



Computación en equipo

Sergio Rajsbaum



Los sistemas de cómputo modernos llegan a constar de miles de computadoras individuales que se comunican y colaboran para realizar alguna tarea. Inicialmente se creía que mientras más computadoras se repartieran el trabajo, más rápido podrían resolver una tarea, pero que sólo podían solucionar el mismo tipo de tareas que una computadora haciendo una sola cosa a la vez. Ahora se sabe que la naturaleza del cómputo distribuido es muy diferente a la del secuencial, debido a la posibilidad de fallas en sus componentes y a imperfecciones en la comunicación entre éstos.

Introducción

¿Cuál es el límite de lo que las computadoras pueden hacer? Ya tenemos computadoras que resuelven problemas matemáticos, que permiten construir grandes edificios y puentes, que juegan ajedrez al nivel de los mejores jugadores del mundo, que pilotan los aviones comerciales en los que volamos, y que producen música hermosa (el programa de David Cope, de la Universidad de California Santa Cruz, por ejemplo, compone al estilo de Bach y de Mozart: <http://artsites.ucsc.edu/faculty/cope/>).

Esto no nos debe sorprender: ultimadamente, las computadoras no son más que una herramienta para extender nuestras mentes, al igual que un telescopio extiende nuestros ojos y unas pinzas la fuerza de nuestras manos. Sin embargo, el descubrimiento en 1936 del joven Alan Turing, de 24 años, nos indica que, de hecho, la mayoría de los problemas no pueden ser resueltos por una computadora. Hay muchos problemas que las computadoras no pueden solucionar simplemente porque hay más problemas que *programas* para resolverlos.

Una computadora no hace más que seguir las instrucciones del programa que está ejecutando, que representa los pasos que hay que seguir para resolver el problema; sin embargo, el que la computadora “sólo” siga instrucciones de un programa no debe ser subestimado pues el programa puede ser enormemente complejo.



Lo que puede o no hacer una computadora *secuencial* (es decir, que hace una sola cosa a la vez) es tema de otros artículos de esta revista. Aquí nos preguntaremos acerca de lo que sí pueden hacer las computadoras cuando trabajan en equipo.

● **Sistemas de cómputo distribuido**

Si una computadora no es más que una herramienta para extender la mente humana, un conjunto de computadoras es una herramienta para extender la capacidad de una sociedad humana.

En una sociedad cada vez más global e interconectada, la capacidad de generación y procesamiento de información se ha incrementado a una velocidad enorme; hoy no alcanzamos siquiera a imaginar sus alcances. Esto ha venido de la mano con el desarrollo de redes de computadoras. El mundo entero está cubierto por millones de computadoras que se comunican, colaboran y resuelven problemas cada vez más sofisticados. Los sistemas de computadoras permean nuestras actividades cotidianas. En realidad, no es una computadora la que maneja el avión comercial, sino muchas; algunas en el avión y otras fuera de éste, que ayudan a la navegación y la coordinación con otros aviones. Los sistemas de control de vuelos en sí, en las torres de control de los aeropuertos, son sumamente complejos y se comunican unos con otros para tomar decisiones. Interactúan con el *sistema de posicionamiento global* (GPS, por sus siglas en inglés) que permite obtener la locali-



zación de un objeto que porta un receptor GPS con una precisión de unos metros (véase Figura 1).

Utilizamos cotidianamente sistemas como los que tienen los cajeros automáticos, que permiten al cliente de un banco obtener dinero en casi cualquier parte del mundo. Para lograrlo, existen computadoras que tienen la información de la cuenta de banco del cliente, con varias copias como respaldo, que se comunican con la computadora del cajero automático, y entre ellas, para mantener las copias con el mismo respaldo.

● **El poder de los sistemas de computadoras**

El resultado de Turing, que mostraba que no importa qué tan rápidas y poderosas sean las computadoras actuales o que puedan construirse en el futuro, pues pueden resolver solamente una pequeña fracción de todos los problemas, sorprendió a todos.

Las implicaciones de este resultado se han estudiado por muchos años, y no únicamente en ciencias de la computación: también en filosofía y otras áreas. Con el advenimiento de Internet y el desarrollo de los sistemas de computadoras, ha habido cada vez más interés por entender cuáles son los límites de lo que puede hacer no una, sino muchas computadoras cuando se comunican y trabajan en equipo. En el caso de una sola computadora, Turing descubrió, como ya dijimos, que la razón es simple: hay más problemas que soluciones. En 1993 se anunció en el Congreso Anual de Teoría de

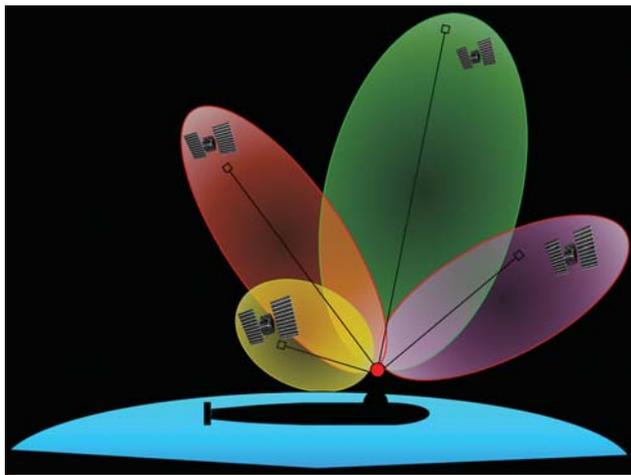


Figura 1. Sistema de posicionamiento global (GPS).

la Computación (STOC) de la ACM (*Association for Computing Machinery*, la sociedad académica de ciencias de la computación más importante del mundo) el descubrimiento de otro fenómeno extraño: en los sistemas de computadoras los aspectos de *computabilidad* adquieren una faceta distinta. Aquí también hay muchos problemas que no son computables; no obstante, los límites a lo computable reflejan la dificultad de tomar decisiones frente a la incertidumbre de lo que está sucediendo en otras partes del sistema.

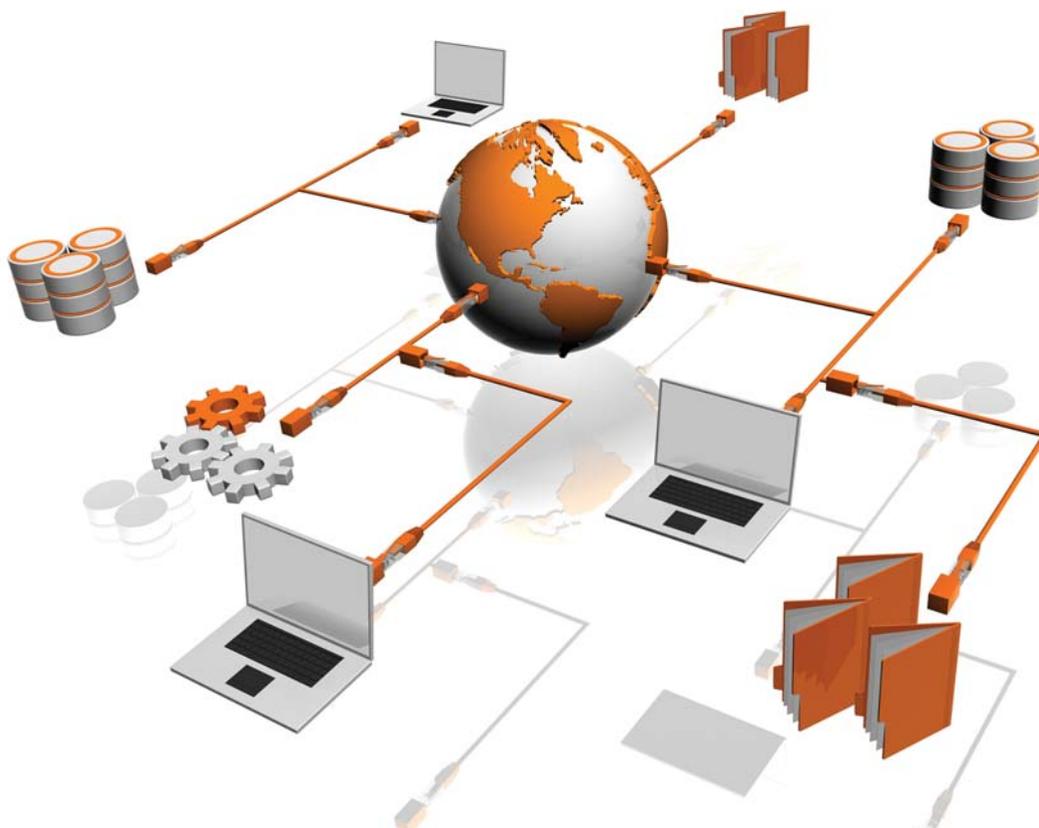
Una computadora del sistema no puede saber con exactitud lo que está sucediendo en otras computadoras en un momento dado; por ejemplo, si un mensaje ya fue recibido o si una respuesta ya fue enviada. En realidad, lo que puede hacer un sistema de computadoras tiene relativamente poco que ver con la capacidad computacional inherente de cada una de las computadoras participantes. Las limitaciones del sistema tienen que ver con que sus componentes pueden fallar, con lo impredecible que puede ser el tiempo que toman los mensajes en ser transmitidos, y la velocidad a la que una computadora puede responderlos.

En todo sistema, las fallas son inevitables y la comunicación es imperfecta. En el congreso STOC de 1993 se anunció un hecho insólito: los argumentos de imposibilidad de sistemas de cómputo son de carácter geométrico. Dos grandes áreas de conocimiento se entrelazan: las ciencias de la computación y la *topología*, una rama de las matemáticas muy desarrollada y abstracta que estudia propiedades geométricas de objetos elásticos.

● **Comunicación distribuida**

Para que exista cómputo distribuido es necesario resolver los problemas de comunicación básicos: lograr que las computadoras se comuniquen lo más confiablemente posible a través de señales de radio, de líneas de comunicación, y la construcción de la infraestructura de comunicaciones correspondiente.

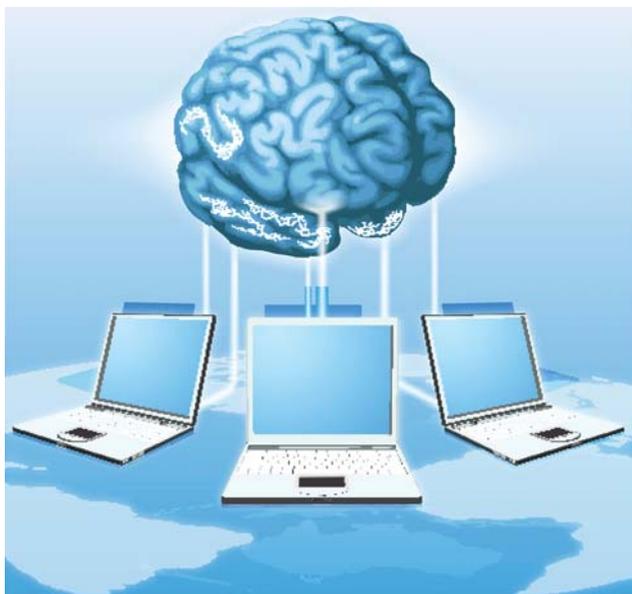
La colocación del primer cable submarino trasatlántico tomó años y representó un enorme esfuerzo económico y de ingeniería. Fue una aventura fascinante. El telégrafo de Samuel Morse había permitido comunica-



ción casi instantánea en Europa y en Norteamérica para mediados del siglo XIX. Por fin existía un sistema que permitía que el ser humano se comunicara más rápidamente de lo había sido posible durante milenios, a pie o a caballo, y reducir el tiempo de entrega de un mensaje de días a segundos: ¡100 mil veces más rápido! Sin embargo, la comunicación entre el viejo y el nuevo continente seguía siendo a través de cartas llevadas por barcos que tomaban semanas en cruzar el Atlántico, y a veces nunca llegaban.

Se había vuelto imprescindible la colocación de un cable submarino que conectara a los dos continentes, pero tomó 12 años conquistar los enormes retos de instalar un cable de 2 500 kilómetros de largo a una profundidad de más de 4 kilómetros. Recorrer el camino que nos ha traído hasta la situación actual es un gran logro de la humanidad. Ahora, además de muchos cables submarinos y terrestres por todo el mundo, existen sistemas de radio y satélites, que permiten comunicarse en fracciones de segundos, con muy alta confiabilidad, a cualquier punto de la Tierra.

La comunicación global es solamente una parte de la historia que es compartida a nivel microscópico por los sistemas distribuidos, en especial del tipo *multicore*, donde un solo chip tiene varias computadoras o *cores* que trabajan en paralelo. Inclusive cualquier teléfono celular moderno tiene un procesador *multicore* que incluye muchos componentes que se comunican entre sí,



no únicamente mediante canales diminutos, sino también mediante sistemas de memoria compartida. Un *core* se puede comunicar con otro, depositando datos en la memoria que otro procesador puede leer. En años recientes los fabricantes de computadoras se han topado con que ya no las pueden hacer más veloces cada año, por lo que ahora la tendencia es construirlas con más y más elementos que trabajen en paralelo.

● Problemas distribuidos

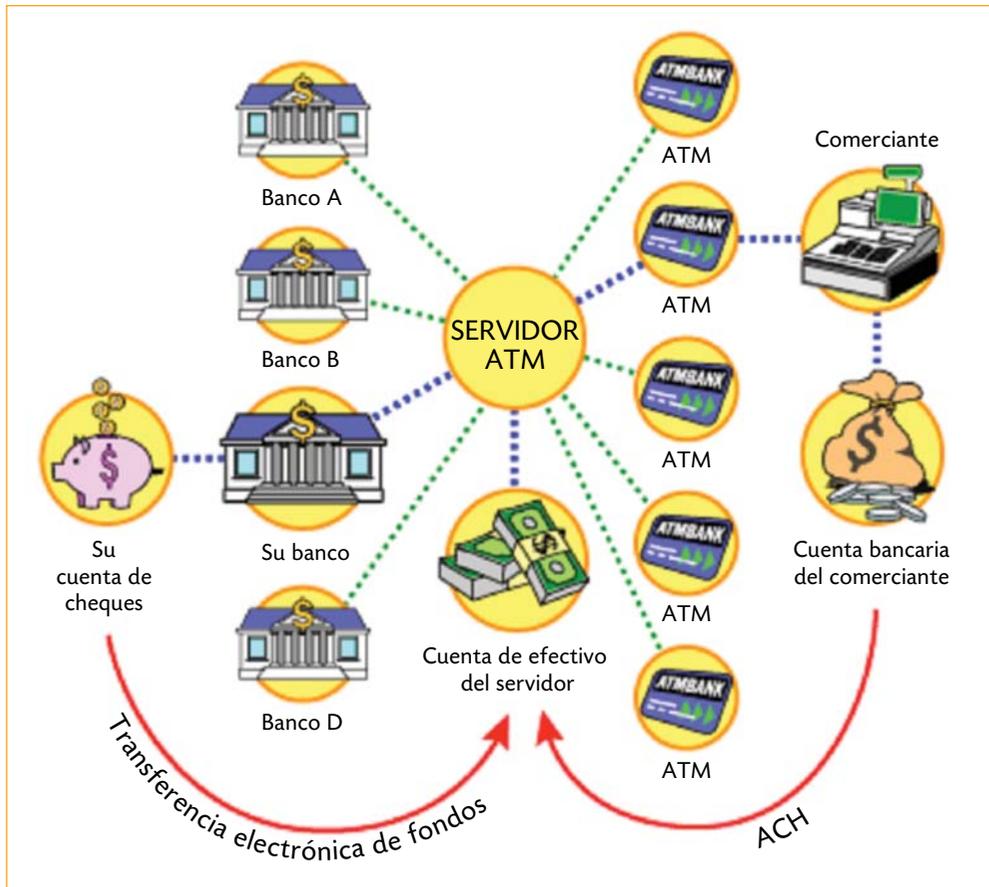
Existe una gran diversidad de *problemas distribuidos* que deben ser resueltos por un sistema en el cual varias computadoras o microprocesadores se comunican entre sí. Quizás los más importantes son los que tienen que ver con problemas de *acuerdo*. Regresemos al sistema GPS de la Figura 1. Hay al menos 24 satélites siempre en operación que transmiten señales con su localización, repetidamente y al mismo tiempo. Para poder enviar las señales lo más simultáneamente posible, cada satélite tiene un reloj atómico, de muy alta precisión.

Hoy hay receptores GPS de precio módico que nos permiten determinar nuestra localización. Un receptor GPS en Tierra recibe estas señales del sistema de satélites. Calcula la distancia a un satélite por el tiempo que tomó en llegarle la señal, ya que ésta contiene la localización del satélite y la hora en la que fue enviada. Esto permite al receptor calcular una esfera alrededor del satélite, en la cual está localizado.

A menos de que algún edificio obstruya la vista al cielo, el receptor GPS recibirá señales de al menos cuatro satélites, y puede calcular las cuatro esferas correspondientes, cuya intersección es la localización del receptor. La precisión de esta localización depende de qué tan bien sincronizados estén los relojes de los satélites. Es por esto que deben resolver entre ellos el problema de quedar de acuerdo con la hora, conocido como *sincronización de relojes*.

● La geometría de los problemas distribuidos

Cada vez que usamos un cajero automático para sacar dinero, el sistema bancario tiene que resolver un



problema de acuerdo. Un ejemplo sencillo nos permitirá comprender cuál es la conexión con la geometría, y más concretamente con la topología.

Supongamos que Alicia está en Acapulco y le indica al cajero que quiere sacar diez pesos de su cuenta. El cajero le envía un mensaje a la central, que está en Monterrey, preguntando el saldo de su cuenta. Comienza un intercambio de mensajes con el fin de resolver el siguiente problema de acuerdo: 1) el cajero entrega los diez pesos a Alicia y el dinero se descuenta de su cuenta, o 2) ninguno de estos actos se realiza. Pero se debe evitar a toda costa entregarle el dinero y no descontárselo de su cuenta, o descontárselo sin entregarlo. El problema se complica cuando existe la posibilidad de fallas y de problemas de comunicación.

Supongamos que usualmente cada mensaje toma un segundo en llegar a su destino. En la Figura 2 se representan algunos de los “mundos posibles”; es decir, distintas situaciones que podrían suceder. Al inicio se representan dos mundos: uno en el que el cajero decide

no entregar el dinero a Alicia (quizá porque Alicia canceló la operación o porque el billete se atoró), y otro en el que el cajero decide que todo está en orden y quiere entregar el dinero. El primer mundo se representa como la arista a , que conecta dos vértices. El vértice negro de la izquierda corresponde al cajero, en su intención de no dar el dinero, y el vértice negro de la derecha representa la intención de sí darlo. El vértice blanco está contenido en la arista a y también en la b , indicando que, para el banco, ambos mundos son posibles, mientras no reciba un mensaje del cajero indicando otra cosa. Estas dos aristas con sus tres vértices forman la *gráfica inicial* de mundos posibles. El cajero envía un mensaje al banco avisando de su intención.

La gráfica del segundo renglón presenta los cuatro mundos posibles, c , d , e , f , un segundo después. Las aristas c y d corresponden al mensaje con intención negativa, la c para el caso en que el mensaje llega y la d para el caso en que no llega. Ahora es el cajero el que no sabe en este momento si el mensaje llegó o no, y

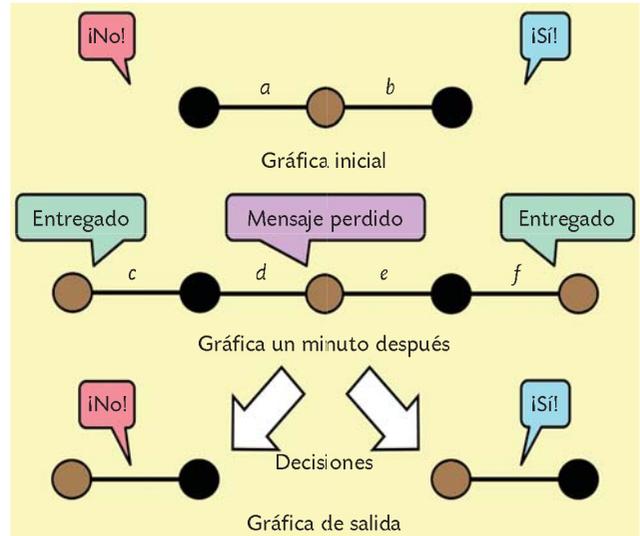


Figura 2. Sistema de cajeros automáticos (ATM).

por eso el vértice negro está tanto en la arista *c* como en la *d*. El vértice blanco central representa la situación en que el mensaje no le llega al banco, y por lo tanto está contenido en las aristas *d* y *e*. Es decir, el banco no sabe si el mensaje que se perdió decía que sí o que no se va a entregar el dinero. Supongamos que en este momento hay que tomar la decisión. Cada uno de los vértices de la segunda gráfica corresponde a lo que sabe un participante en ese momento, y con esa información debe tomar una decisión.

La tercera gráfica consiste en dos aristas disconexas: una en la que ambos deciden “no” y la otra en la que ambos deciden “sí”. En cambio, la segunda gráfica es conexa: consiste en un camino de cuatro aristas. Si colocamos la decisión encima del vértice correspondiente en este camino, cada uno de los vértices de la segunda gráfica estaría etiquetado con “sí” o “no”. El extremo izquierdo debería estar etiquetado con “no”, ya que en este caso no se perdió el mensaje, y el extremo derecho con “sí”, por la misma razón. No importa cómo etiquetemos los vértices internos, va a haber una arista cuyos vértices estén etiquetados con valores distintos, ya que el camino es conexo. Esta arista con dos etiquetas diferentes corresponde a un mundo posible en el que el cajero y el banco toman decisiones contradictorias, ocasionando que el sistema falle.

La falla descrita implica que enviando un solo mensaje, el problema de acuerdo no tiene solución. Apa-

rentemente es necesario que el protocolo continúe, y el banco responda con un mensaje al cajero en el momento de recibir un mensaje. El lector puede verificar que, si el protocolo continúa, se generaría una tercera gráfica de mundos posibles que consistiría en un camino con seis aristas. Como éste sigue siendo conexo, tampoco permite resolver el problema del acuerdo.

Sorprendentemente, no importa cuántas rondas de mensajes intercambien el cajero y el banco, la gráfica final sigue siendo conexa y, por tanto, el problema de acuerdo no se puede resolver. ¡Los errores en el sistema de cajeros automáticos son inevitables! Lo más que puede lograrse es que ocurran con poca frecuencia. Una propiedad topológica, la *conexidad* de la gráfica, es la que evita que el problema tenga solución.

Conclusiones

Se ha repetido una y otra vez que la computación ha sido a tal grado revolucionaria que es prácticamente imposible hacer investigación en ciencias e ingenierías sin la habilidad de pensar computacionalmente. En especial, pensar *concurrentemente* ha tomado una importancia crítica, en parte por el advenimiento de la Internet y los sistemas paralelos *multicore*.

Investigadores como Carl Hewitt, Jan van Leeuwen, Jirí Wiedermann, Leonid Levin y otros han estudiado modelos que intentan capturar aspectos del

cómputo concurrente, probabilista (véase el artículo de H. Zenil en este mismo número de *Ciencia*) y que continúa indefinidamente. Estos modelos pretenden reflejar lo que ocurre en Internet y otros grandes sistemas distribuidos (que pueden inclusive incluir seres humanos), y que no son capturados por modelos clásicos secuenciales como la máquina de Turing.

En buena parte, los aspectos que tienen que ver con la tolerancia a fallas y el no determinismo le dan al cómputo distribuido un carácter particular. En efecto: los resultados clásicos de computabilidad nos dicen que un modelo de cómputo con varias computadoras, sin fallas, puede resolver problemas más rápidamente que uno secuencial, pero no puede resolver más problemas. Desde los tiempos de Turing se han considerado distintos modelos paralelos sin fallas, como las máquinas de Turing de varias cintas, máquinas paralelas PRAM, autómatas celulares, etcétera, todos los cuales pueden ser simulados por una máquina de Turing (aunque muy lentamente). Por tanto, todos estos modelos tienen el mismo poder de computabilidad.

No obstante, en cualquier modelo realista de computación distribuida, cada participante (cibernético o humano) conoce al inicio sólo una parte del estado global del sistema; las incertidumbres ocasionadas por fallas, así como la incapacidad de predecir la velocidad de procesamiento y comunicación de los componentes del sistema limitan a cada participante, proporcionándoles una imagen incompleta del estado global del sistema, que continúa más allá del inicio del cómputo. Las fallas pueden inclusive ser malignas cuando un agente extraño ha logrado tomar control de algunas computadoras o canales de comunicación del sistema, lo cual hace aún más difícil la construcción de sistemas distribuidos eficientes, correctos y seguros.

Sergio Rajsbaum es investigador del Instituto de Matemáticas de la Universidad Nacional Autónoma de México (UNAM) desde 1991, y miembro del Sistema Nacional de Investigadores. Cursó el doctorado en Ciencias de la Computación del Technion en Israel, e hizo una estancia posdoctoral en el Instituto Tecnológico de Massachusetts (MIT), en Estados Unidos. Su área principal de investigación es la teoría de la computación, especialmente el cómputo distribuido tolerante a fallas, en la cual ha publicado más de 100 artículos de investigación. Ha sido investigador visitante en los laboratorios de investigación de Hewlett Packard, de IBM, en la Universidad de París, la Universidad de Rennes, la Universidad de Ottawa y la Universidad de California en Los Ángeles, entre otros. rajsbaum@unam.mx

Lecturas recomendadas

- Gralla, Preston (2006), *How the Internet works*, 8a. edición, Que Publishing.
- Herlihy, Maurice y Nir Shavit (2008), *The art of multiprocessor programming*, Morgan Kaufmann.
- Hewitt, Carl (2012), "What is computation? Actor model versus Turing's model", en Héctor Zenil, *A computable universe*, World Scientific. Disponible en <<http://what-is-computation.carlhewitt.info>>.
- Rajsbaum, Sergio (coordinador, 2009), *Conocimientos fundamentales de computación*, UNAM (disponible también dentro de la *Enciclopedia de conocimientos fundamentales UNAM-Siglo XXI*, 1a. edición 2010, Siglo XXI /Universidad Nacional Autónoma de México).
- Ryan, Johnny (2011), *A history of the Internet and the digital future*, Reaktion Books.
- Standage, Tom (2007), *The victorian Internet: the remarkable story of the telegraph and the nineteenth century's online pioneers*, Walker and Company.
- Steele Gordon, John (2003), *A thread across the ocean: the heroic story of the transatlantic cable*, Harper. Disponible en <<http://www.pbs.org/wgbh/amex/cable/index.html>>.