de dos conferencias sobre la IS, en 1968 y en 1969 [3].

Esto permitió dar el impulso y difusión que se necesitaba para su exposición global. Algunos investigadores y, por qué no, historiadores, consideran éste el inicio de la IS como profesión. Además, se considera que fue impulsada por una crisis que ocurrió entre 1960 y 1980, conocida en la literatura como la Crisis del Software [4], durante este periodo los proyectos de desarrollo sobrepasaron su planificación en tiempo, esfuerzo y costo. Incluso, algunos proyectos causaron pérdidas de vidas humanas.

El desarrollo de software ha evolucionado, desde sus inicios con los primeros lenguajes de programación, pasando por la programación procedimental, orientada a eventos, orientada a objetos, hasta la actualidad con el desarrollo de software dirigido por modelos (Model-Driven Development) [5], conocido también como la antesala o fundamentos del desarrollo sin código (No-Code) y Low-Code (código bajo), así como con el desarrollo de código fuente asistido por la muy de moda Inteligencia Artificial (IA).

La evolución en el desarrollo de software ha llevado al surgimiento de lo que se conoce como los desarrolladores ciudadanos o citizen developers [6]. Son trabajadores que no tienen un conocimiento especializado en el desarrollo de software, sea para el entorno de escritorio, web o móvil, que no forman parte de un equipo de desarrollo o

Consejo Directivo AMexComp

Presidente:

Dr. Carlos Artemio Coello Coello

Vicepresidente:

Dr. Eduardo F. Morales Manzanares

Tesorero:

Dr. Efrén Mezura Montes

Secretaria:

Dra. María del Pilar Gómez Gil

Secretario:

Dr. Hugo Terashima Marín

Vocal:

Dra. Marcela Quiroz Castellanos

Comité Editorial del Boletín AMexComp

Dr. Carlos Artemio Coello Coello

Dra. Marcela Quiroz Castellanos

Dra. María del Pilar Gómez Gil

Esperamos sus contribuciones y avisos al correo del boletín:

boletin@amexcomp.org.mx

las cuales son muy importantes para mantener vivo el boletín.

incluso del departamento de tecnologías de la información de una empresa, pero que pueden crear software para uso empresarial o personal. El concepto de los desarrolladores ciudadanos surge con el desarrollo automatizado y semiautomatizado de software. Si bien es verdad que un desarrollador de software profesional no quedará desplazado por un desarrollador ciudadano, es importante mencionar que cierto tipo de aplicaciones móviles, de escritorio o web, si pueden ser realizadas por los desarrolladores ciudadanos. Para comprender mejor esto, explicaremos el significado de Low-Code y No-Code. El primero se utiliza para generar parcialmente un software, es para la semiautomatización. Por ejemplo, si necesito crear una aplicación robusta o que repita patrones de comportamiento, es posible generar gran parte del código fuente del software a desarrollar, pero requiere que alguien con al menos un conocimiento básico del código finalice el proceso de construcción e implementación. La segunda, por su parte, permite crear aplicaciones completamente funcionales sin escribir una sola línea de código fuente, claro, de baja complejidad que no requieran conexiones a sistemas software de terceros, por ejemplo, el conectarse con un servicio de facturación o de cobro bancario. Empresas internacionales han presentado herramientas bajo el concepto de Low-Code (Microsoft con Power Apps, WebRatio y Quickbase) y No-Code (Webflow, Wordpress, Xano) que permiten crear software rápidamente. En la Figura 1, se observa la diferencia entre el Low-Code y el desarrollo de software mediante la escritura de código tradicional [7]. Tanto el Low-Code y el No-Code, normalmente cuentan con un entorno de trabajo con un editor gráfico que permite solo configurar mediante asistentes o arrastrar controles para diseñar y en un proceso background se genera el código fuente de la aplicación a desarrollar, ya sea en su totalidad o parcialmente.

La implementación de un proceso de desarrollo de software mediante el Low-Code y el No-Code son opciones sólidas que ayudarán a crear e implementar software rápidamente y a un menor costo en comparación con la programación tradicional. En este contexto, es importante saber cuál elegir para implementar. Se recomienda elegir una plataforma No-Code si se tienen recursos limitados y se necesita crear aplicaciones de poca complejidad lo más rápido posible, siendo consciente de las limitaciones funcionales del No-Code. En cambio, es conveniente seleccionar una plataforma de Low-Code si sus aplicaciones son sencillas por naturaleza, pero tienen complejidades que podrían requerir el trabajo de desarrolladores adicionales, por ejemplo, conexiones con software de terceros, o si desea acelerar tanto a los desarrolladores profesionales como a los usuarios empresariales. Para más información sobre los conceptos de Low-Code y No-Code se refiere al lector a este artículo: Challenges of low-code/no-code software development: A literature review. En la figura 2, tomada de [8], se observa el landscape del No-Code. En ella se muestra un panorama de los distintos campos de aplicación en los que existen herramientas No-Code. Por otra parte, se encuentra el desarrollo de software asistido por IA. Este cambio de paradigma está impulsado por Codex, un modelo de aprendizaje automático de la empresa de investigación y desarrollo de IA OpenAI [9].

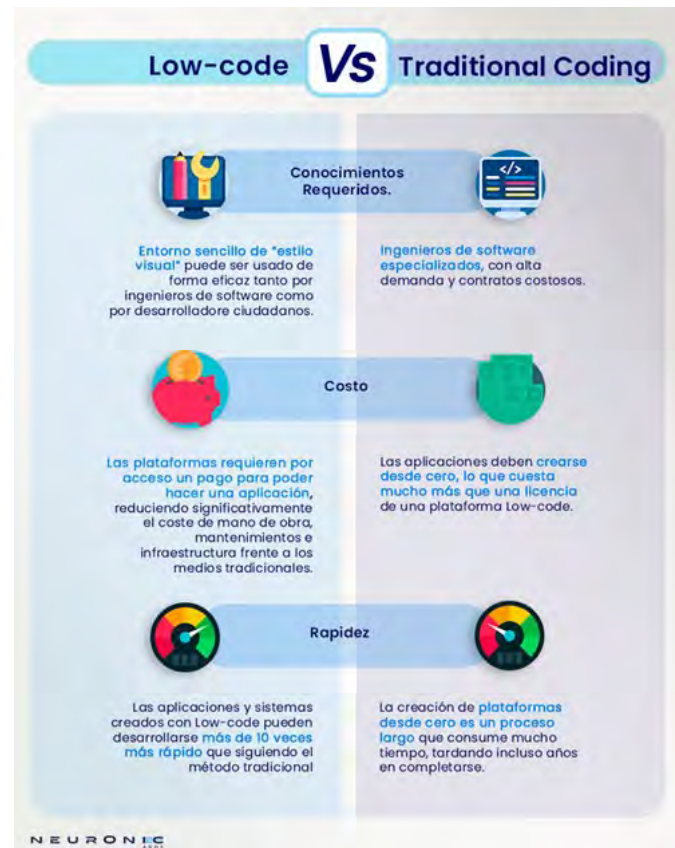


Figura 1. El Low-Code vs la escritura de código fuente tradicional para desarrollar software.



Figura 2. Ejemplos de aplicaciones No-Code en distintas áreas.

La importancia de Codex radica en el hecho de que puede traducir comandos de lenguaje natural en código en más de una docena de lenguajes de programación. Codex desciende de GPT-3 [10], el modelo de lenguaje natural autorregresivo de OpenAI. A continuación, se mencionan algunos ejemplos de aplicaciones de la IA en la generación de código fuente: AlphaCode [11], un nuevo sistema de inteligencia artificial para desarrollar código desarrollado por DeepMind que entiende el lenguaje natural y lo traduce a un lenguaje determinado. Se encuentra también GitHub Copilot [12], el cual es una herramienta que utiliza también el modelo OpenAI Codex para sugerir código y funciones completas en tiempo real, directamente desde tu editor. Amazon CodeWhisperer [13], que permite a los desarrolladores crear aplicaciones de manera rápida y segura. Proporciona sugerencias de código en tiempo real que van desde fragmentos hasta funciones completas, seguimiento de referencias y análisis de seguridad. Por último, Code GPT [14] es una extensión que nos permite utilizar la API (Application Programming Interface) REST (REpresentational State Transfer) oficial de OpenAI en Visual Studio Code. En la Figura 3, se observa la tendencia en la aplicación de la IA como auxiliar en el desarrollo de software. Se observan líneas clave que permitirán evolucionar la forma en que construimos software, de tal forma que también será necesaria esa evolución teórica-metodológica en la Ingeniería de Software.

Finalmente, como dato interesante, Gartner predice que se prevé que las plataformas de aplicaciones de código bajo (Low-Code) sean el componente más grande del mercado de tecnología de desarrollo de código bajo, con un crecimiento del 25 % hasta alcanzar casi 10 mil millones de dólares en 2023. Además, que para 2026, los desarrolladores fuera de los departamentos de tecnología formales representarán al menos el 80 % de la base de usuarios de herramientas de desarrollo de código bajo, frente al 60 % en 2021 [15], es decir, los ciudadanos desarrolladores.



Figura 3. Tendencias en la aplicación de la IA como auxiliar en el desarrollo de software.

Referencias

[1] Sitio Web Oficial World Wide Web Consortium (2023): <https://www.w3.org/People/Berners-Lee/>

[2] Artículo Web (2023): Alan Turing y el nacimiento de la computación: <https://blogs.uoc.edu/informatica/alan-turing-ii-el-nacimiento-de-la-computacion/>

[3] Randell, Brian. (1979). Software Engineering: As it was in 1968.. 1-10.

[4] Martínez, R. N. (2015). El proceso de desarrollo de software. IT Campus Academy.

[5] Pastor, O., España, S., Panach, J. I., & Aquino, N. (2008). Model-driven development. Informatik-Spektrum, 31, 394-407.

[6] Lebens, M., Finnegan, R. J., Sorsen, S. C., & Shah, J. (2022). Rise of the citizen developer. Muma Business Review, 5, 101-111.

[7] Artículo Web (2023): ¿Qué es el Low-Code?: <https://neuronic.com.ar/low-code-que-es.html>

[8] Artículo Web (2023): No-Code Landscape: <https://community.nasscom.in/communities/it-services/rapidly-expanding-no-code-landscape>

[9] [9] Sitio Web Oficial OpenAI (2023): <https://openai.com/>

[10] Sitio Web Oficial de GPT-3 (2023): <https://openai.com/product>

- [11] [11] Sitio Web Oficial de AlphaCode (2023):<https://alphacode.deepmind.com/>
- [12] Sitio Web Oficial de GitHub Copilot (2023):<https://github.com/features/copilot>
- [13] Sitio Web Oficial de Amazon CodeWhisperer (2023): <https://aws.amazon.com/es/codewhisperer/>
- [14] Sitio Web Oficial de CodeGPT (2023): <https://codegpt.co/>
- [15] Artículo Web (2023): Gartner Forecasts Worldwide Low-Code Development Technologies Market to Grow 20 % in 2023.<https://www.gartner.com/en/newsroom/press-releases/2022-12-13-gartner-forecasts-worldwide-low-code-development-technologies-market-to-grow-20-percent-in-2023>

El primer programa de computadora

Por:

Raúl Rojas

Universidad Libre de Berlín

Si hacemos una búsqueda en Internet o en Wikipedia acerca del primer programa jamás escrito para una computadora, encontramos que éste sería un programa de 1843 que calcula los números de Bernoulli (atribuido a Ada Lovelace, pero probablemente escrito en realidad por Charles Babbage). Con la apertura al público del archivo póstumo de los escritos de Babbage, por parte del Museo de Ciencia de Londres, podemos corregir este error.

Resulta que Babbage escribió una serie de 26 programas para la Máquina Analítica (MA) a partir de agosto de 1837. La MA fue diseñada por el inventor inglés y hubiera sido la primera computadora del mundo de haber sido terminada. El diseño incluía un procesador y una memoria, coordinados ambos por sendas lectoras de tarjetas perforadas. El procesador podía ejecutar las cuatro operaciones aritméticas básicas mientras que la memoria podía proporcionar los dos argumentos para la operación y recibir el resultado para ser almacenado.

El título de este primer programa de Babbage es “Notaciones y Cálculos” y la primera línea proporciona su fecha: “No. 1. 4 de agosto de 1837”. El archivo Babbage lista este programa como "BAB L1". El programa encuentra las dos soluciones de un sistema de dos ecuaciones lineales en dos variables:

	Computation	Code
1	$b'a$	$v_7 = v_5 \cdot v_1$
2	$b'c$	$v'_1 = v_5 \cdot v_3$
3	ba'	$v'_3 = v_2 \cdot v_4$
4	bc'	$v'_2 = v_2 \cdot v_6$
5	$bc' - b'c$	$v'_6 = v'_2 - v'_1$
6	$b'a - ba'$	$v''_2 = v_7 - v'_3$
7 *	$\frac{bc' - b'c}{b'a - ba'}$	$v'_4 = \frac{v'_6}{v'_2}$
8		$v''_1 = v_1 = a$
9		$v''_2 = v_2 = b$
10		$v''_3 = v_3 = c$
11	ax	$v''_5 = v''_1 \cdot v'_4$
12	$-c - ax$	$v'''_1 = -v''_3 - v''_5$
13 *	$\frac{-c - ax}{b}$	$v'''_5 = \frac{v'''_1}{v'_2}$

$$ax + by + c = 0$$

$$a'x + b'y + c' = 0$$

Las soluciones son:

$$x = \frac{bc' - b'c}{b'a - ba'}, y = \frac{-ax - c}{b}$$

Figura 4. Tabla 1: El programa de Babbage para encontrar las soluciones de un sistema de dos ecuaciones lineales. El programa no chequea si los denominadores en las divisiones (pasos 7 y 13) son cero o no.

suponiendo que los denominadores no son cero, en las dos divisiones.

Babbage llamaba a las localidades de memoria variables y, por ejemplo, el contenido de la dirección 5 en la memoria se denotaba por v_5 . El programa de Babbage asume que hemos almacenado los tres coeficientes de la primera ecuación en las localidades de memoria 1, 2 y 3, y los coeficientes de la segunda ecuación en las localidades 4, 5, y 6. Bajo esos supuestos, la Tabla 1 que se muestra abajo, resuelve el sistema de ecuaciones en 13 pasos. El pseudocódigo del programa se muestra en la tercera columna, mientras que la segunda columna es un comentario que revela qué cálculo se ha realizado. Babbage le agregaba un apóstrofo a una variable cuyo contenido era modificado por una operación aritmética (lo cual no es realmente necesario). El programa calcula la solución para x en siete pasos, luego restaura los coeficientes de la primera ecuación en los pasos 8, 9 y 10, para finalmente encontrar el valor de y . El valor de x queda almacenado en la dirección cuatro de memoria y el valor de y en la dirección 5.

La explicación de este primer programa de la historia es un poco más complicada de lo aquí brevemente expuesto, pero el texto completo de la descripción del programa se puede encontrar en: R. Rojas, “The First Computer Program”, <https://doi.org/10.48550/arXiv.2303.13740>. El lector interesado puede encontrar en esta referencia todos los fascinantes detalles de éste, el primer programa de la historia.

El Dr. Carlos A. Coello Coello recibió el 2023 SIGEVO Outstanding Contribution Award

Por:

Dr. Hugo Terashima Marín

Tecnológico de Monterrey, Escuela de Ingeniería y Ciencias

El pasado 19 de julio, el Dr. Carlos A. Coello Coello recibió el ACM *Special Interest Group on Evolutionary Algorithms (SIGEVO) Outstanding Contribution Award* de manos del Dr. Franz Rothlauf, quien presidía el *SIGEVO Executive Committee*. Este premio se otorga por contribuciones verdaderamente distintivas a la computación evolutiva, realizadas en un período de al menos 15 años".

El evento se llevó a cabo en Lisboa, Portugal, en el marco de la *2023 Genetic and Evolutionary Computation Conference (GECCO'2023)* y reunió a varios de los investigadores que recibieron este reconocimiento en años pasados, tales como: Marc Schoenauer, Darrell Whitley, Kenneth de Jong, Una May O'Reilly, Erik Goodman y Wolfgang Banzhanf.



En la ceremonia de este año se distinguió con el mismo reconocimiento a la Dra. Emma Hart, quien actualmente es la Editora-en-Jefe de la revista *Evolutionary Computation* (publicada por MIT Press).



Felicitaciones al Dr. Coello Coello por tan importante distinción y que nos enorgullece a todos en la Comunidad de Computación en México..

Eventos

TNDI 2023

Taller Nacional de Drones
Inteligentes 2023

5 y 6 de octubre de 2023: Evento.

CIMPS 2023

12th International Conference on
Software Processes Improvement

18-20 de Octubre, 2023: Evento.

SIMDIA 2023

Second Ibero-American Symposium
of Master and Doctorate in Artificial
Intelligence

9-10 de Noviembre, 2023: Simposio

ICSC-CITIES 2023

VI Ibero-American Congress of Smart
Cities

13-17 de Noviembre, 2023: Presential
and on-line congress.

MICAI 2023

22nd Mexican International
Conference on Artificial Intelligence

13-28 de Noviembre, 2023: On-site
conference.

Mérida, Yucatan, México.

ISCFI 2023

10th International Conference on
Soft Computing & Machine
Intelligence

Noviembre 25-26 : Virtual conference.
Mexico City, Mexico

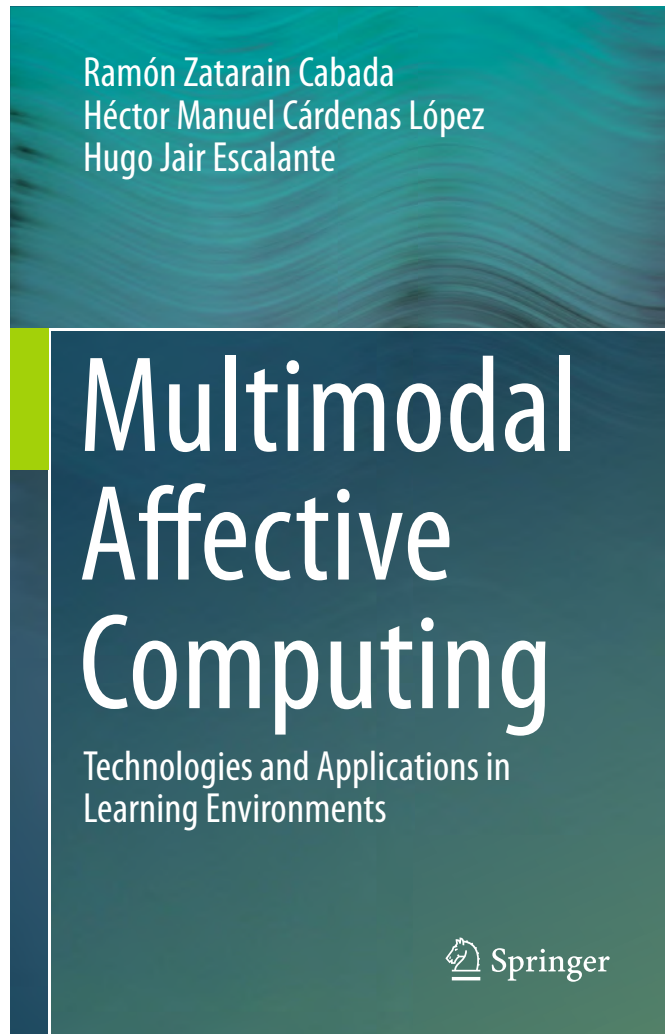
Reseña del libro: **Multimodal Affective Computing, Technologies and Applications in Learning Environments**

Por:

Ramón Zatarain Cabada (TecNM-Instituto Tecnológico de Culiacán)

Héctor Manuel Cárdenas López (TecNM-Instituto Tecnológico de Culiacán)

Hugo Jair Escalante (Instituto Nacional de Astrofísica, Óptica y Electrónica)



La exploración de nuevos conocimientos ha sido un motor que ha acelerado el desarrollo de nuevas áreas tecnológicas y nichos de la ciencia que a su vez abren oportunidades y promueven el descubrimiento de más conocimientos. Algunos nuevos conocimientos son más útiles e impulsan más notoriamente el desarrollo de áreas tecnológicas y científicas que buscan innovaciones y nuevos lineamientos para, a su vez, tener avances más significativos. Dentro de las áreas importantes de mayores y más rápidos avances tecnológicos se encuentra las Ciencias de la Computación donde la Inteligencia Artificial se ha convertido en una rama generadora de nuevos campos de estudio que abren ventanas que permiten vislumbrar enormes cantidades de conocimiento por descubrir para sustentar la continuidad del progreso de la sociedad como tal.

Existen varias ramas de la Inteligencia Artificial que se destacan por su aporte al rápido desarrollo de los sistemas inteligentes y entre esas áreas se encuentra la Computación Afectiva, la cual se inició con la noble intención de “humanizar” al máximo las metodologías que tienen en cuenta las emociones, sentimientos y personalidad para mejorar su funcionamiento proporcionando capacidades que tengan en cuenta los estados afectivos inherentes a la raza humana. Esto ha generado un campo emergente de Inteligencia Artificial conocido como Computación Afectiva (AC).

Por otro lado, se ha estado desarrollando otro campo independiente, particularmente orientado a beneficiar todo tipo de procesos educativos. Originalmente este campo se conocía como Sistemas Tutores Inteligentes (ITS), que tras adoptar nuevas tendencias y herramientas educativas pasaron a ser lo que hoy conocemos como Ambientes Inteligentes de Aprendizaje (ILE). La combinación de estas dos metodologías ha generado una simbiosis sinérgica que en muy poco tiempo ha demostrado importantes avances y beneficios, y ha permitido el surgimiento de sorprendentes campos de aplicación antes no contemplados ni intuidos. En este libro titulado “Multimodal Affective Computing, Technologies and Applications in Learning Environments”, se han recopilado diferentes metodologías de Inteligencia Artificial que permiten la implementación de estados afectivos en ambientes inteligentes de aprendizaje. Dentro del material proporcionado por los autores, el lector encontrará una descripción detallada y bien organizada de

las características más relevantes de las dos metodologías principales, desde sus fundamentos, su soporte teórico esencial hasta su fusión y algunas aplicaciones prácticas exitosas.

Los conceptos básicos de Computación Afectiva, Aprendizaje Automático y Reconocimiento de Patrones en Computación Afectiva y Ambientes de Aprendizaje Afectivo están escritos de manera integral y fácil de leer. En la segunda parte, se presenta una revisión general de un campo emergente llamado Análisis de Sentimientos para Ambientes de Aprendizaje, incluyendo un recorrido descriptivo sistemático a través de temas como la creación de recursos para la detección de sentimientos, métodos para la representación de datos, diseño y prueba de modelos de clasificación e integración de estos modelos a un sistema de aprendizaje. Las metodologías correspondientes al Reconocimiento Multimodal de Emociones Orientadas al Aprendizaje se presentan en la tercera parte donde se presentan temas como construcción de recursos para la detección de emociones, métodos para la representación de datos, sistemas de reconocimiento multimodal y reconocimiento multimodal de emociones en ambientes de aprendizaje con todos sus subtemas. La cuarta y última parte del libro está dedicada a un campo amplio de aplicación de la combinación de metodologías que es el Reconocimiento Automático de la Personalidad, abordando temas como la creación de recursos para el reconocimiento de la personalidad, métodos para la representación de datos, modelos de reconocimiento de la personalidad y reconocimiento multimodal de la personalidad para computación afectiva que completa el conjunto de temas que cubren el enfoque integral de la Computación Afectiva tal y como puede explotarse positivamente en Ambientes Inteligentes de Aprendizaje. Este libro puede ser muy útil no sólo para principiantes que deseen iniciarse en la práctica de estas útiles metodologías emergentes sino también para avanzados y expertos en la práctica y desarrollos del campo en su conjunto. Cualquier investigador, con o sin experiencia, deseoso de encontrar nuevos conocimientos capaces de mostrar horizontes atractivos y útiles tendientes a hacer que los ambientes inteligentes de aprendizaje consideren los estados afectivos humanos en todos sus procesos, quedará satisfecho tras consultar este libro. Encontrarán los materiales e ideas necesarios para dotar a cualquier ambiente de aprendizaje, no sólo de inteligencia que permita personalizar el proceso de aprendizaje sino también de la posibilidad de tomar decisiones y reaccionar ante estados emocionales y la personalidad del usuario.

El libro “Multimodal Affective Computing, Technologies and Applications in Learning Environments” fue publicado por la editorial Springer Nature en 2023 con ISBN 978-3-031-32541-0, ISBN 978-3-031-32542-7 (eBook) y <https://doi.org/10.1007/978-3-031-32542-7> . Para comentarios o sugerencias, se solicita atentamente contactar a los autores mediante los correos electrónicos siguientes: ramon.zc@culiacan.tecnm.mx, hector.cl@culiacan.tecnm.mx y hugojaier@inaoep.mx.

Reseña del Libro “Konrad Zuse’s Early Computers” de Raúl Rojas González

Por:

Rafael Gamboa Hiraes

Al utilizar las hojas electrónicas actuales rara vez recapacitamos en cómo se hacían cálculos similares en el pasado. Los hoy programadores de los más diversos lenguajes de programación consideramos los ingenios de cálculo como dispositivos que realizan labores cada vez más “rutinarias”. Aparentemente lo que actualmente todavía es “novedad” o trabajo que hay que hacer “a mano” (por ejemplo los programas básicos) eventualmente los elaborará una “computadora” o como actualmente se dice, una “inteligencia artificial”. Por ello, el libro que ahora nos trae Raúl despierta curiosidad al enterarnos que las computadoras desarrolladas en Norteamérica no fueron las únicas ni las primeras. En Europa, específicamente en Alemania, se desarrolló de 1936 a 1945 un conjunto de máquinas para el cálculo automatizado. La persona detrás de esto fue el ingeniero e inventor Konrad Zuse (1910-1995). El autor del libro llevó a cabo entrevistas con el inventor y plasmó de manera fácil de comprender la arquitectura y la implementación de las mismas en los dispositivos de cálculo conocidos como Z1 (la computadora mecánica) Z2, Z3 (desarrolladas durante la guerra) y la Z4, desarrollada en la posguerra y vendida al Instituto Tecnológico de Zurich. Un hecho importante es que esta computadora ya contaba con la posibilidad de operar con saltos condicionales. Una cuestión muy interesante e importante es la manera en que las operaciones aritméticas básicas, los condicionales para formar ciclos, y los condicionales en general, fueron tomando forma en estas máquinas y permitieron ejecutar cálculos que anteriormente eran hechos “a mano”, con regla de cálculo, ábacos y tablas de logaritmos.

Los capítulos del libro se pueden leer por separado pues están auto contenidos. Un elemento primordial a considerar en la descripción de las máquinas cubiertas en el libro es la evolución de los conceptos de cálculo y almacenamiento binario, el concepto de “bit” y la representación de números (enteros y de punto flotante). Los conceptos de “procesador”, “memoria”, “programa”, “subrutina”, “teclado” y “salida de resultados” se encontraban ya presentes en estas máquinas, acercándose mucho más a las ideas y conceptos de las computadoras modernas.

El libro muestra con diagramas la manera en que las distintas partes de las máquinas estudiadas llevaban a cabo la ejecución de las instrucciones, el acceso a la memoria, la entrada de datos y la entrega de los resultados. Es fascinante ver y entender cómo el ingenio humano (y en especial del Ingeniero Zuse) concibió y construyó estos dispositivos. Es por ello que la lectura de este libro resulta muy recomendable no solamente para los especialistas de este campo sino también para el público en general, sobre todo ahora que se nos viene encima el tsunami de la “inteligencia artificial”.



Figura 5. La computadora Z1

Reseña de "Seminario de Divulgación Aportaciones de la Inteligencia Artificial en la sociedad"

Por:

Dr. Allan Antonio Flores Fuentes aafloresf@uaemex.mx,

Dr. Juan Fernando García Mejía fgarciam@uaemex.mx,

Dr. Everardo Efrén Granda Gutiérrez eegrandag@uaemex.mx,

Universidad Autónoma del Estado de México, Centro Universitario Atlacomulco

El 30 de mayo del presente 2023, se llevo a cabo de manera presencial, el Seminario de Divulgación: “Aportaciones de la Inteligencia Artificial en la sociedad 2023A,” organizado por el Cuerpo Académico “Desarrollo de software, dispositivos y sistemas aplicados a la innovación tecnológica” y co-organizado por la Licenciatura en Ingeniería en Computación y Maestría en Ciencias de la Computación, de la Universidad Autónoma del Estado de México, con sede en el Teatro del Pueblo de la Cd. de Atlacomulco, Estado de México [1] . Por primera ocasión en este año 2023, se buscó realizar un foro donde estudiantes de posgrado y jóvenes investigadores, presenten a través de una ponencia la divulgación del conocimiento adquirido durante sus trabajos de investigación en el área de las Ciencias de la Computación. En la edición 2023 del Seminario de Divulgación se presentaron las ponencias:

1. “Tendencias y aplicaciones en el procesamiento del lenguaje natural”
2. “Visión por Computadora para la detección de padecimientos febriles en humanos”
3. “Algoritmo Clonal para el ajuste de un portafolio de inversión”



Figura 6. Participantes del Seminario de Divulgación: Aportaciones de la Inteligencia Artificial en la Sociedad 2023A

El Seminario de Divulgación fue presentado al público en general, mediante la difusión a través de un cartel digital publicado por redes sociales por parte del Centro Universitario UAEM Atlacomulco [2]. El evento no tuvo costo, siendo una retribución a la sociedad y llevando conocimiento e interés por las áreas de las Ciencias de la Computación y sus aplicaciones. Cabe resaltar que en el Seminario de Divulgación participaron dos Instituciones de Educación Superior de la zona norte del Estado de México, la Universidad Mexiquense del Bicentenario y el Tecnológico de Estudios Superiores de San Felipe del Progreso, contando con la asistencia de un total de 80 invitados externos, y 120 alumnos de la comunidad estudiantil del Centro Universitario UAEM Atlacomulco.

El Seminario de Divulgación “Aportaciones de la Inteligencia Artificial en la Sociedad 2023A”, en su primera versión tiene como finalidad convertirse en un foro referente para la presentación de aportaciones derivadas de trabajos de investigación realizados por alumnos de posgrado, jóvenes investigadores e investigadores reconocidos en el área de Ciencia de la Computación, que ofrezcan una retribución a la sociedad sobre temas actuales en los que pueden incursionar, y, además, ser un punto de encuentro para la comunidad que discute y planea sobre los retos futuros del área no solo a nivel regional en la zona norte del Estado de México, sino también a nivel nacional, promoviendo temas actuales.

Referencias

[1] Teatro del Pueblo, Ciudad de Atlacomulco, Estado de México. <https://acortar.link/HwQCEq>

[2] Cuerpo Académico, “Desarrollo de software, dispositivos y sistemas aplicados a la innovación tecnológica,” Centro Universitario Atlacomulco, Universidad Autónoma del Estado de México. <https://acortar.link/LOXgqR>

Invitamos a los colegas a seguir nuestra página de Facebook y a contribuir con contenido.



AMexComp

Taller sobre Descubrimiento Causal

Por:

Dr. L. Enrique Sucar, INAOE

Del 19 al 21 de junio del presente se llevó a cabo el primer taller sobre descubrimiento causal (Workshop on Causal Discovery – CaDis) en el Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), en Tonantzintla, Puebla. El objetivo del taller fue, por un lado, dar a conocer avances recientes en razonamiento y descubrimiento causal; y, por otro lado, integrar a la comunidad en México interesada en estos temas, además de promover el área con estudiantes e investigadores.

El taller incluyó un día completo de tutoriales, presentando aspectos teóricos y prácticos del razonamiento y descubrimiento causal. Además, se tuvieron dos pláticas invitadas con expertos internacionales, 8 ponencias y un póster. El programa detallado se puede consultar en la página del evento, donde se incluyen las presentaciones de las ponencias y el video de una de las conferencias magistrales:

<https://cadisworkshop.com.mx/>

Se están preparando unas memorias del taller a ser publicadas por la Academia Mexicana de Computación.



En el taller se tuvo una asistencia de cerca de 50 personas, con representación de 10 instituciones académicas y de investigación, y dos de la industria, de México; además de participantes internacionales, uno de España y dos de Estados Unidos de América. Dado el interés de la comunidad y los buenos comentarios de los asistentes, se planea realizar un segundo taller en 2024, posiblemente en conjunción con alguna conferencia como Iberamia o MICAI.

El desarrollo de los modelos causales está teniendo un importante crecimiento en los últimos años, con avances significativos en el razonamiento causal (intervenciones y contrafactuales), así como en el aprendizaje de los modelos a partir de datos. Estos modelos

permiten realizar otro tipo de inferencias que no son posibles con otras técnicas de inteligencia artificial. Por ejemplo, predecir el efecto de intervenciones (¿Cuál sería el impacto de cierta campaña de publicidad?); e imaginar lo que hubiera posado si se hubieran realizado acciones diferentes a las que ocurrieron, conocidos como contrafactuales (¿Se hubiera perdido el cliente si le hubiéramos ofrecido ciertos incentivos?).

Otra ventaja es que pueden ayudar a hacer explicaciones, algo importante para muchas aplicaciones de los sistemas inteligentes. El aprender estos modelos de datos observacionales (sin tener que hacer experimentos) es todo un reto, ya que en general no es posible obtener un modelo único, por lo que es un tema atractivo de investigación. Para aquellos interesados en adentrarse en estos temas, les recomiendo el libro de J. Pearl y D. Mackenzie, *The Book of Why*, que presenta una introducción accesible a estos temas.

Los 8 principios de la ACM para el desarrollo, implementación y uso de Inteligencia Artificial (IA) Generativa

Por:

Dra. María del Pilar Gómez Gil

INAOE

Después de un arduo trabajo en el que participaron decenas de especialistas de Europa y Estados Unidos, el pasado 11 de Julio del presente año, la “Association for Computing Machinery” (ACM) anunció la publicación de sus “Principios para el desarrollo, implementación y uso de tecnologías de IA Generativa” [1]. Esta publicación, disponible en [2], es una más de varias reacciones que se han dado este año, ante una situación de alta expectativa pero a la vez alta preocupación en la industria productora y usuarios consumidores de tecnologías basadas en IA. En México ya hemos sido testigos de varias pláticas, algunas de ellas dictadas por miembros de AMEXCOMP, en donde hemos tratado de explicar nuestro sentir ante los retos y oportunidades que conllevan el uso de Inteligencia Artificial Generativa. Asimismo, los legisladores mexicanos ya han empezado a buscar guías indispensables para poder establecer un orden básico al respecto de este punto.

El documento de la ACM establece 8 principios, de los cuales 4 están enfocados específicamente a sistemas generativos mientras que los otros 4 son una adaptación tomada de otro documento, producido el año pasado por la misma ACM, y titulado “Declaración de principios para sistemas algorítmicos responsables” [3].

El documento inicia dando una definición general del término “Inteligencia Artificial Generativa” como “las técnicas y herramientas que pueden usarse para crear nuevo contenido, incluyendo texto, audio, imágenes, video, código para computadoras y otros artefactos técnicos”. En todo el documento llama la atención la cantidad y el detalle de información colocada en pies de página, en donde se hace notar a la persona lectora varios de los riesgos y preocupaciones que aún no están completamente estipulados ni regulados, y que merecen un estudio mucho más profundo, tanto por parte de los expertos tecnólogos, como de todos los actores involucrados en sistemas basados en IA Generativa.

Enseguida se listan los 8 principios y algunos comentarios de su descripción que personalmente encontré importantes:

Principios específicos a IA Generativa:

1. **Límites y guías en el uso e implementación de los sistemas.** Aquí se resalta la necesidad de que las regulaciones se revisen, a fin de limitar el uso de sistemas generativos, de forma tal que sea claro el riesgo que puede presentarse al usarlos y la necesidad de que las personas formen parte directa del ciclo de vida del sistema. Los proveedores deben evaluar el impacto de sus sistemas antes de liberarlos, y se recomienda el uso de definiciones jerárquicas de niveles de riesgos que se pueden presentar al usar los sistemas.
2. **Propiedad.** Este principio está relacionado con una de las principales preocupaciones actuales en el uso de IA Generativa. Es necesario asegurar que los derechos de propiedad intelectual no son rebasados por los sistemas

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.

generativos y que se protege el derecho de los creadores humanos de los datos que alimentan el entrenamiento de estos sistemas.

3. **Control de datos personales.** Los usuarios de los sistemas deben tener la posibilidad de determinar si permiten o no que sus datos personales se utilicen para entrenar o facilitar el uso de los sistemas generativos.
4. **Capacidad de corrección.** Los sistemas deben presentar los medios necesarios para registrar los errores que producen y ofrecer mecanismos de corrección de estos errores.

Principios anteriores adaptados a sistemas generativos:

1. **Transparencia.** Los sistemas basados en IA Generativa deben especificar muy claramente lo que producen a los diferentes actores involucrados, enfatizando que se interactúa con un sistema artificial y no con una persona humana. Además, los sistemas deben informar que pueden cometerse errores, y deben proveer mecanismos que identifiquen que los productos son generados de forma artificial. Esto último puede requerir el uso de marcadores criptográficos en los objetos generados.
2. **Procesos de auditoría.** Los proveedores deben asegurarse de que el funcionamiento de sus sistemas es registrado siempre que no viole la privacidad, a fin de que puedan ser auditados y corregidos de ser necesario.
3. **Limitaciones al impacto ambiental.** Se recomienda desarrollar estrategias para medir y limitar el impacto ambiental en el entrenamiento y uso de los sistemas generativos.
4. **Mayor seguridad y privacidad.** Los sistemas generativos están propensos a nuevos tipos de riesgos en seguridad y privacidad, por lo que es necesario implementar mecanismos adicionales que permitan auditar y controlar estos nuevos retos.

Es fundamental que todos conozcamos estos principios y establezcamos un diálogo y actividades constantes en nuestras organizaciones para hacer del uso de la IA generativa una herramienta que ayude al bienestar general, así como estar preparados para controlar los riesgos que implica un uso irresponsable de estas herramientas.

Referencias:

- [1] Association of Computing Machinery. “World’s Largest Association of Computing Professionals Issues Principles for Generative AI Technologies” New Release, July 11, 2023. Disponible en: <https://www.acm.org/media-center/2023/july/tpc-statement-generative-ai>
- [2] Association of Computing Machinery. “Principles for the development, deployment, and use of generative AI technologies”. June 27, 2023. Disponible en: <https://www.acm.org/binaries/content/assets/public-policy/ustpc-approved-generative-ai-principles>
- [3] Association of Computing Machinery. “2022 Statement on principles for responsible algorithmic systems”, October 26, 2022. Disponible en: <https://www.acm.org/binaries/content/assets/public-policy/final-joint-ai-statement-update.pdf>



Detalle de la página de la ACM relacionada al consejo de políticas tecnológicas globales

<https://www.acm.org/public-policy/ustpc>

Recordando a ...

Timothy John Berners-Lee nació el 8 de junio de 1955 en Londres, Inglaterra. Sus padres (Mary Lee Woods y Conway Berners-Lee) eran matemáticos y habían trabajado en la *Ferranti Mark 1* desarrollada en la Universidad de Manchester, que fue la primera computadora digital con programa almacenado y de uso general que estuvo disponible a la venta.

Timothy fue el mayor de cuatro hermanos y fue el único de ellos que heredó el interés por la electrónica, aunque decidió estudiar física en la Universidad de Oxford, de donde se graduó en 1976. Mientras cursaba sus estudios universitarios, construyó una computadora rudimentaria con piezas usadas de un televisor que compró en una tienda de reparaciones.

Después de graduarse de la universidad, trabajó como ingeniero en la compañía de telecomunicaciones *Plessey* en Poole, Dorset. En 1978, ingresó a *D.G. Nash*, en Ferndown, Dorset, donde ayudó a crear software para impresoras. De junio a diciembre de 1980, se incorporó como un contratista independiente al denominado *Conseil Européen pour la Recherche Nucléaire* (CERN). Mientras se encontraba en Ginebra, Suiza, propuso un proyecto basado en el concepto de hipertexto, a fin de facilitar la compartición y la actualización de información entre los investigadores. Para demostrar sus ventajas, desarrolló un prototipo llamado ENQUIRE. Aquí cabe mencionar que fue Ted Nelson y no Berners-Lee el que acuñó el término “hipertexto” en los 1960s. Nelson imaginó una plataforma en línea que reemplazaría a las editoriales convencionales y pensó que los autores podrían crear ligas entre documentos de tal forma que los lectores pudieran seguirlas de un documento a otro. Hacia finales de los 1980s, el hipertexto era un área activa de investigación, pero en la práctica solo se utilizaba en sistemas cerrados (es decir, sistemas que no estaban en línea) tales como el sistema de ayuda de Windows y la plataforma de documentos electrónicos de la Macintosh llamada *Hypercard*.

Una vez terminado su contrato con el CERN, Berners-Lee comenzó a trabajar en la empresa *John Poole's Image Computer Systems Ltd*, ubicada en Bournemouth, Dorset. Ahí trabajó en un proyecto sobre “llamadas a procedimientos remotos en tiempo real”, en el cual adquirió experiencia en redes de cómputo. En 1984, regresó al CERN, esta vez con un puesto formal. En esta ocasión, trabajó inicialmente en el grupo de adquisición de datos y control, con la tarea de capturar y procesar datos experimentales. Aunque el CERN se dedica a la física de partículas, tenía enormes necesidades de cómputo y tenía a un gran número de empleados. Su enorme red interna estaba conectada a Internet (de hecho, el CERN era el nodo más grande de Internet en Europa). En marzo de 1989, Berners-Lee comenzó a circular un documento titulado “Manejo de Información: Una Propuesta” en el cual proponía un sistema de publicaciones mediante hipertextos basado en el uso de Internet. Berners-Lee argumentaba que este sistema le ayudaría al CERN a manejar la gigantesca cantidad de documentos, código y reportes producidos por sus miles de empleados, muchos de los cuales eran visitantes temporales. Berners-Lee utilizaba una computadora NeXT en el CERN y ésta se convertiría en el primer servidor web del mundo. Mike Sendall, que era el jefe de Berners-Lee pensó que la propuesta era “vaga, pero emocionante”. Curiosamente, Robert Cailliau había propuesto de forma independiente otro proyecto para desarrollar un sistema de hipertexto en el CERN, pero decidió unirse a la propuesta de Berners-Lee. Su propuesta inicial se basaba en el sistema ENQUIRE y dio origen a la *World Wide Web*, para lo cual Berners-Lee diseñó y construyó el primer navegador web. Su software también funcionaba como un editor (llamado *WorldWideWeb*), el cual se ejecutaba en el sistema operativo *NeXTStep* y también diseñó el primer servidor Web, llamado CERN HTTPd (que es una abreviación de *Hypertext Protocol daemon*).



En mayo de 1990, Mike Sendall autorizó a Berners-Lee a dedicar algo de tiempo a su idea, justificando esto como una prueba de la estación de trabajo NeXT. Esta computadora tenía un sistema operativo basado en Unix que optimizaba la implementación de aplicaciones gráficas. Berners-Lee dedicó sus primeros meses a trabajar en las especificaciones en un intento por atraer el interés de compañías de software existentes que usaban hipertexto. Para octubre de 1990, había comenzado a implementar el prototipo de su navegador Web y del software del servidor. El 20 de diciembre de 1990 publicó el primer sitio web, en el cual proporcionaba una explicación de lo que era la World Wide Web y la forma en la que la gente podía usar un navegador, así como la manera de montar un servidor web. Para el 6 de agosto de 1991, después de más desarrollos y de realizar algunas pruebas en el CERN, usó el Internet para anunciar su nueva “World Wide Web” y para distribuir el nuevo software que había desarrollado. La World Wide Web, al igual que la mayoría del Internet en esa época, estaba íntimamente ligada al sistema operativo Unix y a su filosofía de construir nuevos sistemas mediante la recombinación de herramientas existentes. También seguía la filosofía de lograr compatibilidad a través de protocolos de comunicación en vez de basarse en código, hardware o un sistema operativo estándar. Sus especificaciones para el nuevo sistema condujeron a tres nuevos estándares de Internet:

1. **HTML** (*Hyper Text Markup Language*) se usó para especificar la forma en la que el texto de una página web sería etiquetado (por ejemplo, como una hiperliga, como un párrafo ordinario o como un encabezado de nivel 2). Esto fue una aplicación del estándar SGML (*Standard Generalized Markup Language*).
2. **HTTP** (*Hyper Text Transfer Protocol*) especificaba las interacciones a través de las cuales los navegadores Web podían solicitar y recibir páginas en HTML desde servidores Web. Esto convirtió al HTTP en un protocolo para transferir archivos, el cual era fácil de diseñar e implementar debido a los estándares y el software existentes en Internet. De entre ellos, destaca el TCP/IP (por el cual Vinton Cerf y Robert E. Kahn obtendrían el *ACM Turing Award* en 2004), el cual proporcionaba la infraestructura necesaria para transferir datos a través de la red de un programa a otro.
3. **URL** (*Uniform Resource Locator*) se basó en el hecho de que los sitios de Internet habían estado usando el Sistema de Nombres de Dominios desde mediados de los 1980s. La novedad, sin embargo, era que el uso de `http://` decía a los navegadores Web y a los usuarios que esperaran un servidor Web. La información después de la primera diagonal “/” identificaba la página que se solicitaba en la computadora huésped. Berners-Lee también especificó formatos URL para recursos existentes de Internet, incluyendo servidores de archivos, servidores de gopher (que era un sistema de navegación basado en hipertexto) y para huéspedes de telnet que permitían realizar conexiones a terminales.

El URL era la invención más simple de las tres, pero fue crucial para diseminar el uso de la Web, porque resolvía el problema del “huevo y la gallina” que enfrenta cualquier tipo de sistema de comunicaciones. ¿Para qué querría alguien montar un sitio Web si casi nadie tenía navegadores Web? ¿Por qué ejecutar un navegador Web cuando casi nadie tenía un servidor Web que pudiese visitarse? El sistema basado en el uso de URL convirtió a los navegadores Web en una opción conveniente para acceder a los recursos existentes, catalogados en páginas Web. Para 1992, el *Whole Internet Catalog* decía: “la World Wide Web no ha sido explotada debidamente todavía. . . El hipertexto se usa primordialmente como una forma de organizar recursos que ya existen.”

Pese a que, en un principio, el CERN no dedicó muchos recursos a apoyar la Web, posteriormente incorporó a algunas personas a su desarrollo (principalmente jóvenes que realizaban estancias cortas). Pero resultó más importante la actitud del CERN de permitir el libre uso de los nuevos estándares y del código desarrollado para desarrollar nuevo software que resultara mejor que el existente. Robert Cailliau, del grupo de Sistemas de Cómputo para Oficinas, jugó un papel fundamental en el desarrollo de la Web. En 1991, el CERN produjo un navegador simple, basado en texto, que podía ser accedido fácilmente a través de Internet, así como un navegador para Macintosh, que resultó esencial para diseminar la Web, puesto que las computadoras NeXT fueron siempre muy escasas. Con los años, otras personas implementaron navegadores más rápidos y robustos con nuevas funcionalidades tales como gráficos en las páginas Web, historia de la navegación y botones para moverse hacia adelante y hacia atrás. Mosaic, que fue liberado en 1993 por el *National Center for Supercomputer Applications* de la Universidad de Illinois, puso la Web al alcance de

millones de usuarios.

Berners-Lee atribuiría años más tarde su éxito a la suerte (“estuve en el lugar correcto en el momento correcto”). Pero realmente fue alguien que tuvo éxito en una tarea en la que fallaron equipos más grandes y con muchos más fondos, pues en una labor casi individual, logró establecer las bases para un sistema de hipertexto global que rápidamente se transformó en una infraestructura universal para comunicaciones en línea y que se volvería la base para muchas nuevas industrias. Irónicamente, en 1991, la conferencia *Hypertext* de la ACM rechazó un artículo de Berners-Lee en el que éste describía la *World Wide Web*. Desde una perspectiva científica, la Web parecía evadir muchos de los problemas de investigación relacionados a las capacidades que Ted Nelson consideró esenciales para un sistema de publicaciones basado en hipertexto. Por ejemplo, si una página Web se removía, las ligas que apuntaban a ella dejaban de funcionar. Si la página a la que queríamos conectarnos se modificaba, la información que originalmente prometía su liga podría ya no existir. Las ligas eran unidireccionales (uno no podría ver qué otras páginas estaban ligadas a un documento). No había un índice central, navegable, de sitios Web y de su contenido. Tampoco permitía que quien publicara algo en la Web pudiese recibir retribuciones económicas por su trabajo. Pero tal vez debido a su minimalismo tecnológico, la Web resultó muy atractiva para sus primeros usuarios, lo cual le permitió escalar con facilidad (instalar un servidor Web en una computadora era algo muy fácil) y la carencia de un sistema de indizado evitó sobrecargas. Después de hacer despegar el uso de la Web, surgieron industrias completamente nuevas para proporcionar las capacidades requeridas por la enorme escala que ésta llegó a alcanzar. Un buen ejemplo es *Google*, que se volvió el motor de búsqueda predominante en Internet.

Con el paso de los años, Berners-Lee solo se arrepintió de una de sus decisiones originales de diseño: el uso de “//” en las URLs, lo cual consideró que hacía más largas y difíciles de escribir las direcciones Web sin añadir ninguna funcionalidad útil.

El sucesor de *Mosaic* fue el navegador comercial *Netscape*, que sería usado por cientos de millones de personas y que impulsó los dominios .com en las bolsas de valores de todo el mundo. Para el año 2000, se estimaba la existencia de unos 17 millones de sitios web usados para transacciones comerciales tales como compras y banca en línea, así como para mostrar documentos. En el proceso, HTML empezó a tener extensiones incompatibles y barrocas para hacer cosas como permitir tipos diferentes de letra y formatos de páginas, en vez de enfocarse en el objetivo original de Berners-Lee que era la estructura de los documentos.

En 1994, Berners-Lee dejó su empleo en el CERN para aceptar una oferta como profesor en el Instituto Tecnológico de Massachusetts. Ahí estableció el *World Wide Web Consortium* (W3C) para estandarizar HTML y otros elementos más nuevos de la Web.

Desde finales de los 1990s, Berners-Lee se enfocó principalmente en intentar hacer que las compañías tecnológicas y los que publican en la Web agreguen cierta funcionalidad que él denominó “Web Semántica”. Definió su idea de la siguiente forma: “La Web Semántica es una extensión de la Web actual en la que se le da un nuevo significado, bien definido, a la información, lo cual permite que las computadoras y las personas puedan cooperar entre sí”. Berners-Lee ha usado su enorme peso público para influenciar la forma en la que los gobiernos y las empresas moldean la Web. En 2009, estableció la *World Wide Web Foundation*, que busca la “igualdad digital”. Ha defendido la protección de datos personales y ha criticado el creciente dominio de las plataformas comerciales de redes sociales. A lo largo de los años, Berners-Lee ha recibido numeros premios, incluyendo el ACM Software System Award (1992), la *Royal Society of Arts Albert Medal* (2002), el *Queen Elizabeth II Prize for Engineering* (2013) y el ACM Turing Award (2016). Es IEEE Fellow (desde 2008) y fue nombrado caballero del imperio Británico (2007). Ha recibido también más de 19 doctorados honoris causa.