el faro
la luz de la ciencia

UNAM 60

Años del cómputo en México

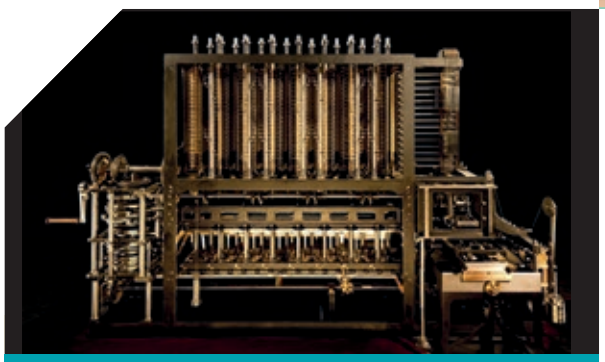
UNAM La Universidad de la Nación

IIMAS UNAM

CViCom

Boletín informativo de la Coordinación de la Investigación Científica • Ciudad Universitaria, agosto de 2018, Número especial

SUMARIO



3 Editorial

4 Asómate a la ciencia
Justina y el mundo de la Robótica
Jesús Savage Carmona

9 Espacio abierto
Pasado y presente del cómputo científico
Suemi Rodríguez Romo y Ricardo Paramont Hernández García

12 Escienci@
Códigos de cadenas en la UNAM
Ernesto Bribiesca Correa

16 Historia de la ciencia
De las tarjetas perforadas a _Miztli_
Patricia de la Peña Sobarzo y Yassir Zárate Méndez

20 Ventana universitaria
Computación concurrente a través del razonamiento secuencial
Sergio Rajsbaum y Michel Raynal

23 A ver si puedes
Alejandro Illanes Mejía

Sesenta años del cómputo en México

Hace seis décadas llegó el primer equipo de cómputo a México. Su destino fue el Centro de Cálculo Electrónico de la Facultad de Ciencias de la UNAM. Con esa adquisición, el país y la Universidad Nacional se incorporaron a un selecto grupo de naciones y de instituciones de enseñanza e investigación que contaban con ese tipo de herramientas. Ya nada sería igual desde entonces.

La IBM 650 fue la primera computadora instalada en América Latina. Ese inmenso paso dado por la UNAM abrió un sendero que sigue recorriéndose a la fecha.

En este 2018, la Universidad, a través del Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS) y el Centro Virtual de Computación (CViCom), celebra 60 años de cómputo con actividades académicas, conferencias, congresos, exposiciones y con este número especial de ***El faro, la luz de la ciencia***.

Así, en esta edición, integrantes de diferentes institutos y centros de investigación y enseñanza de la UNAM y de otras entidades académicas, reflexionan y describen asuntos relacionados con la robótica, el cómputo científico, los códigos de cadena y la computación concurrente.

Sobra mencionar la trascendencia del cómputo y del supercómputo en el desarrollo de la ciencia y la tecnología. Ningún proyecto deja de pasar por el tamiz del cómputo electrónico, que también se toma a sí mismo como objeto de estudio, como explican Suemi Rodríguez Romo y Ricardo Paramont Hernández García.

En tanto, Jesús Savage Carmona repasa los avances logrados por la UNAM en el desarrollo de robots. Para ello, presenta el trabajo desplegado en torno a *Justina*, un dispositivo de servicio que fija el rumbo para el futuro inmediato.

De su lado, Ernesto Bribiesca Correa repasa los alcances de los códigos de cadena, con los que se representa la trayectoria del vuelo de un avión o la ruta de una abeja al encontrar una fuente de polen.

Sergio Rajsbaum y Michel Raynal, este último de la Universidad de Rennes, Francia, explican puntualmente los retos planteados a la concurrencia de los procesos computacionales.

Por último, nuestra habitual sección de Historia de la ciencia ofrece un recuento de los momentos estelares del cómputo y el supercómputo en la UNAM.

Como se mencionó, la Universidad Nacional es pionera en el ámbito de la computación en América Latina, y aún hoy en día se mantiene a la vanguardia con la especialización en Cómputo de Alto Desempeño y el diseño de una nueva licenciatura en Ciencia de Datos, que incorporará la materia de cómputo cuántico.

Por otra parte, el CViCom, creado bajo la gestión del rector Enrique Graue Wiechers en 2017, busca coordinar esfuerzos alrededor del cómputo acorde con las diversas entidades que lo estudian, investigan o desarrollan como una disciplina académica. Además, el IIMAS es la única entidad del Subsistema de la Investigación Científica que tiene como objetivo el cultivo de esta área.

Sea este número especial de ***El faro*** una ventana a las numerosas actividades desplegadas por los investigadores universitarios, así como por sus colegas de otras latitudes, en materia de cómputo.

El faro

En portada

Composición con un fragmento del mural "La historia de la computación" de Facundo Pérez O., Vicente Díaz P., Ángel Reyes M. y Daniel Orales O., y el logo conmemorativo del 60 aniversario del cómputo en México, diseñado por Eprin Varas y el CViCom. El mural se encuentra en el edificio de la DGTIC-UNAM.

DIRECTORIO

UNAM

Dr. Enrique Graue Wiechers
Rector

Dr. Leonardo Lomelí Vanegas
Secretario General

Ing. Leopoldo Silva Gutiérrez
Secretario Administrativo

Dr. William H. Lee Alardín
Coordinador de la Investigación Científica

El faro, la luz de la ciencia

Patricia de la Peña Sobarzo
Directora

Yassir Zárate Méndez
Supervisor editorial

Sandra Vázquez Quiroz
Víctor Manuel Hernández Correa
José Antonio Alonso García
Edgar Vergara Hernández
Colaboradores

Víctor Manuel Hernández Correa
Paola Andrea Moreno Franco
Diseño gráfico y formación

CONTACTO

http://elfaro.cic.unam.mx

boletin@cic.unam.mx

Síguenos en:

Boletin El faro UNAM

@ElfaroUNAM

El faro, la luz de la ciencia, es una publicación de la Coordinación de la Investigación Científica. Oficina: Coordinación de la Investigación Científica, Circuito de la Investigación, Ciudad Universitaria, CP 04510 Ciudad de México. Teléfonos 5550 8834 y 5666 5201. Certificado de reserva de derechos al uso exclusivo del título, en trámite. Impresión: Infagon Web, S.A. de C.V. Domicilio: De la Alcaicería 8, Colonia Zona Norte Central de Abastos, C.P. 09040, Ciudad de México. Tiraje especial: 10,400 ejemplares. Distribución: Coordinación de la Investigación Científica, Ciudad Universitaria.

Prohibida la reproducción parcial o total del contenido, por cualquier medio impreso o electrónico sin la previa autorización.

Justina y el mundo de la Robótica

Dr. Jesús Savage Carmona
Facultad de Ingeniería, UNAM

Los robots industriales, usados principalmente en la manufactura y que en su mayoría son brazos robotizados estacionados en un solo lugar, se introdujeron en las fábricas durante la década de los sesenta del siglo pasado. La figura 1 muestra estos robots en una fábrica de automóviles.[1]

Figura 1: Ensamblado de automóviles usando brazos de robot.

Robots de servicio

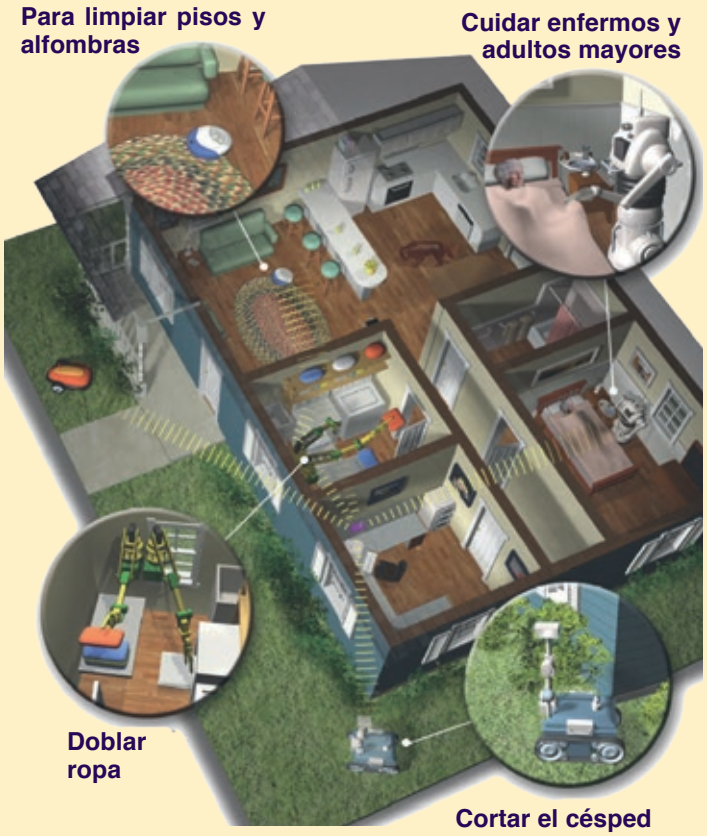
Mucho ha avanzado esta tecnología desde entonces, tanto que se espera en un futuro cercano una demanda masiva de los ahora llamados robots de servicios, cuyo objetivo es simplificar el trabajo humano en casas, oficinas, tiendas, etcétera. Estos son dispositivos ambulantes programables que ofrecen servicios en forma automática o semiautomática. Los servicios no son tareas que ayuden en la manufactura industrial de bienes, sino trabajos para los seres humanos en general y sus equipos terminados. A diferencia de los industriales, estos robots navegan en los espacios que les han sido asignados.

Así como se incorporaron a la vida cotidiana televisores, radios, computadoras y los teléfonos celulares, los robots de servicio también lo harán en algún momento, y llegarán a ser muy familiares. La figura 2 muestra el tipo que se espera tener.[2]

En una casa, en un futuro cercano, no solamente se tendrá un robot, sino varios. Los habrá externamente en las casas; uno de ellos se dedicará a cortar el pasto, mientras que el otro hará rondines de vigilancia. Dentro de la casa funcionarán algunos pequeños que limpien, aspiren y trapeen el piso; uno fijo que se encargue del lavado y planchado de la ropa; otro, tipo humanoide, que ayudará a las personas a traer y llevar objetos de un lugar a otro; este último tendrá capacidades de comunicarse con los integrantes de la casa usando lenguaje natural y planeará las acciones y los movimientos necesarios para hacer sus funciones.

Los robots de servicio son sistemas de software y hardware, autónomos o semiautónomos, que se encuentran en ambientes dinámicos y complejos. Su autonomía consiste en la habilidad de tomar decisiones basadas en una representación interna del mundo. Han surgido de áreas de investigación, entre otras, como son la inteligencia artificial, el procesamiento de señales y la teoría de control.

Los robots de servicio harán lo siguiente para atender a adultos mayores: servir de compañía, explicar accio-



Diseño: Don Foley, Scientific American sp 18, 2008.

Figura 2: Ejemplos de robots de servicio.

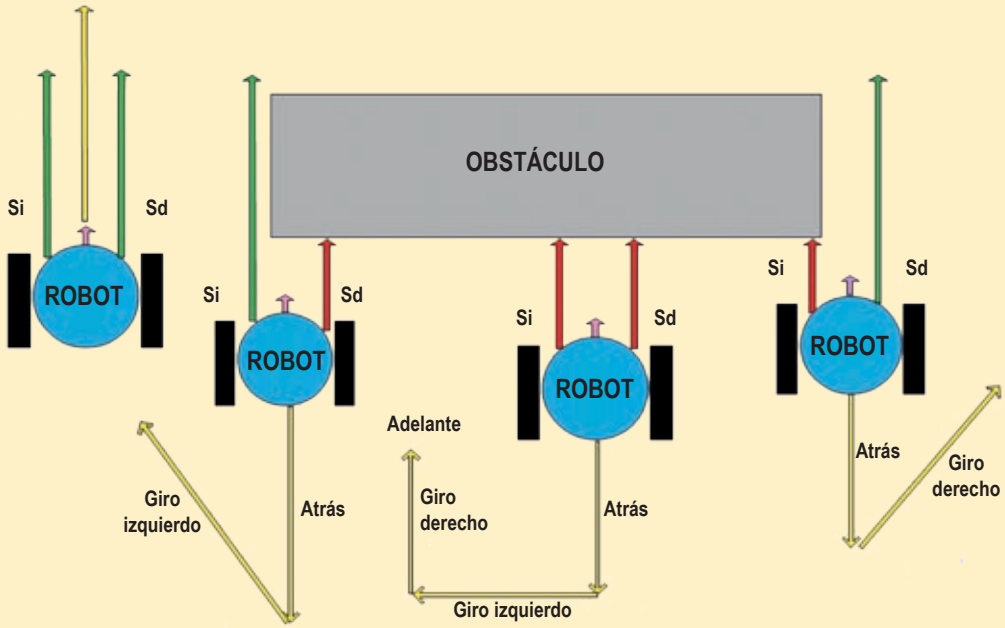


Figura 3. Movimientos de un robot omnidireccional para evadir un obstáculo.

nes rutinarias, llevar lista y toma de medicinas, apoyar en el ejercicio físico, suministrar información de calendario y hora, proveer información espacial, mantener la casa segura para evitar accidentes, por ejemplo, apagar la estufa, detectar caídas o situaciones de riesgo, recoger objetos del suelo. Para el uso de servicios hospitalarios: transporte de medicinas, análisis y alimentos, visitas presenciales de médicos externos usando los robots de servicio como medio de navegación, monitoreo de los signos vitales de los pacientes y para apoyar en el ejercicio físico de los pacientes.

Las capacidades básicas que los robots de servicio deben tener para poder realizar estas tareas son: ser reactivos, es decir, reaccionar oportuna y apropiadamente a eventos imprevistos; ser capaces de resolver tareas por medio de planes; resolver sus tareas eficazmente, haciendo uso de procedimientos probados en situaciones rutinarias; tratar con interacciones, positivas y negativas, con otros robots y humanos.

Los robots necesitan adaptarse a cambios en las condiciones del medio ambiente, ser capaces de aprender a realizar nuevas tareas o variantes de las que ya tienen programadas, a través de demostraciones (de los usuarios), de buscar información (en Internet) y de interactuar con el ambiente.

En el desarrollo de los robots de servicio se han observado tres paradigmas de arquitecturas de robots móviles muy bien marcadas.

La primera, llamada “tradicional”, se aplicó a partir de los años sesenta del siglo pasado, cuando se utilizaban modelos simbólicos para representar el medio ambiente y el uso de técnicas básicas de inteligencia artificial para la búsqueda de caminos y una representación espacio/estado para la planeación de acciones. Un ejemplo de este tipo de arquitecturas es la del robot *Shacky* [3], desarrollado en la Universidad de Stanford.

La segunda tendencia fue implementada en los años ochenta en el MIT, por el investigador Rodney Brooks, llamada “reactiva”, en la cual no se cuenta con una representación del medio ambiente, ni tampoco con módulos de planeación de acciones ni movimiento. Lo que se tiene son varios módulos que responden a estímulos específicos generando salidas inmediatas, las cuales son combinadas o seleccionadas por un árbitro para generar lo que el robot debe hacer bajo ciertas circunstancias. Los módulos pueden utilizar máquinas de estados, redes neuronales o campos potenciales. Un ejemplo de este tipo de arquitecturas es la del robot *Roomba* [4], diseñado por la empresa Irobot.

El paradigma más reciente es el llamado “probabilístico”, basado en el concepto que tanto el sensado del medio ambiente que hace el robot, así como el de sus movimientos, son dependientes de variables aleatorias, que pueden ser manipuladas utilizando conceptos probabilísticos. Un ejemplo de estas arquitecturas es el automóvil autónomo que obtuvo el primer lugar en el DARPA Grand Challenge de 2005, desarrollado por Sebastian Thrun, de la Universidad de Stanford [5].

Cada uno de estos tipos de arquitecturas tiene sus ventajas y desventajas. Así, tomando lo mejor de cada una de ellas, se pueden tener sistemas híbridos, como el que se ha desarrollado con el robot *Justina*, de la Facultad de Ingeniería de la UNAM, el cual se presenta más adelante.

1.1. Máquinas de estado

La forma más directa para programar la navegación de un robot móvil es a través del uso del concepto de máquinas de estado. En estas se tiene una representación del estado en el cual se encuentra el sistema y dadas ciertas condiciones de entradas y el estado presente se calcula el siguiente. La parte fundamental de estas máquinas es el algoritmo que ejecutan. En la figura 3 se muestra un robot omnidireccional que cuenta con dos motores que le permiten ir hacia adelante, atrás y hacer giros hacia la derecha e izquierda, asumiendo que solamente puede hacer giros de 45 grados.

Cuenta con dos sensores que le permiten detectar obstáculos, como se muestra en la figura 3. Si el robot no detecta ningún obstáculo seguirá avanzado; si detecta

un obstáculo con su sensor izquierdo, primero va hacia atrás y después gira hacia la derecha 45 grados; si detecta un obstáculo con su sensor derecho, primero va hacia atrás y después gira hacia la izquierda 45 grados; si detecta un obstáculo con los dos sensores, entonces el robot va hacia atrás, gira dos veces a la izquierda, avanza hacia adelante y finalmente girará hacia la derecha dos veces. Esta explicación se muestra en el algoritmo de la figura 4.

Las máquinas de estado pueden programarse desde muy bajo nivel, usando *flip-flops* en su instrucción y compuertas ANDs y ORs, hasta en uno muy alto usando lenguajes de programación como C++ y Python. También se pueden utilizar para su instrumentación redes neuronales retroalimentadas en FPGAs[6].

1.2. Robot *Justina*

Justina es un robot de servicio completamente diseñado en la Facultad de Ingeniería de la UNAM.

El sistema de operación de *Justina*[7] está basado en una arquitectura híbrida de robots móviles, que consiste de cuatro capas: entrada, planeación, conocimiento y ejecución.

Cada una de ellas cuenta a su vez con varios subsistemas que controlan en forma general el funcionamiento del robot. Operativamente cada uno de los componentes de software se enlaza entre ellos usando el mecanismo de comunicación entre procesos del sistema operativo para robots ROS.

Robot *Justina*.

1.3. Capa de entrada

En esta etapa se procesan las señales provenientes de los sensores internos y externos con los que cuenta el robot, utilizando técnicas de procesamiento digital de señales, para generar una representación simbólica de ellas. Por ejemplo, si el robot está situado enfrente de una mesa donde hay objetos, esta capa indicaría los tipos y nombres de objetos encontrados. Si alguien le da un comando de voz al robot, después de hacer reconocimiento de palabras, se encuentra una representación de la oración, utilizando técnicas de lenguaje natural, que puede aplicarse después por las siguientes capas para realizar lo que se solicita.

Esta capa contiene los siguientes subsistemas:

Sensores internos. Sirven para determinar el estado interno del robot y están formados por los siguientes elementos: codificadores en las ruedas, sensor de carga de la batería, sensor de inclinación, sensor de temperatura, etcétera.

Sensores externos. Los sensores externos sirven para detectar el medio ambiente, utilizándose los siguientes: de contacto, de luz infrarroja, cámaras, micrófonos, láser, sonares, etcétera.

Interpretación/representación simbólica. Con los datos proporcionados por los sensores externos e internos se obtiene una representación simbólica, que se logra utilizando técnicas de procesamiento digital de señales.

Tareas del robot. Un conjunto de tareas que el robot necesita realizar en el tiempo que fueron programadas.

Interfaz humano/robot. La comunicación entre el usuario y el robot es por medio de la voz y gestos manuales; el robot responde usando voz sintética y expresiones faciales simples.

Percepción. El módulo de percepción recibe la representación simbólica de los datos que vienen de los subsistemas: tareas del robot, interfaz hombre-robot y los sensores internos y externos. Con esta representación simbólica se genera una o una serie de creencias.

1.4. Capa de planeación

En esta capa se planean las acciones que tiene que hacer el robot para cumplir las situaciones que se le presentan.

Validación de situaciones. Este subsistema valida las creencias generadas en la capa de entradas, así se reconocen situaciones nuevas en el medio ambiente.

Activación de metas. Dado el reconocimiento de situaciones nuevas un grupo de metas se activa.

Banco de procedimientos. Se tienen procedimientos que resuelven parcialmente problemas específicos; estos están representados usando máquinas de estados. Se tienen procedimientos para tomar objetos, limpiar mesas, encontrar humanos, alinearse a una pared o muebles, etcétera.

Planeador. El planeador intenta alcanzar cada una de las metas seleccionando un conjunto de funciones del banco de procedimientos. Escoge el mejor plan de acciones que resuelve cada una de las metas y los reúne. Se cuenta con dos tipos de planeadores: acciones y movimientos. El planeador de acciones encuentra una secuencia de acciones que lleven a la solución de un problema, se utiliza una máquina de inferencias que realiza búsquedas en una representación espacio/estado. El planeador de movimientos encuentra alguna o la mejor ruta (puede ser la más corta), usando el algoritmo de Dijkstra o A*, que lo lleven de un origen a un destino utilizando un mapa topológico (nodos y conexiones) del lugar.

Reconocedor de excepciones. Si durante la ejecución de un plan suceden cosas inesperadas y el plan necesitara este módulo ofrece esta capacidad.

1.5. Capa de conocimiento

Esta capa contiene una representación del conocimiento con la que cuenta el robot.

Modelo del mundo. Contiene una descripción y la relación que hay entre los objetos, las personas y los lugares con los que interactúa el robot.

Cartógrafo. Este módulo contiene los mapas que representan el medio ambiente, los cuales pueden ser creados en forma manual o por el robot usando algoritmos de creación de mapas (SLAM).

Representación del conocimiento. Usando un sistema basado en reglas, CLIPS (NASA), se representa parte del conocimiento del robot. A través de reglas de producción, que corresponden a ciertas condiciones que ocurren en situaciones cotidianas.

Aprendizaje. El sistema puede aprender a resolver problemas nuevos usando:

- Algoritmos genéticos y programación genética.
- Métodos probabilísticos aplicando cadenas de Markov.
- Agrupamiento utilizando cuantización vectorial.
- Redes neuronales artificiales.

1.6. Capa de ejecución

En esta capa se efectúan los planes generados en la capa de planeación.

Ejecutor. El ejecutor trata de implementar los planes de acciones y de movimientos utilizando procedimientos que resuelven parcialmente problemas específicos.

Supervisor. El supervisor recibe una copia del plan que tiene que implementar el ejecutor y supervisa que se aplique en tiempo y forma.

Comportamientos de reacción. Este subsistema evita obstáculos desconocidos no contemplados por el planeador de movimientos. Los comportamientos pueden

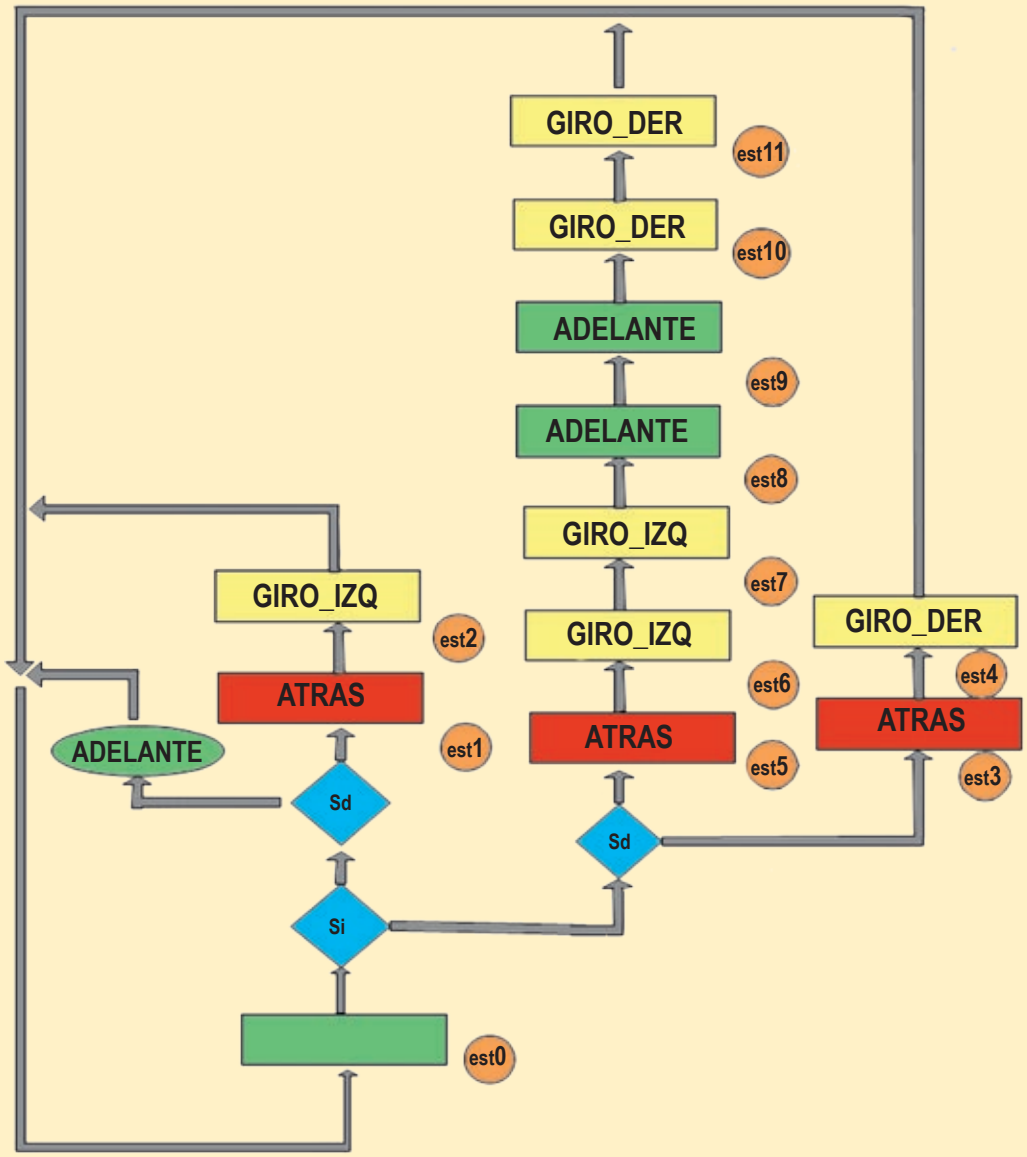


Figura 4: Algoritmo para evadir obstáculos.

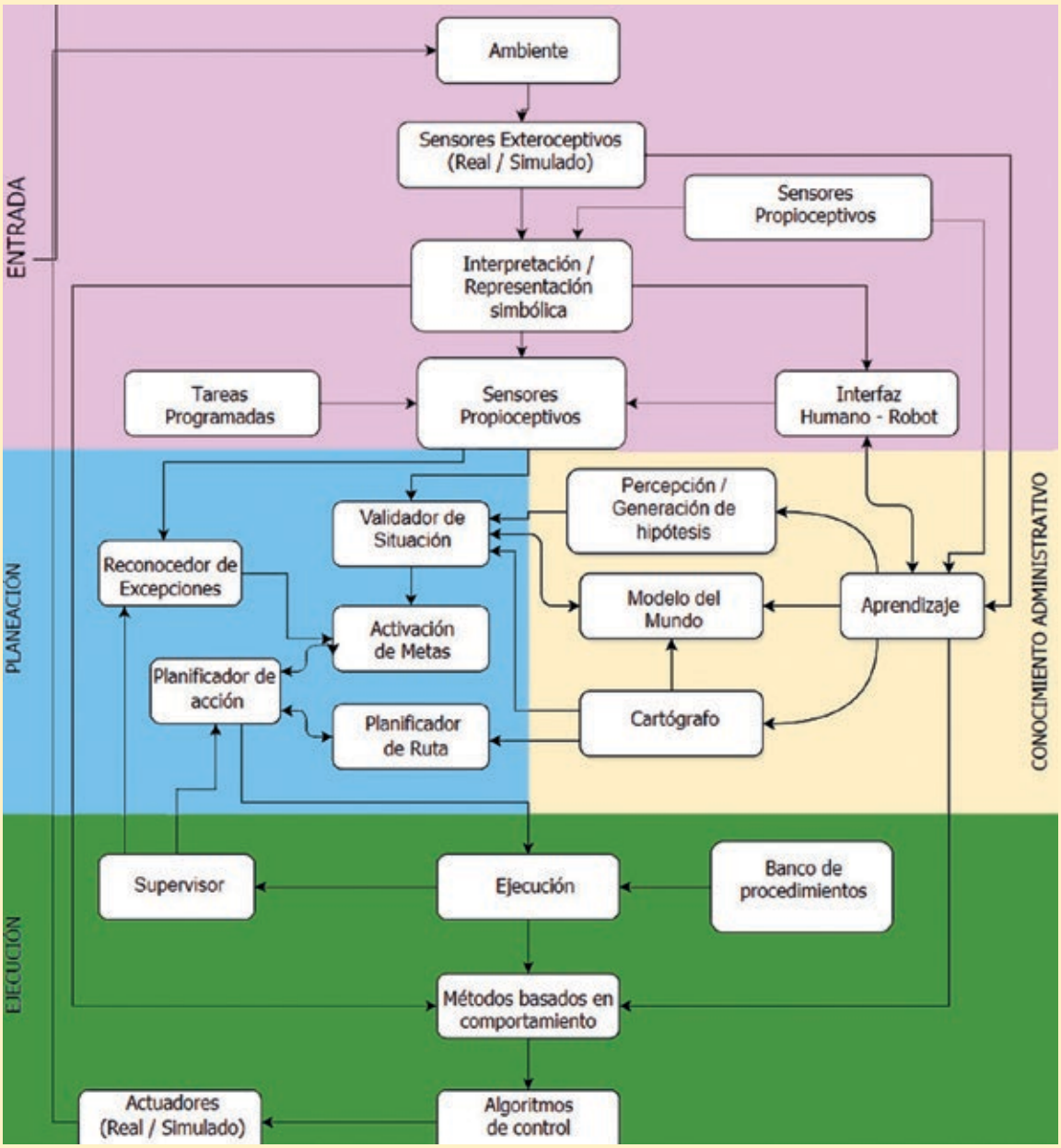


Figura 6: Modelo conceptual de la robot *Justina*.

ser máquinas de estados, campos potenciales, redes neuronales, etcétera.

Algoritmos de control. Son usados para controlar el funcionamiento de los actuadores; en su forma más simple se utilizan controladores PID.

Actuadores. El robot real o virtual recibe los comandos y los aplica. Una interfaz gráfica 3D permite probar diversos robots virtuales, que son una simulación muy próxima de los reales. Los algoritmos que el robot ejecutará se pueden probar primero en los robots virtuales y después en el real. Los robots virtuales pueden ejecutar las mismas órdenes que los robots reales, así como los comportamientos, las ecuaciones de movimiento y las lecturas de los sensores. El ambiente virtual se visualiza por un motor de 3D usando Rviz y Gazibo bajo el sistema operativo de robots (ROS).

Referencias

1 https:/en.wikipedia.org/wiki/Automotive_industry
2 "A Robot in Every Home: Overview/The Robotic Future". Bill Gates, *Scientific American* (2007).
3 http://www.ai.sri.com/shakey/
4 https://www.irobot.com/For-the-Home/Vacuuming/Roomba.aspx
5 S. Thrun *et al.*, "Stanley: The Robot that Won the DARPA Grand Challenge", *Journal of Field Robotics* 23(9), 661-692, 2006.
6 Jesús Savage, Jesús Cruz, Mauricio Matamoros, David A. Rosenblueth, Stalin Muñoz, Marco Negrete, *Configurable Mobile Robot Behaviors Implemented on FPGA Based Architectures*, ICARSC 2016, the 16th IEEE Int. Conf. on Autonomous Robot Systems and Competitions, pp 317-322, Portugal, 2016.
7 Jesús Savage, Marco Negrete, Mauricio Matamoros, Jesús Cruz, Ramón Lagunas and Jaime Márquez, *The Role of Robotics Competitions for the Development of Service Robots*, IJCAI'16, Workshop on Autonomous Mobile Service Robots, New York, USA, 2016.

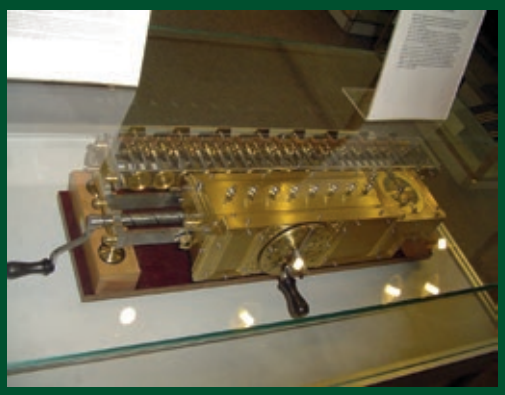
Pasado y presente del cómputo científico

Dra. Suemi Rodríguez Romo y Dr. Ricardo Paramont Hernández García
Facultad de Estudios Superiores Cuautitlán UNAM

Desde la Antigüedad, los seres humanos se percataron de la importancia de realizar cálculos para resolver determinados problemas que requerían de una secuencia de estos para ser resueltos, y muchas veces esas secuencias variaban según el caso. Por ejemplo, determinar el porcentaje de una herencia para el hijo primogénito en que ésta fuera mayor que para el hijo más joven. Aquí se pueden identificar dos problemas: la realización de las operaciones aritméticas de suma, resta, multiplicación y división, y la secuencia en la que se debían aplicar las operaciones.

Pasos iniciales

Los primeros ingenios para hacer de manera mecánica las operaciones de suma y resta (Schickard y Pascal), y las de multiplicación y división mediante sumas y restas sucesivas (Leibniz), fueron creados en el siglo XVII. Fue hasta finales del siglo XIX que se construyó una máquina para realizar directamente la operación de multiplicación (Bollée) y hasta principios del siglo XX, otra que hacía la multiplicación y la división de manera directa (Monroe).



En el siglo XIX, Charles Babagge diseñó una máquina que no acabó nunca de construir; el aparato se basaba en tecnología mecánica, usaba mecanismos de relojería y telares franceses. Este ingenio mecánico era digital y estaba diseñado para usar números de base diez.

A principios del siglo XX, Leonardo Torres y Quevedo fue el primero en notar que la máquina diseñada por Babagge podía ser desarrollada con componentes electromecánicos. Más avanzado el siglo XX, se demostraría que la electromecánica era la siguiente tecnología que haría avanzar la construcción de los autómatas de cálculo. Un par de años antes del inicio de la Segunda Guerra Mundial, George Stibitz, quien trabajaba para los Laboratorios Bell, inició la elaboración de calculadores, usando tecnología electromecánica. Trabajaban con números binarios para números complejos.

Izquierda. Réplica de la máquina de Leibniz en el Museo Technische Sammlungen de Dresde, Alemania.
Arriba. Leonardo Torres y Quevedo.
Derecha. En 1920 se presenta el aritmómetro electromecánico, considerado el primer ordenador por su capacidad de realizar comparaciones, tomar decisiones y almacenar información.

Tiempos modernos

En 1944, como resultado de cinco años de colaboración entre la empresa IBM (Clair D. Lake) y la Universidad de Harvard (Howard Aiken), se construyó el Calculador Automático de Secuencia Controlada (ASCC) o MARK I.

La siguiente etapa en el desarrollo de las máquinas de cálculo se dio en la Escuela Moore, de la Universidad de Pensilvania, por encargo del Laboratorio de Investigación en Balística, donde se diseñó una máquina de cálculo por Eckert, Mauchly y Goldstine, quienes trabajaban para esta escuela. Se trataba de la computadora electrónica llamada Integrador Numérico Electrónico y Computadora (ENIAC). La memoria principal y los dispositivos de cálculo eran circuitos electrónicos. Esta máquina trabajaba con números digitales base diez. La coordinación del funcionamiento de sus componentes se diseñó para que operara con un reloj. Fue entregada en 1946.

Augusta Ada King, condesa de Lovelace, conocida como Ada Lovelace. Matemática y escritora británica considerada como la primera programadora de ordenadores. Su fama se debe principalmente a su trabajo sobre la máquina analítica de Charles Babbage, lo que se reconoce hoy como el primer algoritmo para ser procesado por una máquina.

Fue en la compañía IBM donde se construyó la máquina que almacenaría sus programas en la memoria principal, de la misma forma que se apilaban los datos; se trataba del IBM Calculador Electrónico Seleccionador de Secuencias Controladas (IBM SSEC). Este dispositivo ya tenía las principales bases de la arquitectura de las computadoras modernas. El encargado de su desarrollo fue Frank Hamilton, que en 1948 concluyó su construcción.

Desde finales de la década de los años cuarenta del siglo XX, quienes podían programar las computadoras eran los expertos que conocían su arquitectura y su lenguaje de máquina exclusivamente.

A mediados de la década de los sesenta, Gordon Moore, un experto en la elaboración de los circuitos integrados para computadoras, publicaba una predicción: el número de componentes que se colocaría en uno de estos circuitos se duplicaría cada año. Esto resultó cierto hasta 1975. Para los siguientes 10 años predijo que la duplicación de los componentes ocurriría cada dos años; después de medio siglo, esa predicción sigue vigente. Esto nos habla de los vertiginosos avances en la miniaturización de componentes. Acompañando a esa miniaturización, viene un aumento en su desempeño, de manera que operan también más rápidamente.

Con el advenimiento de los lenguajes de alto nivel, la programación se democratizó. En 1981 salió a la venta al público la primera computadora personal de IBM profundizando la ampliación del cómputo. A partir de entonces, la interacción de las personas con las computadoras fue directa.

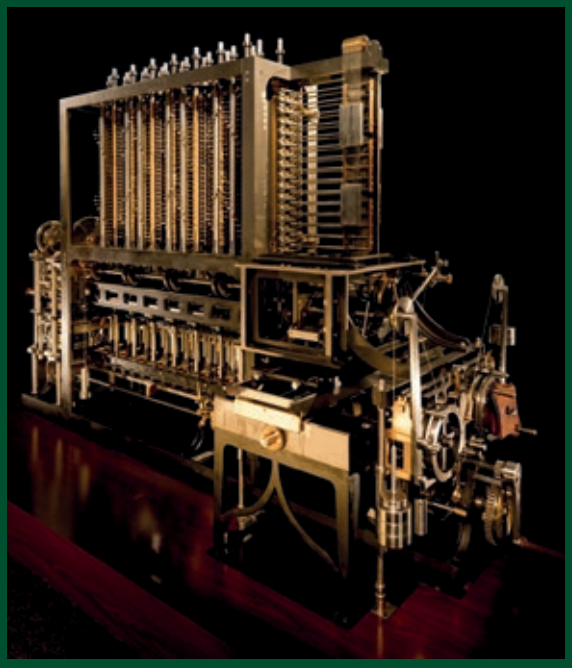
Usos

En la relación de la computadora y el individuo se proponen modelos de lo que nos interesa y se simula lo que pasaría con estos modelos en distintas circunstancias. Más aún, el hecho de tener una gran cantidad de información que aparentemente no es útil, también genera la necesidad de usar la ciencia e ingeniería de lo computable. Algunos ejemplos sencillos de entender son los siguientes.

No podemos hacer experimentos con el ciclo de vida de las galaxias o de las estrellas, pero sí podemos proponer un modelo y simular lo que arrojaría como datos para compararlo con la realidad. También podemos obtener datos de observaciones y tendríamos una base de datos muy amplia que, sin la necesidad de modelo alguno, nos pueda proporcionar información directa del fenómeno en cuestión. Exactamente lo mismo ocurre con los fenómenos climáticos, comportamiento de bolsas de valores y efectos económicos, entre otras aplicaciones. En otras ocasiones el problema al que nos enfrentamos es de seguridad, como es el caso de las plantas nucleares, estabilidad de grandes construcciones, diseño de carros, aviones, materiales y moléculas con actividad biomédica, entre otros.

El cómputo científico es un pilar transdisciplinario indispensable de

La máquina analítica de Charles Babbage tenía dispositivos de entrada basados en las tarjetas perforadas de Jacquard, un procesador aritmético, que calculaba números, una unidad de control que determinaba qué tarea debía ser realizada, un mecanismo de salida y una memoria donde los números podían ser almacenados hasta ser procesados. El Museo de Ciencia de Londres, custodio de la obra de Babbage, completó en 1991 la máquina diferencial Nº2 para celebrar el bicentenario del nacimiento del matemático. Con este proyecto Doron Swade, curador de computación de dicho museo y su equipo, demostraron que tanto la máquina diferencial de Babbage como el sistema de impresión funcionaban correctamente.



(casi) todas las investigaciones científicas y el desarrollo tecnológico en las universidades, instituciones gubernamentales y el sector privado.

Un equipo de investigación moderno debe incluir a varias personas expertas en diversos campos como las matemáticas, las ciencias, estadística, ingeniería y con conocimientos en ciencia e ingeniería de la computación en campos como la visualización, los paradigmas de la programación, la ciencia de datos y el cómputo de alto rendimiento, entre otras áreas. Estos equipos serán capaces de resolver problemas fundamentales de la ciencia, la ingeniería, la tecnología, ciencias sociales y humanidades gracias a numerosas herramientas computacionales; el reconocimiento de patrones, los fundamentos de los algoritmos, la inteligencia artificial y el aprendizaje supervisado o no supervisado, entre otras.

Algunas ciencias, como la Química, podrían sufrir cambios drásticos. ¿Quieres sintetizar una molécula en particular para alguna explicación específica? Como diría Pablo Carbonell, de la Universidad de Manchester, en Inglaterra, usa tu nuevo asistente de laboratorio: la inteligencia artificial. Para eso el equipo de Marwin Segler, de la Universidad de Münster, en Alemania, ya usó exitosamente redes neuronales para evitar la retrosíntesis, el procedimiento normalmente usado y que se caracteriza por ser muy lento. No en vano el presidente francés Emmanuel Macron está dispuesto a invertir 1,500 millones de euros en el área de inteligencia artificial en su país.

Cuando los estudiantes, aun los de doctorado, tienen tiempo libre, pueden descubrir cosas interesantes. Alberto Martín-Martín, estudiante de bibliometría y ciencias de la comunicación de la Universidad de Granada, dedicó su verano a obtener artículos gratis del motor de búsqueda de Google Scholar, burlando su sistema de verificación para detectar humanos 'CAPTCHAs' cientos de veces, acción nada fácil; su éxito fue rotundo.

La ciencia e ingeniería de la computación no tienen un carácter ético intrínseco. Es quien las usa el que asume esta condición. Muchas voces claman con alarma sobre el destino, en nuestra opinión ineludible, de la humanidad en un contexto pleno de avances en cómputo. La respuesta es simple: será el que la humanidad misma decida. Esto debido a las inquietudes éticas derivadas de lo hecho por Martín-Martín, Cambridge Analytica o Facebook, y lo poco que nosotros sabemos del manejo que se da a nuestros datos ya que estos ejemplos solamente son la punta del iceberg.

El inglés Charles Babbage (1791-1871), profesor de la Universidad de Cambridge, diseñó y desarrolló parcialmente una calculadora mecánica capaz de calcular tablas de funciones numéricas por el método de diferencias. También diseñó, pero nunca construyó, la máquina analítica para ejecutar programas de tabulación o computación; por estos inventos se le considera como "El Padre de la Computación".
Arriba. El *Mark I* fue el primer ordenador electromecánico construido en IBM y enviado a la Universidad de Harvard.

México y la UNAM

Se preguntarán el estado de la investigación e innovación en cómputo en México, y aun cuando abordamos varios temas mencionados, necesitamos avanzar mucho y más rápido solamente para estar a la altura de lo que se hace ahora en el resto del mundo.

En cuanto a lo que hacemos en la UNAM, seguimos siendo uno de los principales actores a 60 años del cómputo en nuestra *alma mater*, como lo evidencia nuestra historia. Sin embargo, y debido a que se necesita la formación de grupos transdisciplinarios para lograr avances significativos, nuestra estructura académico-administrativa no ha ayudado mucho.

Ante este panorama, el rector Enrique Graue Wiechers creó el Centro Virtual de Computación (CViCom) (https://cvicom.unam.mx/), el 25 de febrero de 2017, que organiza su primer coloquio del 8 al 10 de agosto del año en curso, donde conoceremos y debatiremos sobre las distintas líneas de investigación abordadas por nuestra comunidad actualmente.

REFERENCIAS

R. Moreau, *The computer comes of age*. Traducción del francés de J. Howlett, The MIT Press, Cambridge, Massachusetts, Capítulo 1 "The birth of the computer", pp. 4-47, 1984.
https://www.obliquity.com/computer/fortran/history.html, consultada el 18/abril/2018.
Gerard O'Regan *Introduction to the history of computing. A computing history primer*. Springer. Suiza, p. 97, 2016.
Elizabeth Gibney. *How to blow up a star.* "Supernova simulations are resolving a 50-year-old mystery about stellar death throes". *Nature* 556, 287-289, 2018.
Holly Else. "How I scraped data from Google Scholar. A researcher explains how — and why — he spent a whole summer harvesting information from the platform, which is notoriously hard to mine". *Nature news*, 11/ abril /2018.

Códigos de cadenas en la UNAM

Dr. Ernesto Bribiesca Correa
Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, UNAM

La trayectoria del vuelo de un avión; la ruta que sigue una abeja al encontrar una fuente de polen; la señal de un electrocardiograma; el contorno de un tumor; la estructura de un árbol o de un coral son ejemplos que pueden ser representados y analizados por códigos de cadenas. Un código de cadena es una secuencia ordenada de símbolos o caracteres, una representación unidimensional y se mantiene constante ante las transformaciones afines, como el cambio de posición del objeto, la rotación, el origen de construcción y opcionalmente ante cambios de tamaño. Así, una cadena es una secuencia ordenada de símbolos que forman un alfabeto finito, permitiendo el uso de técnicas gramaticales para la clasificación y análisis de la forma de los objetos.

El primer código de cadenas publicado corrió a cargo del profesor Herbert Freeman[1], de la New York University, en 1961. Este código se usa para representar curvas en dos dimensiones y regiones planas, por medio de cadenas. Cada elemento de la cadena representa un segmento de línea recta de longitud previamente definida a diferentes direcciones a múltiplos de 45 grados.

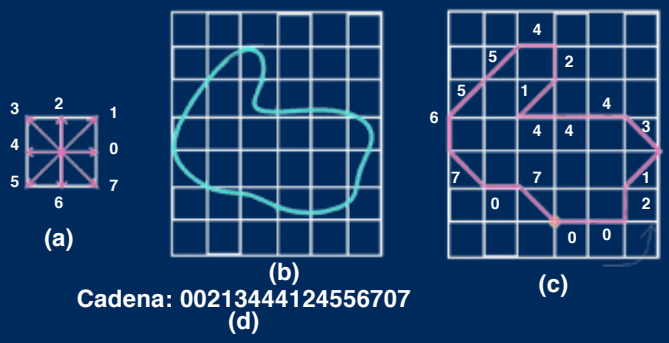


Figura 1. Cadenas de Freeman: (a) elementos de las cadenas; (b) un ejemplo de una curva cerrada simple; (c) curva discretizada indicando los elementos del código de Freeman; (d) la cadena de elementos de la curva.

En la figura 1(a) se aprecian las diferentes direcciones de los elementos de las cadenas de Freeman. Los elementos van del 0 al 7; los elementos 0, 2, 4 y 6 tienen una longitud de uno, y los elementos 1, 3, 5 y 7 una longitud de raíz de 2. Todas sus posiciones son discretas, es decir, inician y terminan en un vértice de la cuadrícula. De esta manera, cualquier curva en el plano ya sea cerrada o abierta puede ser representada usando este código.

La figura 1(b) muestra un ejemplo de una curva simple cerrada. La figura 1(c) muestra la versión discretizada compuesta por 17 segmentos de línea recta de la curva mostrada en (b) indicando sus 17 elementos de la cadena. Finalmente, la figura 1(d) nos enseña la cadena de la forma de la figura mostrada en (b), la cual es 00212444124556707. El origen de la curva está representado por un pequeño círculo. Conociendo la cadena, las coordenadas (x,y) del origen de la curva y el tamaño de la cuadrícula es posible reconstruir la curva discreta sin pérdida de información. Así, una característica de los códigos de cadena es su capacidad de compresión sin pérdida de información.

El Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS) de la UNAM ha tenido una participación activa en la conceptualización y diseño de códigos de cadena; tal ha sido su participación, que en la actualidad algunos de sus códigos forman parte de la literatura internacional del área. En 1980, con la participación del IIMAS se concluye la teoría de los *shape numbers*[2], que consiste en representar cada forma por un número entero, en una relación biunívoca, es decir, cada forma solo tiene un número que la representa: no hay dos formas con el mismo número, ni un número que represente a dos formas diferentes.

Los números de forma se representan a diferentes niveles de resolución, permitiendo así tener una medida de similitud entre formas. Así, el número de forma lleva de una manera implícita la geometría de la forma que representa. A continuación se mencionan tres de los códigos de cadenas elaborados en el IIMAS que han tenido un impacto considerable.

Código de cadena de vértices

En 1999 se publicó el Código de Cadena de Vértices[3] (VCC, siglas en inglés de Vertex Chain Code). Este código fue desarrollado en el IIMAS y diseñado especialmente para usarse con objetos extraídos de imágenes digitales, por lo tanto cada objeto está compuesto por un número finito de pixeles conectados unos a otros. El contorno o perímetro de cualquier región discreta compuesta

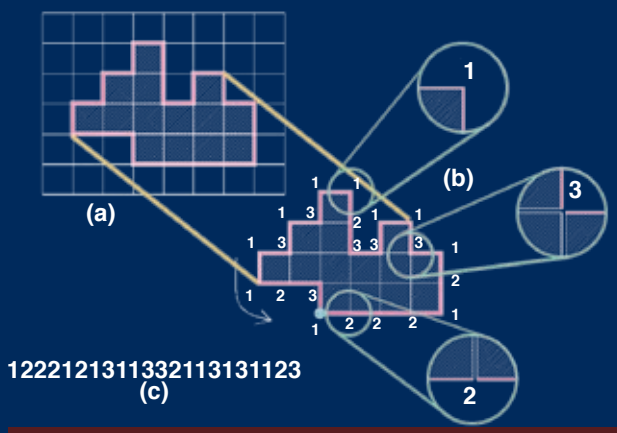


Figura 2. El código de cadena de vértices: (a) un ejemplo de una región compuesta por pixeles; (b) generación del VCC; (c) cadena de elementos de la región presentada en (a).

Figura 3. Oclusión de objetos usando el VCC: (a) Un ejemplo de hoja de higo; (b) elementos de la cadena de la hoja de higo; (c) un ejemplo de hoja de limón; (d) elementos de la cadena de la hoja de limón; (e) traslape entre las hoja de higo y de limón; (f) elementos de la cadena del traslape de la hojas antes mencionadas.

de pixeles puede representarse como una cadena. El VCC significa el número de vértices de los pixeles de la forma que tocan el contorno de la región. La figura 2(a) muestra un ejemplo de una región compuesta de 14 pixeles. La figura 2(b) muestra los elementos del VCC. Es fácil notar que el elemento "1" corresponde a un vértice que toca el contorno. El elemento "2" corresponde a dos vértices de dos pixeles. Finalmente, el elemento "3" corresponde a tres vértices de tres pixeles de la región.

Así, de una manera muy simple, se van generando los elementos del VCC de cualquier región. La figura 2(c) muestra la cadena considerando el origen representado por un punto. Así, la cadena de elementos del ejemplo de la forma mostrada en la figura 2(a) es 1222121311332113131123; es importante hacer notar que esta cadena de elementos no está basada en símbolos como en el código de cadenas de Freeman, que bien podrían ser letras o números. En el VCC los elementos de la cadena nos hablan de propiedades intrínsecas de la forma que representan, nos indican el número de vértices que toca el contorno. ¿Cuántos son rectos? Número de doses ¿Cuántos son cóncavos? Número de treses ¿Cuántos son convexos? Número de unos. Así, el VCC permite tener un descriptor único de forma, el cual indica propiedades geométricas.

En la figura 3 se muestra una aplicación del VCC en la determinación y detección de objetos con oclusión; la identificación de objetos con oclusión es un tópico importante en el área de reconocimiento de patrones. Así, las características locales de los objetos son de importancia cuando hay oclusión de los mismos. La oclusión se da cuando un objeto es sobrepuesto con otro. Las figuras 3(a) y (b) muestran un ejemplo de una hoja de higo y sus elementos de cadena, las 3(c) y (d) ilustran una hoja de limón y su cadena, respectivamente. Finalmente, las figuras 3(e) y (f) muestran el traslape entre ambas hojas. Note también que ambas hojas están rotadas y que se mantiene constante parte de las cadenas de ambas hojas, permitiendo así la identificación de propiedades locales. El VCC ha tenido diferentes tipos de aplicaciones, sobresale no solo para reconocimiento de objetos sino para ser usado en técnicas de compresión.

Código de cadena de cambio de pendiente

El código de cadena de cambios de pendiente[4] (SCC, siglas en inglés de Slope Chain Code). El código de cadena SCC de una curva bidimensional se obtiene por medio de ir colocando segmentos de línea recta alrededor de la curva (los puntos finales de los segmentos siempre tocando la curva, así se obtiene una mejor descripción de la curva), posteriormente se calculan los cambios de pendiente entre los segmentos contiguos y se normalizan en una escala de -1 a 1 en vez de -180 a 180 grados. El SCC puede ser calculado para curvas bidimensionales abiertas o cerradas[4] y también para estructuras arborescentes[5]. Entre las principales características del SCC sobresalen su independencia en traslación, rotación, escala de manera opcional; el tamaño de sus segmentos es constante para toda la curva, no usa ninguna cuadrícula, sus cambios de pendiente son ilimitados del -1 al 1 y sus elementos producen alfabetos finitos, donde es posible utilizar técnicas gramaticales para su clasificación.

La figura 4 muestra cómo se calcula el SCC de una curva, la figura 4(a) ilustra un ejemplo de una curva: primero se selecciona la longitud del segmento que se va a utilizar (figura 4(b)). Segundo, se escoge un punto final sobre la curva como el origen (figura 4(c) representado por un punto). Tercero, se traza una circunferencia en el origen de la curva de radio igual al tamaño del segmento seleccionado. La intersección entre la circunferencia y la curva determina el siguiente centro de la siguiente circunferencia, de esta manera se va formando la curva discreta y así sucesivamente, como se muestra en la figura 4(c).

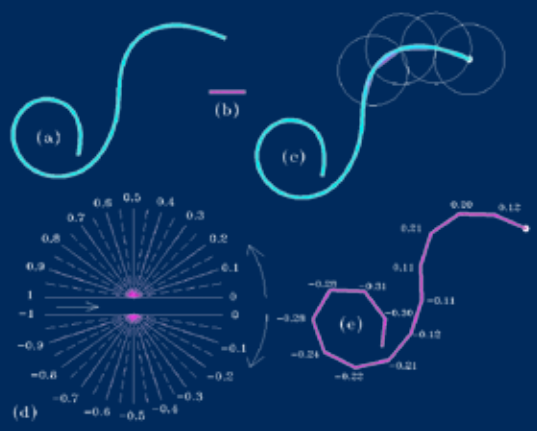


Figura 4. Cálculo del SCC de una curva: (a) un ejemplo de una curva continua; (b) tamaño del segmento; (c) cálculo de la curva discreta; (d) cambios de pendiente del -1 al 1; (e) curva discreta con sus elementos de cadena.

La figura 4(d) muestra el rango de cambios de pendiente normalizados del -1 al 1 y finalmente en la figura 4(e) se muestra la curva discreta obtenida con todos sus cambios de pendiente, que forma una cadena de elementos: 0.12, 0.20, 0.21, 0.11, -0.11, -0.12, -0.21, -0.22, -0.24, -0.28, -0.28, -0.31, -0.30. Usando el VCC es fácil obtener la inversa de una cadena recorriendo los elementos en sentido inverso y cambiándolos de signo; geométricamente es recorrer una curva abierta en el otro extremo. También el cálculo de la cadena de una imagen espejo de un objeto se obtiene por medio del cambio de signos de los elementos de la cadena.

Una aplicación interesante del SCC es el cálculo de la tortuosidad de una curva, que está relacionada con la cantidad de vueltas y excentricidad de las mismas; se calcula por medio de la suma absoluta de todos los cambios de pendiente. La tortuosidad es un parámetro importante de medición de los vasos sanguíneos. Enfermedades tales como diabetes, hipertensión y patologías vasculares afectan la morfología de los vasos sanguíneos. Un ejem-

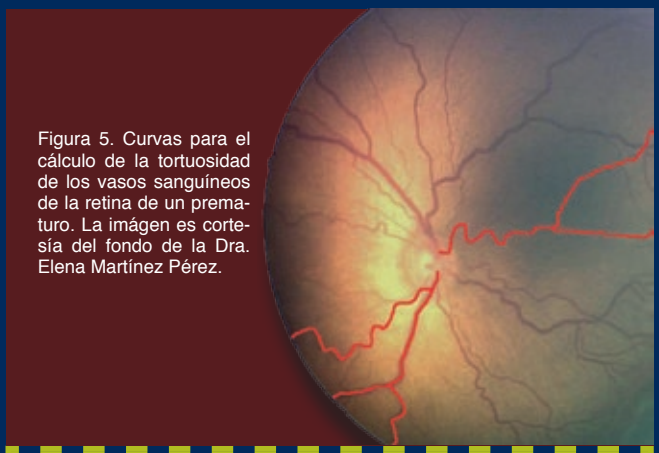


Figura 5. Curvas para el cálculo de la tortuosidad de los vasos sanguíneos de la retina de un prematuro. La imágen es cortesía del fondo de la Dra. Elena Martínez Pérez.

plo donde la tortuosidad es una métrica importante es en el apoyo del diagnóstico de la retinopatía de los prematuros (ROP, por sus siglas en inglés). ROP es una enfermedad del ojo que afecta a los prematuros. La figura 5 muestra una imagen típica de la retina de un prematuro y las curvas de los vasos sanguíneos a los que se les calculó la tortuosidad. Usando el SCC también es posible representar estructuras de árbol, esto ha extendido sus aplicaciones, por ejemplo, identificación de especies de equinodermos[5]. Ver figura 6.

Código de cadena para curvas 3D

La representación de curvas en el espacio es de gran importancia en diferentes campos de la ciencia debido al gran número de aplicaciones que tienen. Así, al IIMAS en el año 2000 la revista *Pattern Recognition* le publica el código de cadenas para curvas 3D[6], basado en la teoría de los *shape numbers*[2] y en los 3-D *stick bodies*[7]. Este es el primer código de cadenas a nivel mundial en ser invariante a la rotación. Así, los elementos de la cadena para representar cualquier curva discreta 3D están definidos como sigue:

Figura 6. Identificación de especies de equinodermos por medio del uso del SCC para estructuras arborescentes. Imagen cortesía de la M. en C. Guadalupe Bribiesca Contreras.

El elemento de cadena “0” indica que no hay cambio de dirección del último segmento, es decir, sigue en línea recta y está representado en la figura 7 en la parte superior con el color verde.

El elemento “1” indica un cambio de dirección a la derecha considerando el cambio de dirección anterior y se muestra en la figura 7 en la parte superior con el color cian.

El elemento “2” indica un cambio de dirección tipo escalera y se muestra en la figura 7 en la parte superior con el color amarillo.

El elemento “3” indica un cambio de dirección a la izquierda considerando el cambio de dirección anterior y se muestra en la figura 7 en la parte superior con el color magenta.

El elemento “4” describe un cambio de dirección de regreso, es decir, de vuelta en U como se ilustra en la figura 7 en la parte superior con el color rojo.

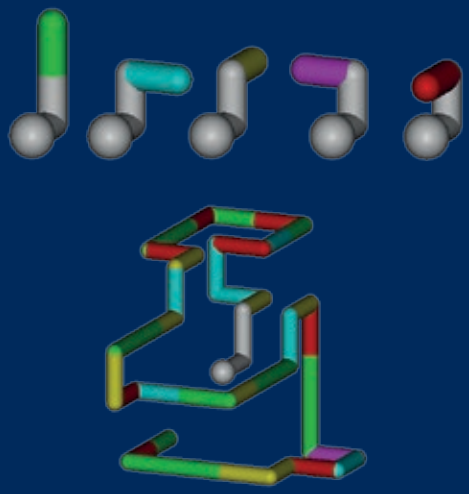


Figura 7. Definición de los elementos de código de cadenas 3D en la parte superior, de izquierda a derecha: “0”, “1”, “2”, “3” y “4”, respectivamente. En la parte inferior de la figura un ejemplo de una curva discreta.

En la parte inferior de la figura 7 se presenta un ejemplo de una curva discretizada. La forma de calcular la cadena es la siguiente: primero, se selecciona un origen de la curva, en este caso representado por una esfera; segundo, se selecciona el primer elemento que corresponde a la curva, en este caso, es el “2”, para calcular cada cambio de dirección es necesario considerar el último ángulo recto.

Es muy importante imaginarnos que vamos caminando sobre la curva, estamos frente a un ángulo recto, ¿cuál es nuestro siguiente paso? Los dos primeros segmentos al iniciar la curva no son etiquetados, debido a que sirven de referencia para definir el primer elemento, es decir, al “2”; para calcular el siguiente elemento hay que considerar los dos segmentos anteriores y corresponde al “1”, y así sucesivamente, hasta terminar la curva.

Finalmente, la cadena de toda la curva es 211410 4040042120024102001240031422. Este código para

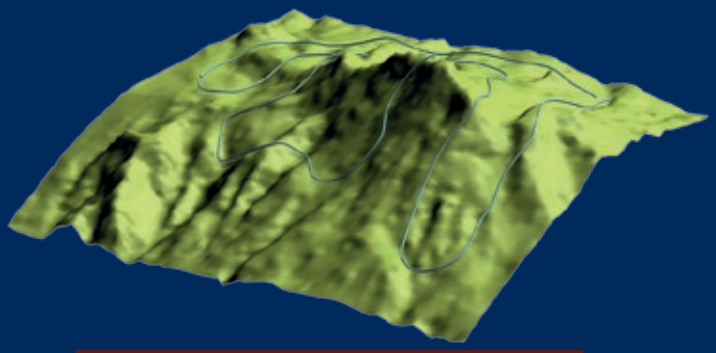


Figura 8. Trayectoria de un avión volando sobre el volcán Iztaccíhuatl.

curvas 3D es invariante a la traslación y rotación; también es fácil calcular la cadena de la curva espejo por medio del remplazo de los elementos “1” por “3” y viceversa los “3” por “1”.

Este código de cadenas 3D ha tenido importantes aplicaciones, como la representación de la trayectoria de un avión volando sobre el volcán Iztaccíhuatl. Una ventaja que se obtiene de usar este código es su representación compacta. La figura 8 muestra el volcán Iztaccíhuatl y la trayectoria de la ruta del avión.

Otra aplicación importante de este código ha sido la descripción de algunas curvas matemáticas conocidas, tal como la curva de Hilbert[8]. Esta curva es famosa porque tiene un patrón que se repite parecido a una silla de montar y en sus diferentes etapas va cubriendo el espacio que la contiene. La figura 9 muestra una de las etapas de la curva de Hilbert compuesta por 4096 elementos.

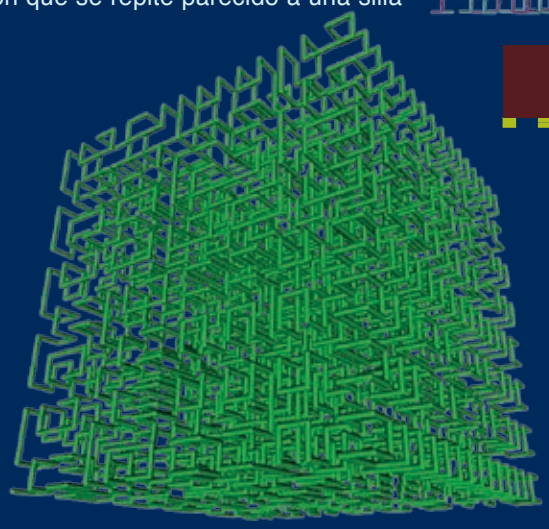


Figura 9. Curva de Hilbert compuesta por 4,096 elementos, usando el código cadena de curvas 3D.

El código de cadenas 3D también es útil para representar estructuras de árbol. En la figura 10 se describe un árbol que cubre el espacio usando solamente los elementos “1” y “3” del código en una manera iterativa.

A nivel internacional, el código de cadenas 3D ha tenido interesantes aplicaciones. Algunas de ellas se enlistan a continuación: el Dr. Steve Huffman, de la Universidad de Leipzig, en Alemania, lo ha usado para representar cadenas de DNA; en la Universidad de Colorado, en los Estados Unidos de América, lo han empleado para el análisis de movimientos del cuerpo humano, en aplicaciones de rutinas de ballet clásico y danza moderna[9], también en el análisis de movimientos en terapias de rehabilitación.

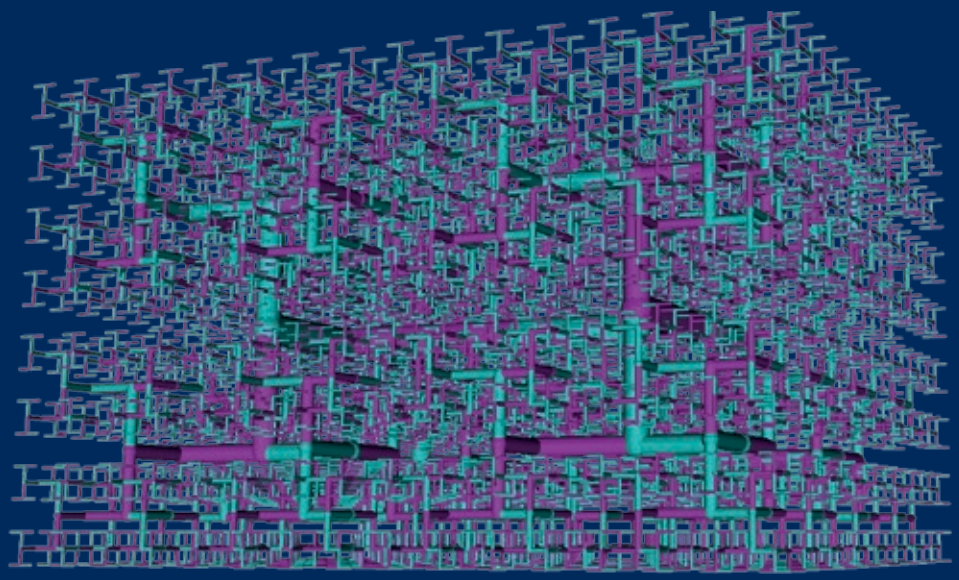


Figura 10. Árboles que llenan el espacio, usando el código cadena de curvas 3D.

Bibliografía

[1] H. Freeman, *On the encoding of arbitrary geometric configurations*, IRE Transactions on Electronic Computers EC-10 (1961) 260-268.

[2] E. Bribiesca and A. Guzmán, *How to describe pure form and how to measure differences in shapes using shape numbers, Pattern Recognition 12* (1980) 101-112.

[3] E. Bribiesca, *A new chain code, Pattern Recognition 32* (1999) 235-251.

[4] E. Bribiesca, *A measure of tortuosity based on chain coding, Pattern Recognition 46* (2013) 716-724.

[5] E. Bribiesca and G. Bribiesca-Contreras, *2D tree object representation via the slope chain code, Pattern Recognition 47* (2014) 3242-3253.

[6] E. Bribiesca, *A chain code for representing 3D curves, Pattern Recognition 33* (2000) 755-765.

[7] A. Guzmán, *Canonical shape description for 3-d stick bodies*, MCC Technical Report Number: ACA-254-87, Austin, TX 78759, 1987.

[8] W. Gilbert, *A cube-filling Hilbert curve Mathematical Intelli-gencer 6* (1984) 78.

[9] F. Torres and E. Chow, *Human body motion analysis with time-3D curve and Labnotation*, Department of Computer Science, Universidad de Colorado en Colorado Springs, USA, 2010.

De las tarjetas perforadas a Miztli

Patricia de la Peña Sobarzo
y Yassir Zárate Méndez

El 8 de junio de 1958 está marcado en los anales de la UNAM y de México. Formalmente, en esa fecha comenzó la era de la computación en nuestra universidad y en el país. Esta es parte de la historia.

Hace seis décadas, el Centro de Cálculo Electrónico, ubicado en la Facultad de Ciencias de la UNAM, echó a andar la primera computadora del país. La Universidad y México se apuntaban así en una lista sumamente selecta. A la luz de los resultados obtenidos de manera inmediata, se justificó la adquisición de ese equipo, una IBM 650, que si bien no era nueva, ofrecía un rendimiento espectacular, que sirvió en numerosísimos campos.

Un papel fundamental para la llegada de ese dispositivo lo jugaron Sergio Beltrán, Alberto Barajas, Carlos Graeff y Nabor Carrillo, este último en su papel de rector, que supo advertir la trascendencia que iba a tener el cómputo electrónico.

La IBM 650

El 8 de junio de 1958 entró en la historia de la UNAM y de México. Ese día, se creó el Centro de Cálculo Electrónico, cuyo primer director fue Sergio Francisco Beltrán López.

El Centro fue el sitio elegido para albergar la primera computadora que llegó a México, la ya mencionada IBM 650, instalada en la planta baja de la Facultad de Ciencias, en Ciudad Universitaria.

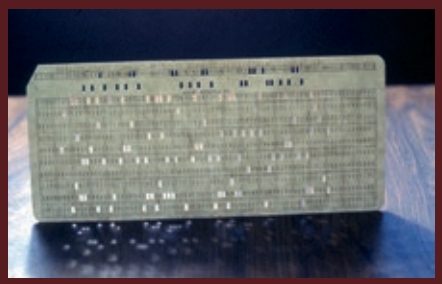
El equipo había pertenecido a la Universidad de California y tenía aproximadamente tres años de uso, aunque no pasó en plena propiedad: la UNAM la rentaba en 25 mil pesos mensuales. Por esta razón, se diseñó un plan de recuperación de gastos, que pasó por la venta de parte del tiempo de operación a empresas y dependencias gubernamentales, como el Instituto Mexicano del Seguro Social y Petróleos Mexicanos (Pemex).

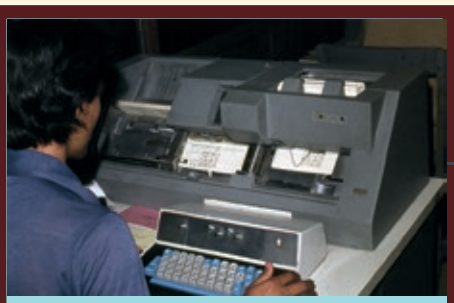
Con la llegada de la IBM 650 inició formalmente la era de la computación en la UNAM y en el país. Además, la creación del Centro de Cálculo Electrónico abrió paso a las primeras clases de programación, inteligencia artificial y lenguajes formales; también albergó numerosos proyectos patrocinados. Además de Pemex, otras entidades gubernamentales o paraestatales, como la Comisión Federal de Electricidad y Nacional Financiera, recibieron apoyo, amén de la comunidad universitaria.

La IBM 650 contaba con 2,000 bulbos y podía procesar un programa de instrucciones almacenado en un tambor magnético que giraba a 12,500 revoluciones por minuto; los datos se insertaban en la máquina por medio de tarjetas perforadas.

En ese entonces la Dirección General de Servicios Escolares de la UNAM operaba con un sistema similar, con el cual proporcionaba el servicio de estadísticas escolares, directorios de alumnos, listas de asistencia, actas de exámenes y boletas de calificaciones.

La tarjeta perforada era básicamente una lámina hecha de cartulina con información en forma de perforaciones según un código biquinario, formado de dos bases numéricas, la binaria (cero y uno) y la quinaria de cinco números, siendo el primer medio utilizado para ingresar información e instrucciones a una computadora.





Izquierda: tarjeta perforada con datos e instrucciones de algún programa para ser cargado en la computadora Burroughs B-6700 en el Centro de Servicios de Cómputo y ser procesada. Arriba: perforadora de tarjetas siendo utilizada por un usuario.

Computadora IBM 650. Alberto Barajas y Carlos Graef en el Centro de Cálculo Electrónico en 1958. Arriba: el ingeniero Sergio Beltrán.

Para dimensionar el espectacular avance que se ha tenido desde entonces, basta con mencionar que esa computadora poseía una memoria limitada, que ahora cualquier calculadora de bolsillo supera con creces.

Impacto y nuevas computadoras

A pesar de funcionar con esas condiciones, la IBM 650 permitió la solución de problemas de física, astronomía, geofísica, ingeniería, economía, sociología, lingüística y procesos administrativos.

Varios institutos y centros de investigación resultaron inmediatamente beneficiados con la IBM 650. Así, el Instituto de Biología desarrolló bioestadística aplicada a problemas taxonómicos; el Instituto de Ingeniería efectuó análisis sísmicos de estructuras y simulación del funcionamiento de una presa; el Instituto de Geofísica efectuó estudios de integración de órbitas de protones primarios, mientras que el Observatorio Astronómico Nacional comprobó densidades, potenciales y velocidades de escape en una galaxia esférica, en el nivel de simulación matemática. En otro contexto, el Instituto de Investigaciones Históricas mejoró un estudio de léxico estadístico de la lengua maya.

La revolución informática estaba encendida y ya nada pudo detenerla. Al contrario. A la IBM 650 siguieron equipos de mayor potencia. En los sesenta la Universidad adquirió dos computadoras Bendix G15, de 16 kb de memoria; el paquete incluyó una Bendix G20 y una Gama 30, de 32 kilobytes. Esta última era una máquina de segunda generación, ya que contaba con transistores en lugar de bulbos.

Estos aparatos eran más rápidos, más pequeños y con menos necesidades de ventilación. Comenzaba así una larga carrera por la compactación, que se mantiene hasta la fecha.

A la par de la adquisición de equipos, la Universidad comenzó a desarrollar programas y lenguajes computacionales. En el Centro de Cálculo Electrónico se habilitaron sendos departamentos de Teoría Matemática de la Programación y de Cibernética y Teoría de la Información, donde se discutían, analizaban y resolvían tópicos como nuevos lenguajes para computadoras, diseño lógico, selección automática de información, desarrollo de modelos matemáticos y simulación, entre otros.

Siendo el primer doctor en computación en el país, Renato Iturriaga de la Fuente, a su regreso del Instituto de Tecnología Carnegie Mellon, propuso al rector Pablo González Casanova la transformación del Centro de Cálculo Electrónico en el Centro de Investigaciones en Matemáticas Aplicadas, Sistemas y Servicios. Este nuevo organismo, dependiente de la entonces Coordinación de Ciencias, fue creado en 1970, iniciando formalmente sus actividades de investigación en computación, en estadística, sistemas y programas. Iturriaga de la Fuente fue nombrado su primer director.

Panorámica de la supercomputadora CRAY YMP/464, también llamada *Sirio*, en la sala de supercómputo de la Dirección General de Servicios de Cómputo Académico.

Después de algún tiempo, en 1973 Iturriaga propuso la división del CIMASS en dos centros: el CIMAS (Centro de Investigaciones en Matemáticas Aplicadas y en Sistemas) y el Centro de Servicios de Cómputo (CSC), quedando al frente el doctor Tomás Garza y el ingeniero Francisco Martínez Palomo, respectivamente. Ambas entidades formaron parte del Consejo Técnico de la Investigación Científica.

Se afianzaba así la nueva era...

El supercómputo

De 1958, cuando la UNAM instala la primera computadora en América Latina y se funda el Centro de Cálculo Electrónico, a 2018, ha ocurrido una revolución tecnológica sin precedentes. Tras la llegada de la IBM 650 a la Universidad, vino en cascada la serie de acontecimientos aquí reseñados.

Después del breve recorrido por los inicios del cómputo en la UNAM, nos avocamos ahora a la etapa siguiente: el supercómputo.

Unidades de disco removible DISK PACK'S de la Computadora Burroughs B-6700 en el Centro de Servicios de Cómputo en 1974.

Computadora UNISYS A12 ubicada en la sala de cómputo del Centro de Servicios de Cómputo.

En marzo de 2003 la DGSCA puso en operación la AlphaServer SC 45, nombrada *Bakliz*, que significa "infinito" en maya. Se trataba de una supercomputadora escalable y de memoria distribuida, 60 veces más potente que la Cray.

Primeras generaciones

A inicios de los noventa, la UNAM suscribió un convenio de investigación y desarrollo con la Cray Research Inc., proveedora de la primera supercomputadora en México y en América Latina. La Cray Y-P4/464, conocida como *Sirio*, se instaló en la Dirección General de Servicios de Cómputo Académico (DGSCA), el 14 de noviembre de 1991. De inmediato se apuntaron 46 grupos de investigación para emplearla. Un año después sumaban 150 proyectos en ciencias químicas, biológicas, físicas, matemáticas de la computación y de materiales e ingeniería, entre otros.

Para abril de 1997, la primera supercomputadora ya había utilizado más del 90% de su capacidad de memoria y se encontraba poco menos que saturada. Entonces llegó a la DGSCA otra Cray, modelo Origin 2000, a la que se bautizó como *Berenice* y que inició operaciones en abril de 1997.

Esta supercomputadora se adquirió por recomendación del Comité Académico de Supercómputo y en un principio se pensó que podría complementar a *Sirio*, ya que era más pequeña. Con las Cray, la UNAM se puso a la vanguardia tecnológica.

En diciembre de 2002, la DGSCA adquirió una nueva supercomputadora con procesamiento en paralelo, la AlphaServer SC 45, nombrada *Bakliz*, que significa "infinito" en maya. Se trataba de una computadora escalable y de memoria distribuida, 60 veces más potente que la *Cray*. Fue puesta en operación en marzo de 2003. *Bakliz* operó en el sistema de administración de recursos Sierra Cluster, el primero en su tipo en Latinoamérica y uno de los pocos en el mundo con la versión Tru64 UNIX 5.1b.

La arquitectura de esta supercomputadora fue tan eficiente, que una versión similar, pero de mayores dimensiones, se encontraba entre los primeros seis lugares del supercómputo mundial. Las investigaciones que respaldó esta máquina se relacionaban con el sida, el ADN, el cáncer y la formación de galaxias, entre otros temas.

KanBalam

En febrero de 2006, la Cámara de Diputados asignó a la UNAM y al IPN fondos adicionales. La Universidad decidió utilizar 30 millones de pesos de esos recursos para "fortalecer la infraestructura de supercómputo". Durante el rectorado de Juan Ramón de la Fuente, la Universidad anunció la puesta en marcha de *KanBalam*, la nueva supercomputadora de HP, el 16 de enero de 2007, tres meses después de recibirla en las instalaciones de la DGSCA; tenía la capacidad para siete billones de operaciones aritméticas por segundo.

KanBalam fue una de las supercomputadoras paralelas más poderosas de América Latina con una capacidad de procesamiento de 7.113 teraflops (7.113 billones de operaciones aritméticas por segundo). Disponía de un sistema de 768 discos duros de 200 GB cada uno, con 160 TB de almacenamiento.

La capacidad de cálculo de *KanBalam* fue siete mil veces más potente que la primera supercomputadora de la UNAM, la CRAY-YMP adquirida en 1991, y 79 veces más poderosa en cálculo que la AlphaServer SC45, comprada en 2003. Además, se colocó en el lugar 126 dentro de las 500 supercomputadoras más potentes del mundo, era la 28 entre las supercomputadoras operadas por universidades y la número uno en América Latina. Así, la UNAM se mantenía a la vanguardia en el empleo de equipos de cómputo electrónico, en beneficio no solo de la comunidad de investigadores, sino del país.

Lo de hoy

Para enero de 2013, la Universidad adquirió una supercomputadora tipo clúster, con una capacidad 17.5 veces superior con respecto a *KanBalam* y catalogada como el equipo más robusto de América Latina.

Izquierda. La supercomputadora Origin 2000, bautizada como *Berenice,* inicia operaciones en 1997. Se adquirió por recomendación del Comité Académico de Supercómputo y en un principio se pensó que podría complementar a *Sirio,* ya que era más pequeña. Con las *Cray*, la UNAM se puso a la vanguardia tecnológica. Derecha. En 2007 se instaló la supercomputadora *KanBalam*, que figuró como la 126 entre todas las del mundo, la 28 entre las supercomputadoras operadas por universidades y la número uno en América Latina.

Vista panorámica de la sala de supercómputo, DGSCA, hoy DGTIC.
En 1991, la UNAM pone a disposición de la comunidad universitaria la primera supercomputadora en Latinoamérica: CRAY YMP 432. Se instalan los primeros conmutadores telefónicos digitales de la red universitaria más grande de Latinoamérica.

Esta supercomputadora, llamada *Miztli*, que en náhuatl significa "felino mayor o puma", puede efectuar más de 118 billones de operaciones por segundo y cuenta con una capacidad de almacenamiento en disco de 720 TB.

En 2013 se inició la consolidación del clúster de alto desempeño para la colaboración con el Centro Europeo de Investigación Nuclear y el Instituto de Ciencias Nucleares de la UNAM, en el ámbito de la física de partículas de alta energía dentro del proyecto ALICE del Gran Colisionador de Hadrones, para lo que se adquirió un disco duro de 500 TB y 1,024 núcleos de procesamiento.

El supercómputo contribuye al estudio de la estructura del Universo, de sismos y del comportamiento de partículas subatómicas, así como al diseño de nuevos materiales, fármacos y reactores nucleares. En promedio, *Miztli* apoya anualmente 120 proyectos de investigación, tanto de la UNAM como de otras instituciones que solicitan recursos de supercómputo del Laboratorio Nacional de Cómputo de Alto Desempeño.

En esta línea, en junio de 2017 *Miztli* amplió su capacidad. Las necesidades para hacer uso de herramientas cada vez más poderosas y emprender nuevas investigaciones en la UNAM han requerido que se vaya adaptando esta supercomputadora para abordar nuevos desarrollos tecnológicos. Ahora cuenta con 8,344 procesadores, casi 45 terabytes de memoria, 750 terabytes de almacenamiento y 228 teraflops de rendimiento teórico, equivalentes a 86,365 computadoras personales de última generación. Se trata de una gran supercomputadora, para una gran universidad.

El futuro de la computación

En el marco de la celebración por los 60 años de cómputo en la UNAM, recientemente se efectuó una conferencia sobre computadoras cuánticas y el futuro de la informática, impartida por Robert Sutor, vicepresidente de investigación de IBM y líder del proyecto mundial más exitoso en cómputo cuántico.

Procesar información usando estados cuánticos en lugar de corriente eléctrica dará a las computadoras de nueva generación un poder sin precedentes y seguridad con garantías sin paralelo. Estas nuevas tecnologías comienzan a tener presencia a nivel experimental; hoy en día son de 15 bits cuánticos o qubits (unidades básicas de información en este sistema), pero en los próximos años contendrán cientos de ellos.

Entre las múltiples aplicaciones de estos equipos están simulaciones complejas de medicamentos para las ciencias químicas, la encriptación cibernética para tener mayor seguridad de datos e Internet, y la simulación astronómica. Hoy en día su grupo de investigación en IBM construye un sistema con 50 qubits, el cual se espera revolucione la tecnología.

Nuestra Máxima Casa de Estudios se mantiene a la vanguardia con la especialización en Cómputo de Alto Desempeño y diseña la nueva licenciatura en Ciencia de Datos, que incorporará la materia de cómputo cuántico.

Como se mencionó al inicio, la Universidad Nacional es pionera en el ámbito de la computación en América Latina, y actualmente tiene un convenio de colaboración con IBM sobre programas específicos de cómputo cuántico, inteligencia artificial y supercómputo.

Agradecemos el apoyo de la Dirección General de Cómputo y Tecnologías de la Información y Comunicación y del Centro Virtual de Computación para la elaboración de este artículo.

Miztli, una gran supercomputadora, para una gran universidad cuyo arribo fue en 2013. Anualmente apoya 120 proyectos de investigación, tanto de la UNAM como de otras instituciones que solicitan recursos de supercómputo del Laboratorio Nacional de Cómputo de Alto Desempeño.

COMPUTACIÓN CONCURRENTE *a través del* razonamiento secuencial

Dr. Sergio Rajsbaum y Dr. Michel Raynal
Instituto de Matemáticas, UNAM - Universidad de Rennes, Francia

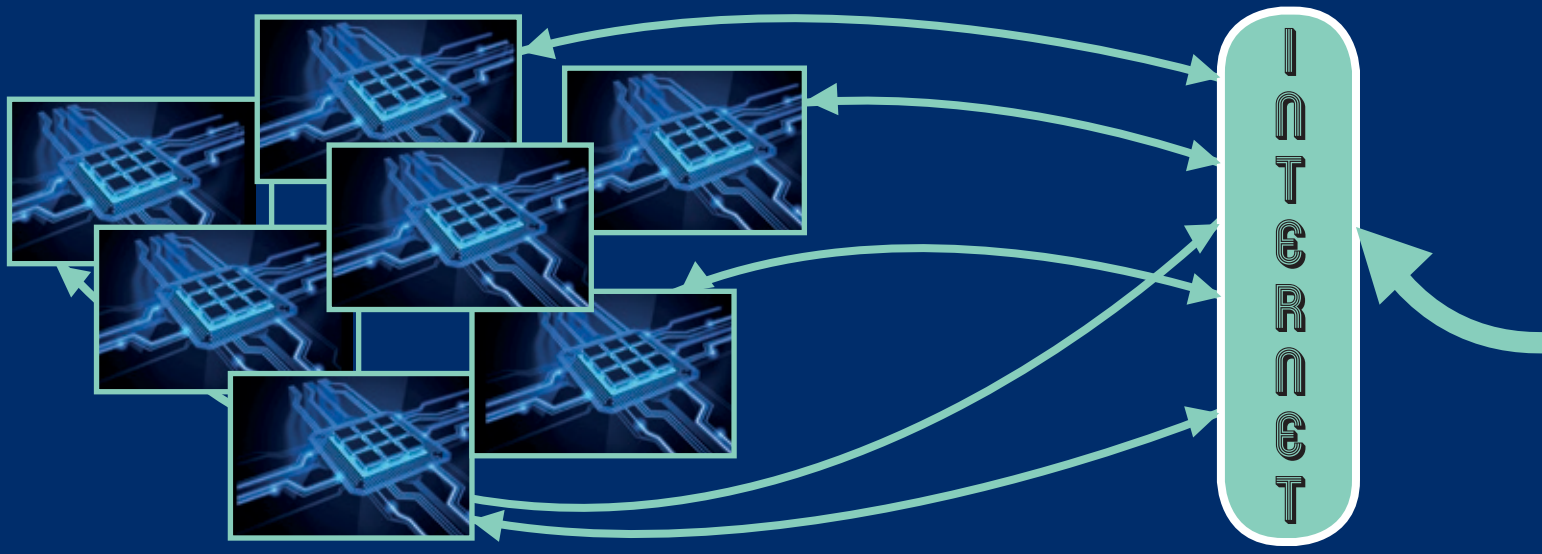
Los sistemas modernos de cómputo ejecutan muchas tareas al mismo tiempo. Es sumamente difícil programar estos sistemas de forma que toleren fallas, que sean eficientes y que escalen bien a más y más tareas. La forma de lograrlo ha evolucionado desde los años sesenta del siglo pasado hasta la fecha, a través de un paradigma que da la ilusión de un mundo secuencial, como si las tareas no se ejecutaran en paralelo.

El reto del cómputo concurrente

El cerebro humano realiza muchas tareas al mismo tiempo, de forma natural y constante. Sin embargo, solo después de un entrenamiento intenso podemos –como un músico o un deportista– estar al tanto y realizar varias actividades simultáneamente. Es infinitamente más fácil razonar y actuar secuencialmente, haciendo solo una cosa a la vez, quizá porque nuestro principal mecanismo de comunicación con los demás es el lenguaje hablado, intrínsecamente secuencial. Pero también porque hacer muchas cosas a la vez es más complejo. Por este motivo en computación también es mucho más fácil razonar sobre un programa secuencial, que sobre uno que consista de varias tareas ejecutadas simultáneamente. La programación concurrente es enormemente difícil porque tiene que hacer frente a muchos comportamientos, con frecuencia impredecibles, ya sea a nivel microscópico, como es el caso de una computadora multi-core con varios procesadores que operan comunicándose mediante una memoria compartida, o cuando las computadoras ubicadas en diferentes lugares interactúan enviándose mensajes a través de una red, para construir los sistemas web que usamos todos los días.

Hoy, el mayor desafío en la ciencia y la tecnología de la información (TI) reside en dominar la concurrencia. Los ingenieros de software están bien preparados para programar sistemas que ejecutan una instrucción tras otra. Los programas concurrentes en cambio son diseñados por especialistas, y aun así con frecuencia tienen errores o problemas de eficiencia. Ahora la concurrencia se está imponiendo a toda la comunidad de TI debido a dos fenómenos disruptivos: el desarrollo de redes de comunicaciones, y la capacidad de aumentar la velocidad de las computadoras a una tasa de crecimiento exponencial. Los aumentos en el rendimiento vienen de la disponibilidad de procesadores colaborando unos con otros, ya sea para incrementar la velocidad de cómputo o para construir sistemas distribuidos y tolerantes a fallas disponibles continuamente y a escala global.

Para ilustrar brevemente la dificultad de la programación concurrente, pensemos en la situación que tiene lugar cuando se programa una cuenta de banco, a la cual varios usuarios tienen acceso. Un usuario podría estar consultando el saldo, mientras otro hace un retiro, y otro más efectúa un depósito. Para programar un sistema como este, se requiere garantizar que las



operaciones concurrentes de los distintos usuarios no interfieran entre sí. En ocasiones es necesario realizar operaciones compuestas, como hacer un retiro solamente si el saldo es mayor a una cierta cantidad, para mantener un saldo mínimo. Si otro usuario hace un retiro al mismo tiempo, el primero podría erróneamente pensar que la cuenta mantiene el saldo mínimo.

Los albores del razonamiento secuencial en el cómputo concurrente

En los años sesenta del siglo pasado se hablaba ya de una crisis, ocasionada por la cantidad de problemas que había al intentar resolver situaciones como la ilustrada en el párrafo anterior. El nacimiento de una ciencia de concurrencia llega a finales de esa década con el paradigma de exclusión mutua, que hace que el sistema se encargue de darle la ilusión a los usuarios de que sus operaciones se ejecutan de forma atómica, una tras de otra, de manera que el estado de la cuenta siempre esté bien definido. En realidad, hay varias tareas concurrentes, que se sincronizan solicitando acceso exclusivo a los datos de la cuenta. Los usuarios nunca perciben el paralelismo.

Tolerancia a fallas y distribución

El enfoque de exclusión mutua tiene la limitante de que si una tarea falla mientras tiene el acceso exclusivo al recurso compartido (la cuenta de banco), puede bloquear todo el sistema, ya sea a nivel interno en un sistema multi-core, o en un sistema distribuido al hacer una reservación de boletos, al interactuar con programas de redes sociales, o llenar un formato por internet.

Por otro lado, el sistema debe encargarse de que no se pierdan los datos de la cuenta. La segunda etapa de desarrollo conceptual en la concurrencia define el paradigma de replicación de estado. El sistema mantiene varias réplicas de la cuenta en servidores en lugares diferentes. Esto tiene además la ventaja de que las operaciones de usuarios cercanos a un servidor pueden ser atendidos por este, y no tienen todos que enlazarse a un mismo lugar con los correspondientes retardos de comunicación y cuellos de botella que habría si todos los datos estuvieran en un solo lugar.

El diseño del sistema debe asegurar la consistencia de los datos replicados en varios servidores. Es necesario que cuando un usuario hace un depósito, este quede registrado en todos los servidores, y que todos apliquen las transacciones en el mismo orden. Para esto es necesario que los servidores se comuniquen unos con otros, y queden de acuerdo en el orden en que van a ejecutar transacciones que se estén solicitando concurrentemente. Se utilizan algoritmos de replicación de estado que toleran fallas, inclusive maliciosas, y se utilizan en los sistemas modernos de forma transparente a los usuarios.

Escalabilidad y auditabilidad - *blockchains*

La tercera etapa de la concurrencia nos lleva a un nivel de abstracción aún mayor, de estructuras de datos distribuidas, ya no simplemente replicadas. Se utilizan con éxito estructuras de datos clásicas como pilas, colas y árboles de forma colaborativa entre procesos concurrentes que maximizan el desempeño aprovechando los procesos rápidos y tolerando retardos o fallas de otros procesos.

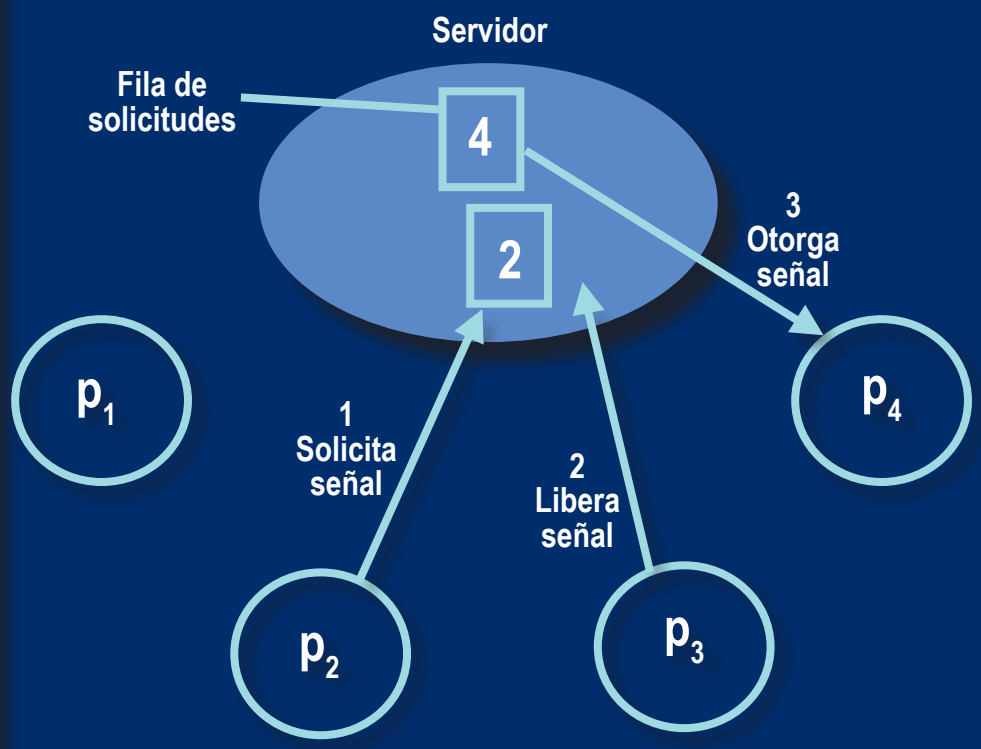
Surgen recientemente dos requerimientos adicionales que nos dirigen hacia el futuro de la ciencia de la concurrencia. El primero tiene que ver con transparencia, en situaciones donde se requiere poder auditar al sistema, en el sentido de que cualquier usuario pueda obtener una lista de las operaciones que se han ejecutado. Se pide que sea imposible pretender que una operación no tuvo lugar, o que se efectuó en otro orden en la secuencia de operaciones. Esto es precisamente lo que logra la estructura distribuida *blockchain*, de la que tanto se habla en las noticias. Mediante técnicas criptográficas, es posible añadirle a cada transacción un apuntador a la transacción prece-

El proceso P_1 no requiere acceso a la computadora del servidor (CS).

El proceso P_2 ha sido anexado a la fila, que ya contenía la solicitud de P_4.

El proceso P_3 sale de la CS.

El servidor elimina el acceso de P_4 y a su vez otorga permiso para que P_4 entre, respondiendo a su solicitud.



dente, que es prácticamente imposible de modificar, y que cualquier usuario la puede auditar.

El segundo requerimiento es la escalabilidad a cientos de miles de transacciones. Los sistemas tradicionales de replicación se vuelven demasiado lentos en este caso, ya que para que los servidores lleguen a un consenso del orden de las transacciones, es necesario que se comuniquen todos con todos, incurriendo en un costo que crece de forma cuadrática en la cantidad de servidores, y rápidamente se vuelve prohibitivo. Una forma en que sistemas tipo *blockchain* le dan la vuelta a este costo, es logrando un consenso simulando una especie de lotería que usa funciones criptográficas *hash*, en el cual, con muy baja probabilidad, sale el boleto premiado, y muy raramente más de uno saca el premio. Para participar en la lotería, el servidor tiene que haber hecho muchas operaciones con el consecuente gasto enorme de energía. El servidor que saca el premio se convierte en el líder, que decide el orden a ejecutar de la siguiente transacción (y obtiene un pago correspondiente, que lo motiva a participar en la lotería).

La reducción de concurrencia a razonamiento secuencial persiste, pasando de exclusión mutua a replicación, luego a distribución y finalmente a escalabilidad auditable global. Las transacciones se emiten concurrentemente pero se ordenan todas en el *blockchain* para darles una semántica secuencial. Entre los muchos retos de la programación concurrente se encuentra el de inventar sistemas con los beneficios de *blockchains*, pero sin el gasto de energía prohibitivo. Otros retos incluyen cómo programar soluciones a problemas inherentemente concurrentes (que no tienen especificaciones secuenciales), diseñar mejores técnicas de programación concurrente, y en general continuar desarrollando la ciencia de concurrencia.

La ciencia del cómputo concurrente

En la UNAM hemos estudiado estos y otros problemas de concurrencia desde una perspectiva científica desde hace más de 20 años, contribuyendo a entender las posibilidades y limitaciones de los sistemas concurrentes. Somos un grupo líder, en especial en los fundamentos para estos sistemas, usando la rama profunda de las matemáticas, topología. Hemos escrito el libro *Distributed Computing Through Combinatorial Topolgy* (Elsevier), además de cientos de artículos; asimismo, hemos colaborado en diversos proyectos de investigación con Estados Unidos, Francia e Israel. También hemos organizado múltiples eventos y formado recursos humanos a lo largo de todos estos años.

Referencias

Maurice Herlihy, Dmitry N. Kozlov, Sergio Rajsbaum: *Distributed Computing Through Combinatorial Topology*. Elsevier-Morgan Kaufmann, 2013.

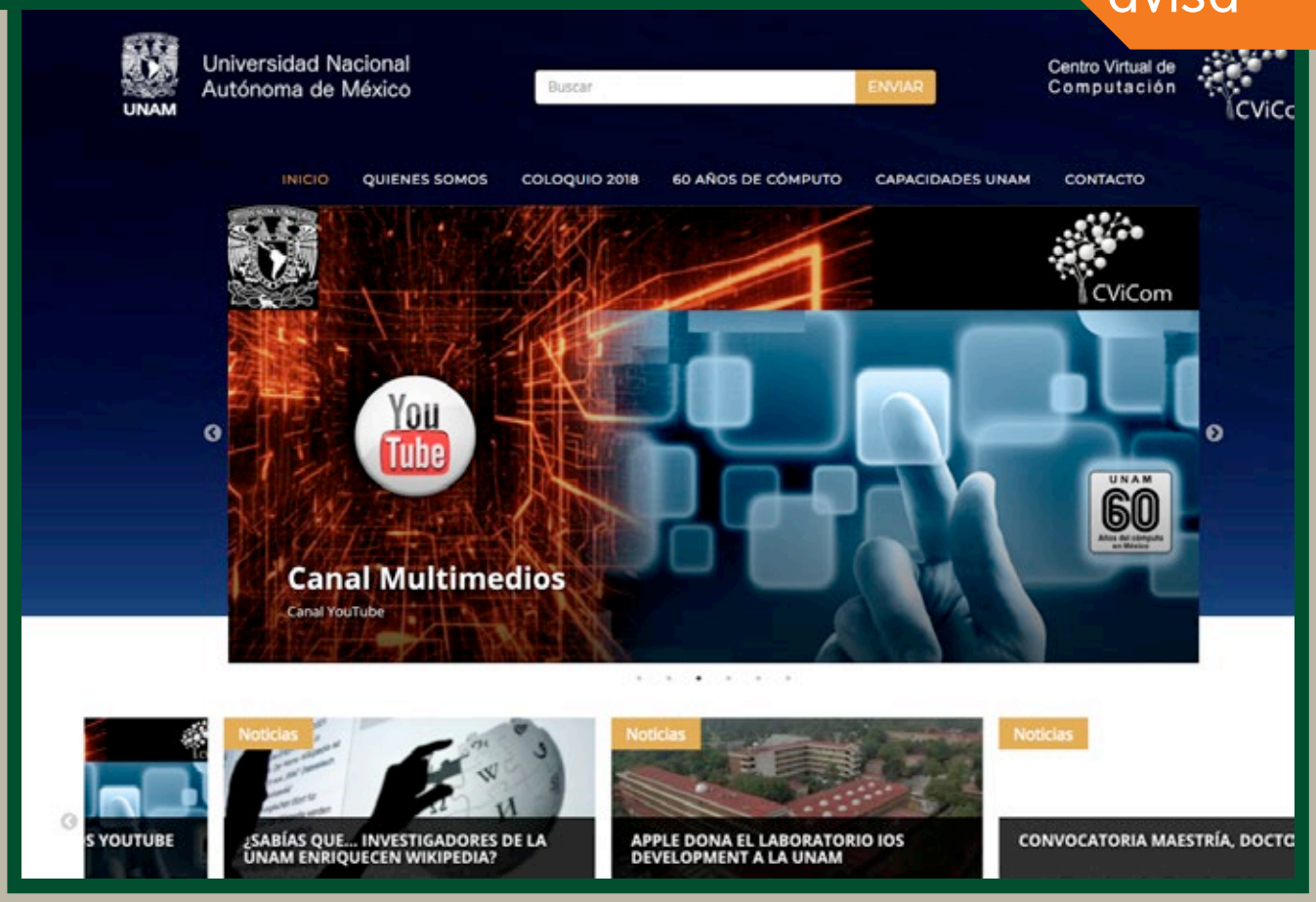
Ming-Yang Kao: *Encyclopedia of Algorithms — Distributed Algorithms*, S. Rajsbaum (Ed.). Springer, 2016.

Michel Raynal. *Distributed Algorithms for Message-Passing Systems*. Springer, 1-500, 2013.

Agradecimientos a PAPIIT IN109917 y Francia INRIA LiDiCo.

El faro avisa

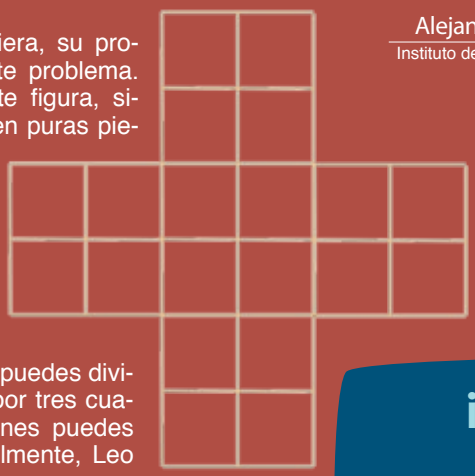
https://cvicom.unam.mx/



A VER SÍ PUEDES

Alejandro Illanes Mejía
Instituto de Matemáticas, UNAM

Para que Leo no se aburriera, su profesora le puso el siguiente problema. Supón que divides la siguiente figura, siguiendo las líneas dibujadas, en puras piezas iguales. ¿Cuántos tipos de piezas puedes conseguir? Por ejemplo, puedes obtener cuadritos de uno por uno (y ese es un tipo de pieza); también puedes dividirla obteniendo rectángulos de dos por uno (y ese es otro tipo de pieza que puedes obtener), no puedes dividirla y tener piezas formadas por tres cuadritos, etcétera. Cuando termines puedes salir al descanso, le dijo. Finalmente, Leo no salió. ¿Tú sí saldrías?



¡Gánate un libro!

Síguenos en Facebook.
Busca en nuestra fanpage la publicación del acertijo, y sé uno de los primeros en dar la respuesta correcta, y además compártela en tu muro.

Boletín El faro UNAM

UNAM 60 Años del cómputo en México

CViCom Centro Virtual de Computación

En el marco de la celebración de los 60 años del Cómputo en México se invita a la comunidad universitaria a asistir al

PRIMER COLOQUIO

del Centro Virtual de Computación

Espacio de encuentro
para académicos que realizan
investigación en Computación
y sus Aplicaciones

Inteligencia Artificial
Nuevos Materiales
Medicina y Salud
Sociedad y Educación
Nuevas Tecnologías
Ambiente y Contaminación

Mesas redondas
Conferencias magistrales
Ponencias

8, 9, 10 Agosto

Auditorio del IIMAS

https://cvicom.unam.mx/coloquio2018/