

¿Es la Ciencia de la Computación Ciencia?

Ésta es la traducción del artículo de [Peter J. Denning](#) titulado “[Is Computer Science Science?](#)”, publicado en la revista “[Communications of the ACM](#)” en Abril de 2005.

(Traducción por: [Marco A. Alvarez](#))

- *¿Cuál es tu profesión?*

Ciencia de la Computación.

- *Oh. ¿Es una ciencia?*

Seguro, es la ciencia del procesamiento de la información y su interacción con el mundo.

- *Yo puedo aceptar que lo que haces es tecnología, pero no ciencia. Ciencia está relacionada con las leyes fundamentales de la naturaleza. Computadoras son hechas por el hombre. Sus principios vienen de otras áreas como la física y la ingeniería electrónica.*

Espera. Hay mucho procesamiento de información natural. Computadoras son herramientas para implementar, estudiar y predecirlos. En los Estados Unidos solamente hay cerca de 200 departamentos académicos en las universidades que así lo reconocen, algunos vienen otorgando grados de Ciencia de la Computación desde hace 40 años.

- *Ellos hacen parte de una ilusión en masa. Los pioneros de tu campo creían por 1950 que su nueva área era ciencia. Ellos estaban equivocados. No hay ciencia de la computación. Computación como arte, si. Computación como tecnología, si. Pero no ciencia. El término moderno Tecnología de la Información es más cercano a la realidad.*

Yo no acepto tus afirmaciones sobre mi área y mi grado. ¿Te importas si vemos más detalles? Vamos a examinar los criterios aceptados para definir una ciencia y ver si la computación los cumple.

- *Estoy escuchando.*

Entendimiento común sobre lo que es ciencia

Nuestra área fue denominada Ciencia de la Computación desde inicios de la década de 50. Luego durante las siguientes 4 décadas, nosotros acumulamos un conjunto de principios que se extienden además de sus fundamentos matemáticos originales para incluir ciencia computacional, sistemas, ingeniería y diseño. El reporte de 1989, “Computación como Disciplina” definió el área como: “La disciplina de Computación es el estudio sistemático de procesos algorítmicos que describen y transforman información: su teoría, análisis, diseño, eficiencia, implementación y aplicación. La pregunta fundamental subyacente a toda la computación es ¿Que es lo que puede ser (eficientemente) automatizado?” [3].

Ciencia, ingeniería y matemática se combinan en una única y potente mezcla en nuestro campo. Algunas de nuestras actividades son primariamente ciencia – por ejemplo, algoritmos experimentales, ciencia de la computación experimental y ciencia computacional. Algunas son primariamente ingeniería – por ejemplo, diseño, desarrollo, ingeniería de software e ingeniería de computación.

Algunas son primariamente matemáticas – por ejemplo, complejidad de algoritmos, software matemático y análisis numérico. Pero muchas son combinaciones. Todos los 3 conjuntos de actividades son trazados sobre los mismos principios fundamentales. En 1989, nosotros utilizamos el término “Computación” en vez de “ciencia de la computación, matemática e ingeniería”. Hoy, computación, ingeniería, matemática, arte y todas sus combinaciones son agrupadas bajo el nombre “ciencia de la computación”.

El paradigma científico que se remonta a Francis Bacon, es el proceso de formar hipótesis y verificarlas a través de experimentos; hipótesis con éxito son vistas como modelos que explican y predicen fenómenos en el mundo. La Computación sigue este paradigma al estudiar el procesamiento de información. El sinónimo Europeo para ciencia de la computación – informática – sugiere de manera más clara que el área es sobre procesamiento de información, no computadoras.

Especialistas en lexicografía ofrecen dos distinciones adicionales. una es entre ciencia pura (básica) y ciencia aplicada; ciencia pura se centra en el conocimiento para su propio desarrollo y ciencia aplicada se centra en el conocimiento de utilidad demostrable. La otra es entre ciencia inexacta (cualitativa) y ciencia exacta (cuantitativa); ciencia exacta trabaja con predicción y verificación por observación, medición y experimentación.

La investigación en computación es abundante en ejemplos del paradigma científico. Investigadores en cognición por ejemplo hipotetizan que mucho del comportamiento inteligente es el resultado de procesamiento de información en el cerebro y el sistema nervioso; ellos construyen sistemas que implementan procesos de información hipotéticos y los comparan con el mundo real. Computadoras en esos estudios son herramientas para verificar las hipótesis; sistemas de éxito pueden ser utilizados inmediatamente. Investigadores en ingeniería de software hipotetizan modelos de cómo es hecha la programación y cómo surgen los defectos; a través de testes ellos procuran entender cuáles modelos trabajan bien y cómo usarlos para crear mejores programas con menos defectos. Investigadores en algoritmos experimentales estudian el desempeño de algoritmos reales en *datasets* reales y formulan modelos para predecir sus requerimientos de tiempo y almacenamiento; ellos pueden un día producir una teoría más precisa que el *Big-O* e incluir una teoría de localidad. El naciente campo de Interacción Hombre Máquina (*Human Computer Interaction*) está examinando las formas en las cuales humanos procesan información e interactúan con procesos automatizados.

Bajo esas definiciones, la computación califica como una ciencia exacta. Ella estudia el procesamiento de información en procesos que ocurren naturalmente en el mundo físico; científicos de la computación trabajan con un aceptado y sistemático cuerpo de conocimiento; mucho de la ciencia de la computación es ciencia aplicada; y la ciencia de la computación es utilizada para predicción y verificación.

La objeción de que la computación no es una ciencia porque estudia objetos hechos por el hombre (tecnologías) es una falacia. La ciencia de la computación estudia procesos de información artificiales y naturales. También ayuda a otras áreas a estudiar sus procesos. Físicos explican el comportamiento de partículas con procesos de información cuánticos – algunos de los cuales, como el enredo cuántico son bastante extraños – y verifican sus teorías con experimentos simulados en computador. Bioinformáticos explican el DNA como información biológica codificada y estudian cómo las enzimas de transcripción lo leen y actúan en él; modelos computacionales de esos procesos ayudan a diseñar terapias específicas a pacientes individuales. Laboratorios farmacéuticos y de materiales crean moléculas hechas por el hombre a través de simulaciones en computador de procesos subyacentes a composiciones químicas.

Para ayudar a definir las fronteras de la ciencia, especialistas en lexicografía también contrastan ciencia con arte. Arte se refiere a las prácticas útiles de un área, no a las pinturas o esculturas. Programación, diseño, ingeniería de *software* y *hardware*, construcción y validación de modelos y la construcción de interfaces de usuario son todos ‘arte’. Si la estética es adicionada, el arte en computación se extiende a

gráficos, *layout*, dibujos, fotografía, animación, música, juegos y entretenimiento. Todo este arte computacional complementa y enriquece la ciencia.

Ciencia en acción

En su notable libro acerca de los trabajos científicos, *Ciencia en Acción*, el filósofo Bruno Latour trae una nota de cautela para la distinción entre ciencia y arte [7]. Todo lo discutido en ésta columna (un sistematizado cuerpo de conocimiento, habilidad para hacer predicciones, validación de modelos), hace parte de lo que él llama ‘*ready-made-science*’, ciencia que está lista para ser usada y aplicada, ciencia que está lista para dar soporte al arte. Mucha ‘*science-in-the-making*’ aparece como arte hasta que se decida llamarla ciencia.

Latour define ‘*science-in-the-making*’ como el proceso por el cual hechos científicos son propuestos, argumentados y aceptados. Una nueva proposición es argumentada y estudiada en publicaciones, conferencias, cartas, correos electrónicos, discusiones, debates, prácticas y experimentos repetidos. Se torna un ‘hecho’ sólo después que gane muchos aliados entre científicos y otros que la usen. Para ganar aliados, una proposición debe ser verificada independientemente por múltiples observaciones y no deben existir contra-ejemplos. Latour tiene a ‘*science-in-the-making*’ como un turbio, político, proceso humano cargado de emociones y polémicas ocasionales. La literatura científica lo ratifica.

Todo lo que Latour dice es consistente con la definición (honrada por el tiempo) del paradigma científico. Luego de suficiente tiempo y validación, un modelo se hace parte del cuerpo de conocimiento científico.

Desacuerdo interno

No todos los científicos de la computación están de acuerdo con que la ciencia de la computación sea una ciencia. Su parecer en esta cuestión parece depender de la tradición en la cual crecieron. Hal Abelson y Gerry Sussman, identificados con las tradiciones matemáticas y de ingeniería en computación, dijeron, “Ciencia de la computación no es una ciencia, y su significado final tiene poco que ver con computadores” [1]. Ellos creen que el significado final está en las notaciones para expresar cómputos. Edsger Dijkstra, un matemático que construyó excelentes algoritmos, frecuentemente argumentó el mismo punto, aunque él también creyó que la computación es una ciencia matemática. Walter Tichy, un experimentalista y experto constructor de software, argumenta que ciencia de la computación es ciencia [12]. David Parnas, un ingeniero, argumenta que la parte de software de la ciencia de la computación es verdaderamente ingeniería [10]. Yo personalmente tengo trabajado en todas las 3 tradiciones de nuestro campo y no veo diferencias marcadas.

Hasta el “*Computer Science and Technology Board of the National Research Council*” no es consistente. En 1994, un panel argumentó que computación experimental es un aspecto esencial del área [9]. En 2004, otro panel discutió los logros de la investigación en ciencia de la computación; aparte de los comentarios sobre abstracción en modelos, ellos apenas dijeron una palabra acerca de la tradición experimental [8].

Paul Graham, un prominente miembro de la generación que creció con computadores, inventó *Yahoo! Store* y tempranas técnicas para filtros de *spam*; se identifica con arte en computación. Él dice: “nunca me gustó el término ciencia de la computación ... ciencia de la computación es una bolsa de sorpresas de áreas relacionadas tenuemente puestas juntas por un accidente de la historia, como Yugoslavia ... Tal vez un día la ‘ciencia de la computación’ será, como Yugoslavia, dividida en sus partes componentes. Eso podría ser bueno. Especialmente si eso significa independencia para mi tierra nativa, ‘*hacking*’” [5]. Él no argumenta contra la ciencia de la computación, pero apela por usar ‘arte

computacional' que es más atractivo para los *hackers* (su término para programadores de élite).

Dana Gardner, del *Yankee Group*, no comparte esta idea. Él compara la situación actual del desarrollo de *software* con el renacimiento pre-industrial, cuando ricos benefactores comisionaron grupos de artesanos altamente entrenados para grandes trabajos artísticos únicos [4]. Él dice, “personas de negocios están trabajando mucho más cerca al reino de Henry Ford, donde ellos buscan reutilización, partes intercambiables, procesos automatizados, líneas de montaje altamente industrializadas”

- *OK, entonces computación tiene mucho arte y su propia ciencia, a pesar de que algunos de ustedes no están seguros si es ciencia. Sin embargo, ¿tiene profundidad la ciencia de la computación? ¿Existen principios fundamentales que no sean obvios para aquellos que no entiendan la ciencia? ¿Quién pensaría que la velocidad de la luz es la misma para todos los observadores hasta que Einstein postuló la relatividad? ¿O que las partículas viajan en ondas probabilísticas hasta que Schroedinger postuló la mecánica cuántica? ¿Hay algo como eso en ciencia de la computación?*

¿Puede la ciencia de la computación sorprender?

En la Figura se pueden ver seis categorías de principios en computación con ejemplos de descubrimientos importantes que no son obvios para amadores [2]. Explotando esos principios, los profesionales son capaces de resolver problemas que amadores encontrarían verdaderamente incomprensibles.

Area	Problem
Computation	<ul style="list-style-type: none">• Unbounded error accumulation on finite machines• Non-computability of some important problems• Intractability of thousands of common problems• Optimal algorithms for some common problems• Production quality compilers
Communication	<ul style="list-style-type: none">• Lossless file compression• Lossy but high-fidelity audio and video compression• Error correction codes for high, bursty noise channels• Secure cryptographic key exchange in open networks
Interaction	<ul style="list-style-type: none">• Arbitration problem• Timing-dependent (race-conditioned) bug problem• Deadlock problem• Fast algorithms for predicting throughput and response time• Internet protocols• Cryptographic authentication protocols
Recollection	<ul style="list-style-type: none">• Locality• Thrashing• Search• Two-level mapping for access to shared objects
Automation	<ul style="list-style-type: none">• Simulations of focused cognitive tasks• Limits on expert systems• Reverse Turing tests
Design	<ul style="list-style-type: none">• Objects and information hiding• Levels• Throughput and response time prediction networks of servers

- *OK. Estoy encontrando esto comprensible. Pero todavía tengo una preocupación. ¿Vale la pena invertir sea mi tiempo o dinero de I+D en ciencia de la computación? En su libro de 1996, *The End of Science*, el periodista John Horgan argumenta que muchos campos científicos han sido saturados. Ya*

se han descubierto muchos de los principios básicos y nuevas descubiertas son menos y menos frecuentes. ¿Porque es diferente la ciencia de la computación? Una vez que la secuencia actual de computer-science-in-the-making se resuelva, y asumiendo que los hackers no se separen, ¿la ciencia de la computación se extinguirá?

La ciencia de la computación prospera en relacionamientos

Horgan argumentó en 1996 que nuevas descubiertas científicas requieren llegar a dominar grandes cantidades de complejidad. En 2004 repitió su conclusión principal: “La ciencia nunca más dará revelaciones tan monumentales como la teoría de la evolución, la relatividad, mecánica cuántica, la teoría del *big-bang*, genética basada en DNA. ... Algunos objetivos poco probables de la ciencia aplicada – como la inmortalidad, astronaves más rápidas que la luz, y máquinas superinteligentes – pueden eludirnos para siempre [6].

¿La ciencia de la computación ya hizo todas las descubiertas que debería? ¿Sólo nos queda un progreso incremental? ¿La ciencia de la computación ha llegado hasta el fin de la era histórica de la ciencia?

Yo no lo creo. Horgan argumenta que el número de áreas científicas es limitado y cada uno está lentamente siendo agotado. Pero la ciencia de la computación está caminando para un lado distinto. Está constantemente haciendo relaciones con otras áreas, cada una abre un nuevo campo. Paul Rosenbloom ha colocado esto elocuentemente en su reciente análisis de ciencia de la computación e ingeniería [11].

Rosenbloom hizo un seguimiento de la historia de la ciencia de la computación en sus relaciones con la física, vida y ciencias sociales. Con cada una la ciencia de la computación ha abierto nuevos campos implementando, interactuando e insertándose en esos campos. Ejemplos incluyen sistemas autónomos, bioinformática, biometría, biosensores, prótesis cognitivas, ciencias cognitivas, cyborgs, computación basada en DNA, computación inmersiva, computación neural y computación cuántica. Rosenbloom cree que el constante nacimiento y riqueza de nuevos relacionamientos garantiza un futuro brillante para el área.

- Todo bien, voy a aceptarlo. Ustedes tienen ciencia, ustedes tienen arte, ustedes pueden sorprender y ustedes tienen un futuro. Pero ustedes también tienen un problema de credibilidad. Por 1960 ustedes alegaron que en breve se construirían sistemas inteligentes que rivalizarían con expertos humanos y harían nuevas descubiertas científicas. Por 1970 dijeron que serían capaces de sistemáticamente producir sistemas de software confiables y seguros. Por 1980 era que desaparecerían el papel, las universidades, las bibliotecas y los viajes diarios al trabajo. Ninguna de esas cosas ocurrió. Por 1990 ustedes contribuyeron a la explosión de Internet y luego fracasaron con la quiebra de las dot-com. Ahora hacen todo tipo de alegaciones acerca de sistemas seguros, bloqueo de spam, colaboración, sistemas empresariales, diseño de DNA, nanotecnología y más. Porqué debería creerles?

Validando las alegaciones de la ciencia de la computación

Acá nos tienes. Hemos permitido que la promoción exagerada de departamentos publicitarios se infiltre en nuestros laboratorios. En una muestra de 400 *papers* de ciencia de la computación publicados antes de 1995, Walter Tichy encontró que aproximadamente 50% de esos modelos propuestos o hipótesis no fueron testados [12]. En otros campos de la ciencia la fracción de *papers* con hipótesis no testadas era cerca de 10%. Tichy concluyó que nuestro fracaso al testar permitió que muchas ideas poco sólidas sean colocadas en práctica restando credibilidad a nuestra área como una ciencia. La relativa juventud de nuestra área – apenas 60 años – no explica el bajo índice de testes. Tres generaciones parecen suficiente tiempo para que los científicos de la computación puedan establecer que sus principios son

sólidos.

La percepción de nuestra área parece ser una cuestión generacional. Los miembros más antiguos tienden a identificarse con una de las tres raíces del área – ciencia, ingeniería o matemática. El paradigma científico es largamente invisible dentro de los otros grupos.

La generación más joven, mucho menos intimidados que los más antiguos con nuevas tecnologías, es más abierta al pensamiento crítico. La ciencia de la computación siempre ha sido parte de su mundo, ellos no cuestionan su validez. En su investigación, están cada vez más siguiendo el paradigma científico. Tichy me dijo que la reciente literatura científica muestra un marcado aumento en testes.

El paradigma científico no ha sido parte de la percepción dominante de la ciencia de la computación. pero en breve lo será.

Referencias

- [1]. Abelson, H.G. and Sussman, G.J. Structure and Interpretation of Computer Programs, 2nd ed. MIT Press, 1996.
- [2]. Denning, P. Great principles of computing. Commun. ACM 46, 10 (Nov. 2003), 15–20.
- [3]. Denning, P. et al. Computing as a discipline. Commun. ACM 32, 1 (Jan. 1989), 9–23.
- [4]. Ericson, J. The psychology of service-oriented architecture. Portals Magazine (Aug. 2004); www.portalsmag.com/articles/default.asp?ArticleID=5872.
- [5]. Graham, P. Hackers and Painters: Big Ideas from the Computer Age. O'Reilly and Associates, 2004.
- [6]. Horgan, J. The end of science revisited. IEEE Computer (Jan. 2004), 37–43.
- [7]. Latour, B. Science in Action. Harvard University Press, 1987.
- [8]. National Research Council. Computer Science: Reflections on the Field, Reflections from the Field. National Academy Press, 2004.
- [9]. National Research Council. Academic Careers for Experimental Computer Scientists and Engineers. National Academy Press, 1994.
- [10]. Parnas, D. Software engineering: An unconsummated marriage. Commun. ACM 40, 9 (Sept. 1997), 128.
- [11]. Rosenbloom, P. A new framework for computer science and engineering. IEEE Computer (Nov. 2004), 31–36.
- [12]. Tichy, W. Should computer scientists experiment more. IEEE Computer (May 1998), 32–40.

Peter J. Denning (pjd@nps.edu) is the director of the Cebrowski Institute for information innovation and superiority at the Naval Postgraduate School in Monterey, CA, and is a past president of ACM.