

REVISTA **Bits**

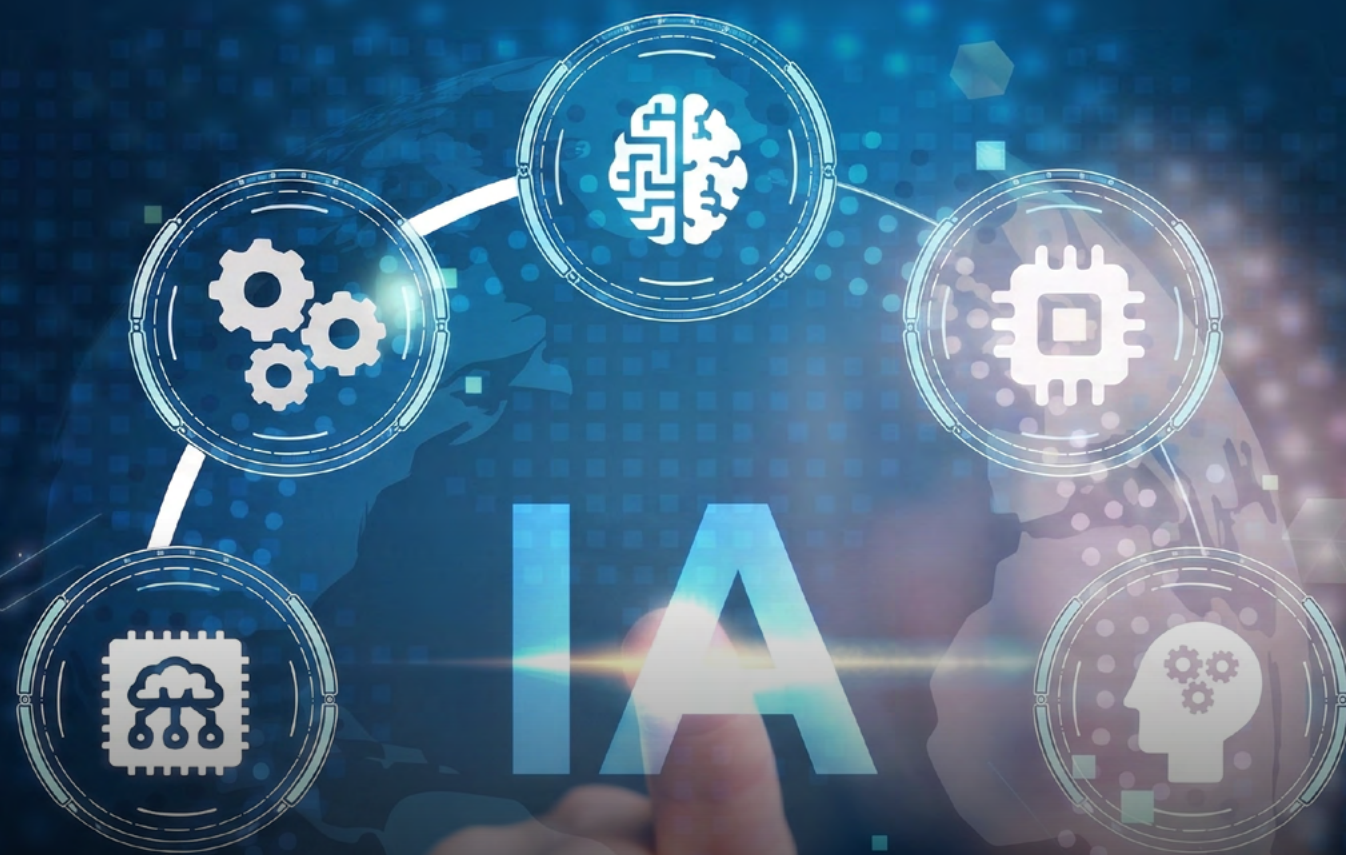
Edición Nº29 / Segundo Semestre 2026

DEPARTAMENTO DE CIENCIAS DE LA
COMPUTACIÓN DE LA UNIVERSIDAD DE CHILE



fcfm

Ciencias de la
Computación
FACULTAD DE CIENCIAS
FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE



Revisitando la IA generativa

Del código a la especificación: Cómo los LLMs
están transformando el desarrollo de software

/ Matías Toro

El cerebro y sus interrogantes: Preguntas
abiertas en neurociencia computacional

/ Felipe Bravo-Márquez y Rodrigo Carrasco-Davis

"El Primer Cerebro Electrónico", Valparaíso,
Chile, 1957

/ Juan Álvarez Rubio

Contenidos

1 Editorial

/ Federico Olmedo

Investigación Destacada

2 Integrando el pensamiento computacional y la programación en la formación inicial docente en Chile

/ Jocelyn Simmonds, Nancy Hitschfeld Kahler, Valeria Henríquez y Eliana Scheihing

Premio Turing

8 Premio Turing 2024: Sutton y Barto, los pioneros del aprendizaje por refuerzo

/ Javier Ruiz Del Solar

Revisitando la IA Generativa

14 La base material de la inteligencia artificial: Chips, datacenters y geopolítica

/ Felipe Bravo-Márquez, Claudio Gutiérrez y Luis Mateu

24 LatamGPT: Construyendo IA desde el sur global

/ LatamGPT Team

31 Del código a la especificación: Cómo los LLMs están transformando el desarrollo de software

/ Matías Toro

38 La IA en la investigación matemática. Herramientas para entender el presente y el futuro

/ Bernardo Subercaseaux

45 Fiscal Heredia: IA generativa al servicio de la investigación penal

/ Richard Weber y Valentina Reyes

51 El cerebro y sus interrogantes: Preguntas abiertas en neurociencia computacional

/ Felipe Bravo-Márquez y Rodrigo Carrasco-Davis

Vinculación con el Medio

71 Vinculación con el medio en el DCC

/ Adolfo Carrasco Acosta

Computación y Sociedad

74 "El Primer Cerebro Electrónico", Valparaíso, Chile, 1957

/ Juan Álvarez Rubio

82 Profesor José Alberto Pino Urtubia fue distinguido como Profesor Emérito de la Universidad de Chile

Estudiantes DCC

88 Tesis y memorias

/ Jesus Perez-Martin, Asunción Gómez, Josefa Robert, Alan Acevedo y Cristóbal Carmona



COMITÉ EDITORIAL

Andrés Abeliuk
María Cecilia Bastarrica
Eduardo Graells-Garrido
Claudio Gutiérrez
Alejandro Hevia
Ana Gabriela Martínez
Jocelyn Simmonds
Iván Sipirán

EDITOR GENERAL

Federico Olmedo

EDITORA PERIODÍSTICA

Ana Gabriela Martínez

DISEÑO

Paulette Filla

FOTOGRAFÍAS E IMÁGENES

Comunicaciones DCC

Revista Bits del Departamento de Ciencias de la Computación de la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile se encuentra bajo Licencia Creative Commons Deed - Atribución/Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional. Basada en una obra en www.dcc.uchile.cl



Revista Bits N°29

ISSN 0718-8005 (versión impresa)

dcc.uchile.cl/bits

ISSN 0717-8013 (versión en línea)

Departamento de Ciencias de la Computación

Avda. Beauchef 851, 3° piso,
edificio norte. Santiago, Chile.
837-0459 Santiago

 dcc.uchile.cl

 +56 22 9780652

 revistabits@dcc.uchile.cl

 / [dccuchile](https://www.dccuchile.cl)

El contenido de los artículos publicados en esta revista, es de exclusiva responsabilidad de sus autores y no refleja necesariamente el pensamiento del Departamento de Ciencias de la Computación de la Universidad de Chile.



Editorial

Federico Olmedo

Editor General
Revista Bits



"Mythos y Fable de Anthropic bloqueados después de que Estados Unidos prohibiera su uso por parte de extranjeros"... Titulares de ese estilo copaban las portadas de los principales medios tecnológicos a mediados de este año. Sin duda, el panorama de la inteligencia artificial (IA) generativa ha evolucionado vertiginosamente en los últimos años. Hemos pasado de los primeros LLMs operados a través de interfaces conversacionales y una promesa incierta de aumento de productividad, a cambios radicales en el quehacer profesional y científico de diversas disciplinas, hasta el punto de transformar a la IA en el centro de disputas geopolíticas globales.

Frente a este escenario, en este número de la Bits vamos a revisitar la IA generativa para abordar nuevos interrogantes. Analizaremos, entre otros, el impacto (¡real!) que está teniendo en ámbitos como el proceso de descubrimiento científico o la ingeniería de software, la forma en que estos modelos se relacionan con la neurociencia, y la relevancia de desarrollar modelos latinoamericanos que reflejen nuestra propia identidad.

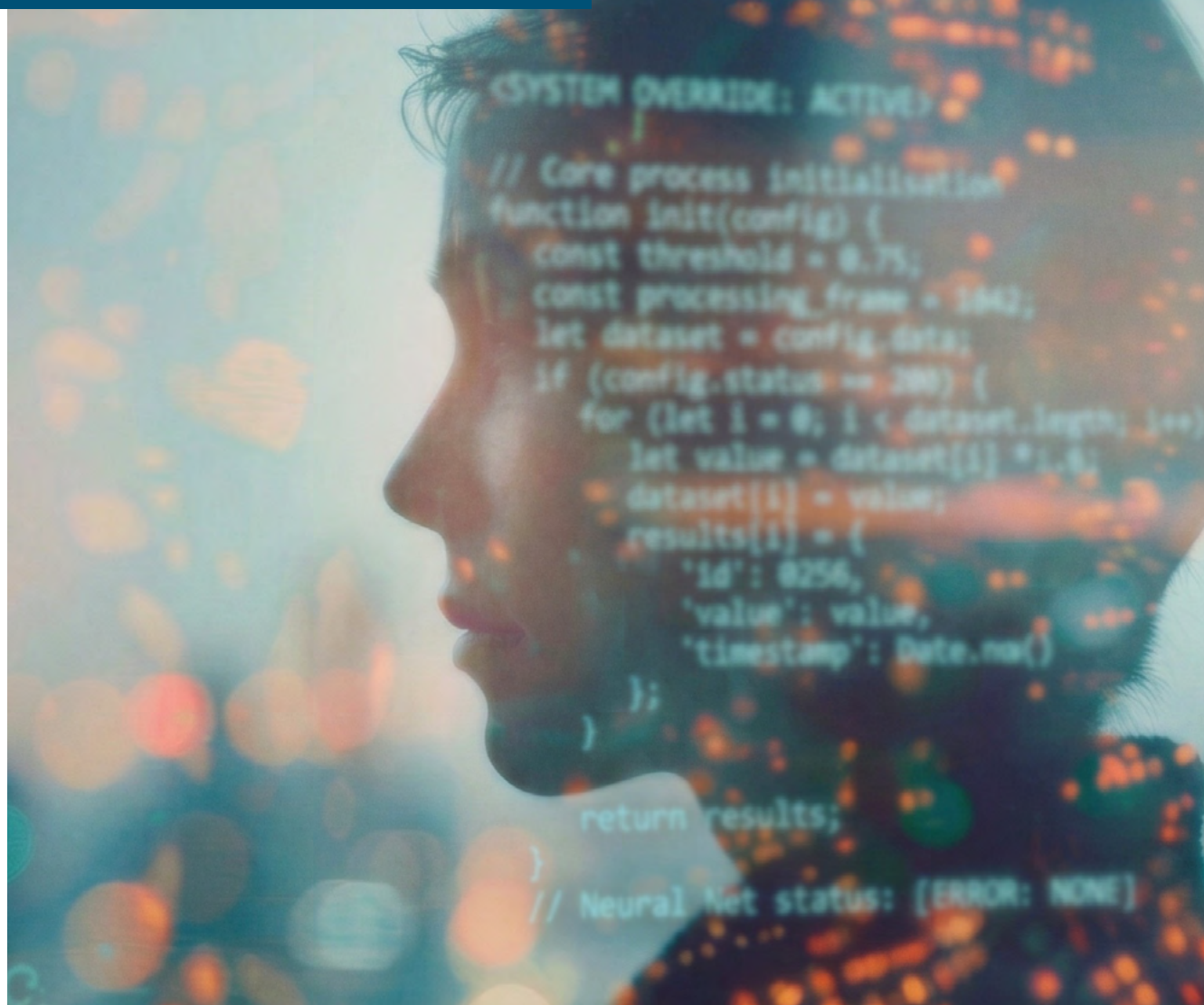
Junto a este tema central, en las demás secciones de la revista abordamos avances recientes, y celebramos el legado y la historia de nuestra comunidad. En "Investigación Destacada" compartimos una propuesta para formar a lxs futurxs profesores de Chile en competencias digitales. En "Premio Turing" destacamos el reconocimiento a Richard Sutton y Andrew Barto, por sus contribuciones al aprendizaje por refuerzo.

En "Computación y Sociedad" viajamos a 1957 para recordar la llegada del primer "cerebro electrónico" a Chile, y rendimos homenaje al Profesor José Alberto Pino Urtubía, cofundador del DCC, recientemente nombrado Profesor Emérito de la Universidad de Chile. Por su parte, en "Vinculación con el Medio" exploramos los distintos puentes que el DCC está tendiendo entre las aulas, la industria y su red Alumni. Cerramos con "Estudiantes DCC", visibilizando el trabajo final de graduadxs recientes.

Esperamos que disfruten de esta edición. Como siempre, recibimos sus comentarios en revistabits@dcc.uchile.cl. **B**

1 Deutsche Welle. 14 de junio de 2026. <https://www.dw.com/en/anthropics-mythos-fable-blocked-after-us-bans-foreign-use/video-77543276>.

Integrando el pensamiento computacional y la programación en la formación inicial docente en Chile





Jocelyn Simmonds

Doctora en Ciencias de la Computación por la Universidad de Toronto, Canadá. Profesora Asociada del Departamento de Ciencia de la Computación de la Universidad de Chile. Sus intereses de investigación están en las áreas de ingeniería de software, el desarrollo del pensamiento computacional en niños y niñas, y cómo mejorar la participación de mujeres en computación y tecnología.

✉ jsimmond@dcc.uchile.cl



Nancy Hitschfeld Kahler

Doctora en Technischen Wissenschaften por la ETH-Zurich, Suiza. Profesora Titular del Departamento de Ciencias de la Computación de la Universidad de Chile. Sus líneas de investigación incluyen mallas de polígonos y poliedros, computación en GPU, algoritmos en ciencia e ingeniería computacional y educación en computación. Participa activamente en comisiones y actividades para atraer mujeres a STEM.

✉ nancy@dcc.uchile.cl



Valeria Henríquez

Doctora por la Universidad Politécnica de Madrid, España. Profesora Auxiliar en la Universidad Austral de Chile. Sus intereses de investigación incluyen tecnologías educativas, métodos ágiles en ingeniería de software y mejora de procesos organizacionales. Su investigación actual se centra en analítica del aprendizaje y la integración de IA generativa en la ingeniería de software.

✉ valeria.henriquez@inf.uach.cl



Eliana Scheihing

Doctora por la Université Catholique de Louvain, Bélgica. Profesora Titular y actual Directora del Instituto de Informática de la Universidad Austral de Chile. Sus intereses de investigación actuales incluyen: alfabetización en inteligencia artificial en contextos escolares, aprendizaje mejorado por tecnologías, analítica del aprendizaje, minería de datos y estadística bayesiana.

✉ escheihi@inf.uach.cl

Resumen / En este artículo presentamos un modelo para desarrollar habilidades de pensamiento computacional y programación usando Aprendizaje Basado en Proyectos en la Formación Inicial Docente universitaria chilena. Este modelo promueve el diseño de proyectos interdisciplinarios que abordan preguntas de investigación derivadas del Currículum Nacional¹ escolar, alineadas a los objetivos de aprendizaje de varias materias, que refuercen tanto el aprendizaje en las materias abordadas y al mismo tiempo desarrollen habilidades de pensamiento computacional, y que más tarde puedan ser aplicados en el aula escolar. El modelo fue cocreado y validado con formadores de docentes de la Universidad Austral de Chile, Universidad Católica del Norte y Universidad de O'Higgins.

Introducción

El desarrollo de habilidades de pensamiento computacional (PC) [1] a partir de una edad temprana se ha convertido en un desafío a abordar en todo el mundo, dado los procesos cognitivos que involucra. Para enfrentar y buscar una solución a un problema mediante el PC, se hace uso de habilidades como la abstracción, la descomposición, el reconocimiento de patrones y la elaboración de algoritmos (mediante la programación). Estas habilidades permiten resolver problemas en disciplinas tales como biología, historia y música, entre otras, y no sólo en el área de la computación y matemáticas.

En Chile, desde el año 2012, tanto universidades como organizaciones privadas han estado liderando iniciativas para desarrollar el PC con actividades extra curriculares. Recién en el año 2018 se incorporó esta temática en el Currículum Nacional, como parte de un curso electivo de profundización de matemáticas para tercero y cuarto medio, pero esto fue sin tomar en cuenta que la mayoría de los programas de pedagogías de las

1 <https://www.curriculumnacional.cl/>



Figura 1 / Parte del equipo FONDEF. De izquierda a derecha: Daniel Estrada, Paulo Contreras, Valentín Muñoz, Nancy Hitschfeld Kahler, Karin Mercado, Eliana Scheihing, Jocelyn Simmonds y Lucía Márquez.

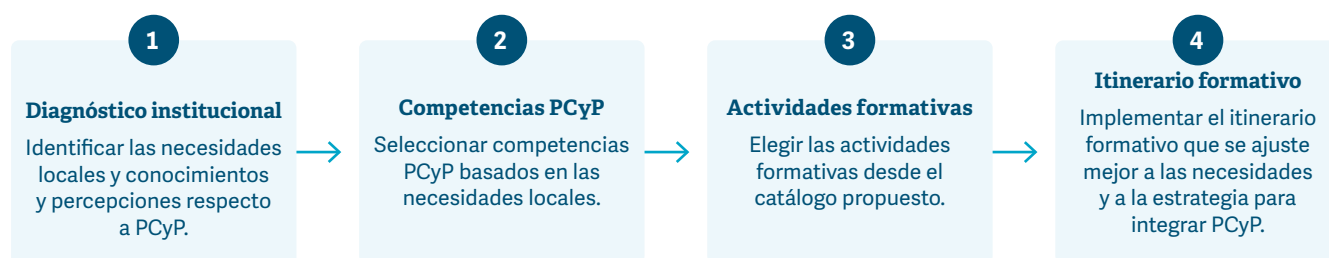


Figura 2 / Esquema del modelo para la integración de PCyP en la Formación Inicial Docente (FID).

universidades chilenas no incorporan el desarrollo de habilidades de PC en sus estudiantes, como tampoco la programación (desde ahora ocuparemos la sigla PCyP para referirnos a ambos tópicos juntos).

Analizando más de 100 programas de pedagogía ofrecidos en Chile al año 2022, solo 16 incluían tópicos de PCyP, y eran principalmente programas de matemáticas o física. Es así como profesores en servicio y estudiantes de pedagogía que deseen desarrollar estas habilidades necesariamente deben hacerlo en forma autodidacta. Además, si bien existen algunas oportunidades de capacitación, no están enfocadas a desarrollar las habilidades del PCyP en el contexto de los objetivos de aprendizaje asociados a las distintas asignaturas en el Currículum Nacional, lo que dificulta su inclusión en el aula.

Las autoras de este artículo y sus colaboradores/as han explorado y desarrollado distintas estrategias y metodologías para introducir PCyP en el aula escolar, para estudiantes y profesores en servicio [2, 3, 4]. Estas experiencias nos han hecho evidente que la inclusión de PCyP en educación primaria y secundaria chilena no se masificará si los estudiantes de pedagogía no adquieren estas habilidades de manera contextualizada en la disciplina que estudian durante su formación.

En este artículo presentamos un modelo que desarrollamos para guiar a Escuelas de Pedagogía en la integración del PCyP en cualquier programa de pedagogía. Este modelo ofrece un

marco estructurado para hacer esta integración de manera sistemática y articulada al contexto regional del programa de pedagogía a actualizar, pero al mismo tiempo abordando los objetivos de aprendizaje que define el Currículum Nacional para la disciplina asociada. Esto es mediante un enfoque de Aprendizaje Basado en Proyectos (ABP), para que a futuro los y las docentes en formación puedan reutilizar las actividades en el aula escolar, cuando ejerzan como profesor/a.

Este modelo fue el resultado principal del proyecto FONDEF IDeA ID23I10335, desarrollado en consorcio por la Universidad de Chile, Universidad Austral de Chile (UACH), Universidad Católica del Norte (UCN) y la Universidad de O'Higgins (UOH), durante julio 2023 – julio 2025 (ver Figura 1). Si bien intuimos que este modelo puede ser ocupado para integrar el PCyP en distintas asignaturas y a distintos niveles escolares, en este proyecto nos enfocamos en validar el modelo para el Currículum Nacional desde séptimo básico a segundo medio, y en asignaturas como matemáticas, ciencias naturales e inglés.

Modelo para la integración de PCyP en la Formación Inicial Docente

El modelo desarrollado consiste de cuatro componentes para integrar PCyP en un programa existente de pedagogía (ver Figura 2):

1. **Diagnóstico institucional:** este componente permite identificar las variables de contexto relevantes para la integración de PCyP, como el nivel de conocimiento que los distintos actores tienen acerca de PCyP, y sus actitudes frente al uso de tecnología en el aula. Este diagnóstico se hace con respecto a los tres tipos de actores claves de la Formación Inicial Docente (FID): docentes en formación, formadores de docentes y autoridades.
2. **Competencias PCyP:** este componente organiza los conocimientos, habilidades y actitudes de PCyP que se han identificado en la literatura. Hay que definir los niveles de logro que desean desarrollar en un programa de pedagogía, basándose en el diagnóstico institucional.
3. **Actividades formativas:** este componente es un catálogo de actividades formativas para la integración del PCyP mediante el Aprendizaje Basado en Proyectos (ABP), buscando desarrollar objetivos de aprendizaje del Currículo Nacional y de las asignaturas FID donde se desarrollan estas actividades.
4. **Itinerario formativo:** este componente ordena las distintas estrategias que se presentan en la literatura para integrar el PCyP en la FID, desde lo microcurricular a lo macrocurricular. Estas se presentan en una matriz de orientaciones por actor relevante, haciendo una relación directa con el diagnóstico institucional.

Pongámonos en el caso que el diagnóstico institucional del programa de pedagogía a actualizar detecta el siguiente contexto:

- Todos los actores presentan un bajo nivel de conocimientos acerca de PCyP.
- La capacidad de la institución para hacer cambios curriculares es media.
- Los formadores de docentes son cautos con respecto a sus conocimientos previos de PCyP.
- La infraestructura tecnológica disponible es de alto nivel.

En este caso, el modelo propone las siguientes orientaciones:

- Hay que formar un equipo de trabajo multidisciplinario, que incluya tanto a formadores de docentes como expertos en PCyP.
- Antes de integrar PCyP en el aula, se debe realizar un taller presencial con los formadores de docentes, donde se entregan conocimientos de PCyP mediante el desarrollo de actividades ABP.

Definimos un problema interdisciplinario que se podía resolver desarrollando las habilidades de Pensamiento Computacional y Programación, y al mismo tiempo lograba el aprendizaje esperado en la asignatura correspondiente.

- Se debe integrar PCyP en asignaturas existentes del programa de pedagogía a actualizar, preferiblemente en el desarrollo de una sola unidad.
- Las actividades ABP iniciales deben ser acotadas, reforzando la conexión entre el desarrollo de PCyP, los objetivos de aprendizaje del Currículo Nacional y la didáctica de la disciplina.
- Las actividades ABP iniciales pueden ser prediseñadas, pero a medida que el equipo de trabajo aumente su nivel de conocimientos de PCyP, se debe avanzar hacia la cocreación de nuevas actividades ABP que se ajusten mejor al contexto local.

La metodología que se siguió para diseñar y validar este modelo es Design Science, centrándose en ciclos de diagnóstico, diseño, pilotaje y validación, para asegurar la pertinencia y aplicabilidad del modelo en distintos contextos universitarios.

Experiencias en el campo

Uno de los aspectos clave de nuestra estrategia para integrar PCyP en la formación de docentes fue diseñar dos actividades pedagógicas contextualizadas en la temática de su disciplina. Estos diseños son lo suficientemente flexibles para ser adaptados a varias disciplinas y al mismo tiempo ser fácilmente transferibles a la enseñanza en colegios en donde trabajarán a futuro estos docentes en formación. Otro aspecto clave es que primero hay que formar (y empoderar) a los formadores de docentes que van a liderar la integración de PCyP en su institución.

Para lograr diseñar actividades contextualizadas, como equipo primero definimos el nivel escolar que queríamos apoyar, luego buscamos objetivos de aprendizaje de distintas asignaturas (al menos 2) en ese nivel que podían ser abordados y/o reforzados usando PCyP. Luego definimos un



Figura 3 / Académicos y académicas de la Universidad Austral de Chile, Universidad Católica del Norte y Universidad de O'Higgins, que participaron en el taller para formadores de docentes, además de integrantes del equipo FONDEF.

problema interdisciplinario que se podía resolver desarrollando las habilidades de PCyP, y al mismo tiempo lograba el aprendizaje esperado en la asignatura correspondiente. Los proyectos creados fueron “Vida Sana” y “Ecosistema”.

El proyecto “Vida Sana” combina objetivos de aprendizaje de matemáticas y tecnología (desde séptimo básico a segundo medio), inglés como segundo lenguaje (séptimo y octavo) y ciencias naturales (desde octavo a segundo medio), es decir, involucra 4 asignaturas. La pregunta de investigación fue “¿Qué es una comida sana?”. Esta pregunta se debe explorar primero de manera desenchufada, de tal manera que los participantes definan estrategias para determinar si un cierto plato de comida se puede considerar sano. Durante este proceso, se definen conceptos claves de PC, usando ejemplos relacionados al proyecto para ilustrar los conceptos de abstracción, descomposición, patrones y algoritmos. Una vez que los participantes diseñan un algoritmo para determinar si una comida es saludable, ocupamos prototipos parciales para introducir conceptos de la programación (en este caso, ocupando AppInventor). Al completar prototipos incrementales, se aprovecha de explicar los conceptos de programación fundamentales, como creación y uso de variables, instrucciones con operaciones matemáticas, depuración y ejecución del programa, de una manera más escalonada.

El proyecto “Ecosistema” aborda objetivos de aprendizaje de ciencias naturales, específicamente de Ciencias de la Tierra y Biología (primero medio), y Matemáticas (primero y segundo medio), y Tecnologías. Este proyecto es guiado por la pregunta de investigación: “¿Qué factores afectan la proliferación de ciertas especies o cuáles logran el equilibrio de un ecosistema?”. Al igual que para el proyecto de Vida Sana, primero se sugiere explorar de manera desenchufada, para un ecosistema desértico, qué tipo de animales existen y cómo se alimentan. Después se simplifica el problema y eli-

gen dos animales relevantes, que estén relacionados entre sí, para diseñar una simulación que permita ingresar ciertos valores que permitan entender cuándo el ecosistema está en equilibrio y cuándo este pierde dicho equilibrio. Este proyecto requiere junto a la creación y uso de variables, incluir condicionales y repeticiones. Para esto diseñamos prototipos incrementales en Scratch que los participantes pueden ir modificando y ejecutando con distintos escenarios.

Usando estos dos proyectos, organizamos un taller de tres días para formadores de docentes. Asistieron 14 profesores de tres universidades antes mencionadas (ver Figura 3). Durante el primer día, los participantes trabajaron el proyecto “Vida Sana” y el segundo día en el proyecto “Ecosistema”. El tercer día, los participantes trabajaron en grupos, definiendo un proyecto para integrar PCyP en un curso en su institución. Los dejamos libres para adaptar uno de los proyectos presentado en el taller o proponer un proyecto totalmente distinto. El requerimiento principal era que ellos tenían que incorporar ese proyecto la próxima vez que dictaran el curso y que debía ser evaluado. El equipo del proyecto FONDEF apoyó a los participantes en la definición del proyecto y la creación del material necesario (como diapositivas y prototipos).

Los participantes del taller realizaron un total de 11 experiencias pilotos para validar el modelo (6 en la UCN, 4 en la UACH y 1 en la UOH). Cuatro de estos pilotos se desarrollaron el primer semestre de 2024, 5 durante el segundo semestre de 2024 y 2 durante el primer semestre de 2025. Diez de estas experiencias se desarrollaron como una actividad ABP dentro de una unidad de un curso existente de alguna carrera de pedagogía. La experiencia restante constó en la realización del curso “Didáctica de la tecnología: Pensamiento Computacional y Programación mediante Aprendizaje Basado en Proyectos para la Formación Inicial Docente” como curso electivo para una serie de carreras de pedagogía (en la

UACH). Este es un ejemplo de itinerario formativo del modelo, y consiste de 5 módulos, que se enfocan en el desarrollo de conceptos claves de PCyP, de la integración de PCyP en el aula mediante el ABP y el uso de diseños didácticos como una forma de sistematizar la creación de este tipo de experiencia de aprendizaje. Este curso se puede hacer en línea, es de libre disposición (ver <https://tilo.dcc.uchile.cl>).

Un total de 263 docentes en formación completaron estas actividades formativas, de 12 programas en las tres universidades participantes. En la UACH, participaron estudiantes de los siguientes programas: Antropología; Pedagogía en Comunicación en Lengua Inglesa; Pedagogía en Educación Física, Deportes y Recreación; Pedagogía en Educación Parvularia; Pedagogía en Lenguaje y Comunicación; Periodismo, y Programa de Formación de Profesores para Enseñanza Media, en las especialidades de artes visuales, biología y ciencias naturales. En la UCN, participaron estudiantes de los siguientes programas: Pedagogía en Educación Básica con Especialización; Pedagogía en Inglés; Pedagogía en Lengua Inglesa para Básica y Media, y Pedagogía en Matemáticas Educación Media. Finalmente, en la UOH participaron estudiantes de Pedagogía en Matemáticas.

Para facilitar la adopción del modelo, creamos un segundo curso en línea, llamado “Modelo de Integración de Pensa-

miento Computacional y Programación en la Formación Inicial Docente”. Este es un curso de 3 módulos, diseñado para guiar a una institución de educación superior en la aplicación del Modelo de Integración de PCyP en la FID. Se basa en el diagnóstico realizado en las universidades Austral de Chile, Católica del Norte y De O’Higgins, las experiencias previas del equipo de investigación y la literatura científica relacionada a experiencias exitosas de inclusión de PCyP en el contexto escolar y universitario.

Conclusión

Dado lo anterior, podemos concluir que el proyecto FONDEF IDeA ID23110335 ha cumplido con éxito su objetivo de proveer a la Formación Inicial Docente de Chile de un Modelo de Integración de PCyP en la FID [5] que toma en cuenta los contextos de tres universidades regionales, anclado en la metodología de Aprendizaje Basado en Proyectos para desarrollar objetivos de aprendizaje del Currículo Nacional. Este resultado no sólo permite elevar las competencias de los formadores de profesores, sino que, a través de ellos, sienta las bases para que las futuras generaciones de docentes estén equipadas para educar en las habilidades digitales y el pensamiento crítico que demanda el siglo XXI, asegurando así una mayor pertinencia y calidad en la educación nacional. ■

Equipo FONDEF

U. de Chile: Jocelyn Simmonds, Nancy Hitschfeld Kahler, Valentín Muñoz, Cecilia Casanova y Catalina Torrent.

UACH: Eliana Scheihing, Valeria Henríquez, Paulo Contreras, Karin Mercado, Daniel Estrada y Lucía Márquez.

Colaboradores

UCN: Víctor González Escobar (investigador FONDEF), Adriana Tapia Barraza, Alejandro Guerrero Salinas, Cristina Manzaneda Herrera, David Rojas Contreras, Michel Molina del Sol, Nathalie Castro Muñoz y Ricardo Guerrero Soto.

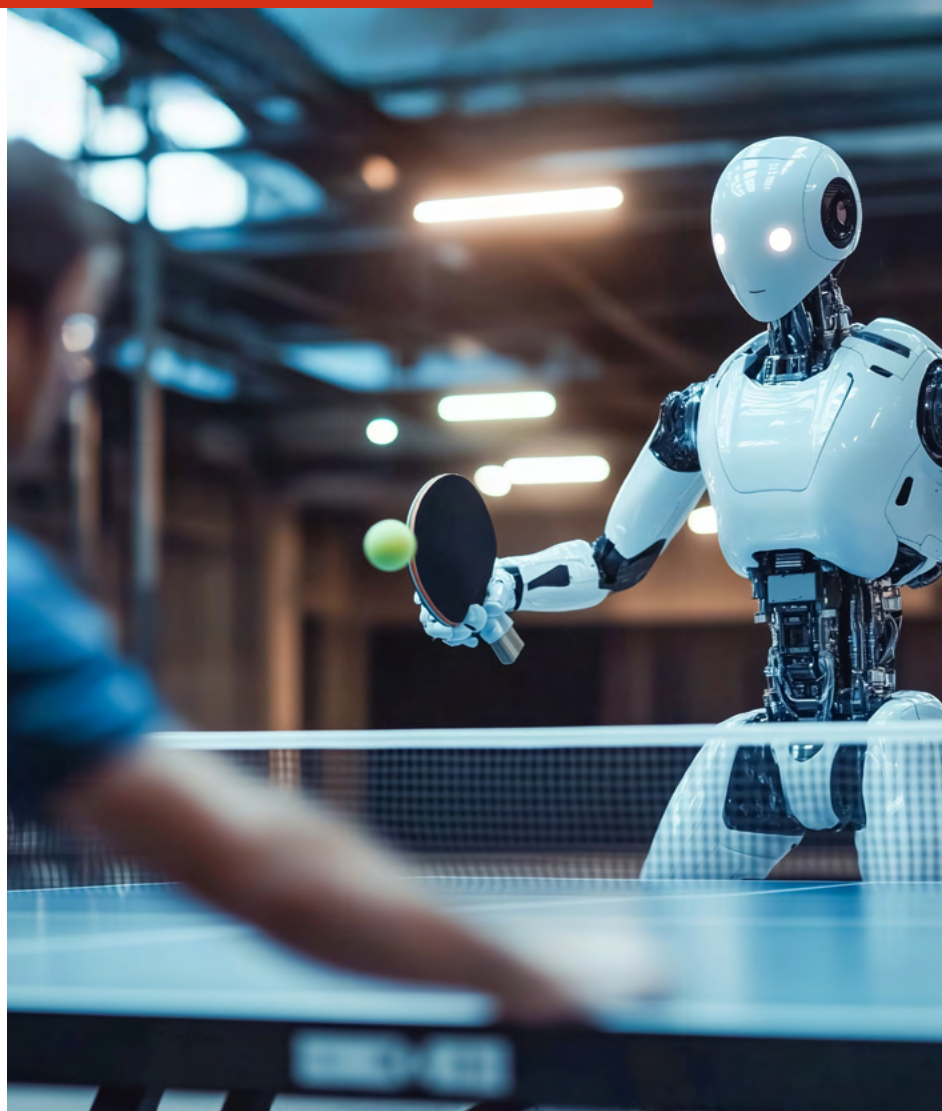
UOH: David Gómez (investigador FONDEF), Jennifer Fuentes Constancio y Valentina Giaconi.

Referencias

- [1] J. M. Wing. 2006. Computational thinking. *Commun. ACM* 49, 3 (March 2006), 33–35. <https://doi.org/10.1145/1118178.1118215>.
- [2] J. Simmonds, F.J. Gutiérrez, C. Casanova, C. Sotomayor, and N. Hitschfeld. 2019. A Teacher Workshop for Introducing Computational Thinking in Rural and Vulnerable Environments. In *Proc. of the 50th ACM Technical Symposium on CS Education (SIGCSE’19)*. ACM, 1143–1149.
- [3] J. Simmonds, F.J. Gutiérrez, N. Hitschfeld, C. Torrent, C. Casanova, S. Toro-Guajardo, and V. Muñoz. 2022. A Model for Infusing Computational Thinking Skills at Teacher Colleges in a Developing Country. In *Proc. of the 17th Workshop in Primary and Secondary Computing Education*. ACM, Article 17, 2 pages.
- [4] V. Henríquez, I. Hilliger. Blended learning in rural K-12 education: Stakeholder dynamics and recommendations. *J. Comput. Assist. Learn.* 40(4): 1463-1480 (2024).
- [5] J. Simmonds, N. Hitschfeld-Kahler, E. Scheihing, V. Henríquez, V. Muñoz, C. Casanova, C. Torrent, P. Contreras, K. Mercado, D. Estrada, L. Márquez. Designing Project-based learning experiences to integrate Computational Thinking skills at Teacher Colleges. *WiPSCE 2024*: 38:1-38:2.

Premio Turing 2024:

Sutton y Barto, los pioneros
del aprendizaje por refuerzo



Javier Ruiz Del Solar

Doctor en Ingeniería por la Technische Universität Berlin, Alemania. Profesor Titular del Departamento de Ingeniería Eléctrica de la Universidad de Chile. Sus líneas de investigación incluyen aprendizaje de máquinas, robótica móvil y aplicaciones de robótica e inteligencia artificial en minería y ambientes industriales. Imparte los cursos Inteligencia Computacional, Procesamiento Avanzado de Imágenes y Robótica Móvil en la Universidad de Chile y es el líder de la línea de I+D de Autonomía Minera del Advanced Mining Technology Center (AMTC).

✉ jruizd@ing.uchile.cl

Resumen / En marzo de 2025, la Association for Computing Machinery (ACM) anunció que el Premio Turing 2024 —considerado el “Nobel de la Computación”— sería otorgado a Andrew G. Barto y Richard S. Sutton “por desarrollar los fundamentos conceptuales y algorítmicos del aprendizaje por refuerzo” [1]. Durante casi cinco décadas, ambos investigadores construyeron las bases de una disciplina que, a partir de ideas tomadas de la psicología animal y la teoría del control óptimo, hoy es protagonista central de los avances más espectaculares de la inteligencia artificial: desde AlphaGo y AlphaZero, pasando por robots que juegan tenis de mesa al nivel de jugadores profesionales, hasta el ajuste fino de modelos de lenguaje como ChatGPT mediante RLHF [2]. Este artículo presenta el aprendizaje por refuerzo, las contribuciones específicas de Sutton y Barto, y los principales hitos tecnológicos que su trabajo hizo posibles.

¿Qué es el aprendizaje por refuerzo?

El aprendizaje por refuerzo (en inglés, *reinforcement learning* o RL) es uno de los tres grandes paradigmas del aprendizaje de máquinas, junto al aprendizaje supervisado y al no supervisado. A diferencia del aprendizaje supervisado, en RL no se entrena al modelo con un conjunto de ejemplos y sus respuestas correctas. En lugar de eso, un **agente** interactúa con el **ambiente**: en cada instante observa el **estado** del ambiente, ejecuta una **acción**, y recibe una **recompensa** numérica que mide qué tan bueno fue lo que hizo. El objetivo del agente no es responder correctamente a una pregunta, sino aprender una **política** —es decir, una forma de elegir las acciones correctas para cada estado del ambiente— que maximice la suma esperada (futura) de recompensas a lo largo del tiempo, un concepto conocido como el **retorno** esperado (ver Figura 1).

Esta formulación, aparentemente simple, captura una clase enorme de problemas que comparten una característica común: la decisión secuencial bajo incertidumbre, donde las acciones de hoy condicionan las posibilidades de mañana. La inspiración biológica es directa. Cuando entrenamos a un perro con golosinas, cuando un niño aprende a caminar tropezándose o a andar en bicicleta cayéndose, seguimos el mismo esquema: probar, recibir retroalimentación, y ajustar el comportamiento. El psicólogo Edward Thorndike, a comienzos del siglo XX, formalizó esta idea con su “ley del efecto”: las acciones seguidas de consecuencias positivas tienden a repetirse, y las que producen consecuencias negativas tienden a evitarse [3]. El aprendizaje por refuerzo

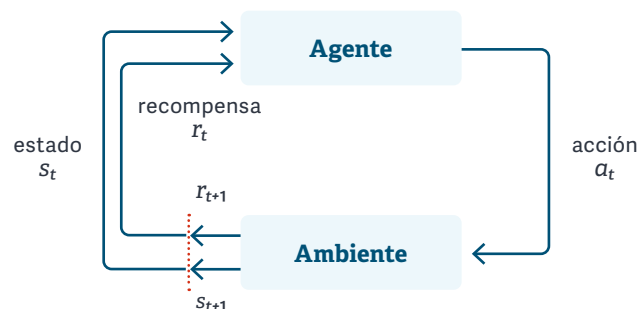


Figura 1 / Esquema del aprendizaje por refuerzo: el agente observa el estado s_t del ambiente, ejecuta una acción a_t , y recibe una recompensa r_{t+1} junto con el siguiente estado s_{t+1} . El objetivo es aprender una política $\pi(a/s)$ que maximice la recompensa acumulada esperada.

Cuando un niño aprende a caminar tropezándose o a andar en bicicleta cayéndose, seguimos el mismo esquema [que en reinforcement learning]: probar, recibir retroalimentación, y ajustar el comportamiento.

computacional, también llamado aprendizaje reforzado, es, en cierto sentido, una traducción matemática y algorítmica de este principio.

La diferencia del RL con el aprendizaje supervisado merece ser enfatizada. En el aprendizaje supervisado, el sistema recibe pares (entrada, respuesta correcta) y aprende una función que mapea las entradas en salidas para, por ejemplo, clasificar imágenes, predecir precios o traducir oraciones. En RL, en cambio, nadie le dice al agente cuál es la acción correcta; al ejecutar una acción, en un cierto estado del ambiente, el agente sólo recibe una señal escalar de recompensa, a menudo ruidosa y muchas veces *retardada*. A partir de esta recompensa debe aprender la política, es decir qué acciones elegir en cada situación/estado: la acción que realmente importó pudo haber ocurrido decenas o cientos de pasos antes de que se observe su consecuencia. Este es el famoso problema de *asignación de crédito temporal* (*temporal credit assignment problem*), uno de los desafíos centrales de RL.

¿Cuándo conviene usar RL? Típicamente, cuando se cumplen tres condiciones: (i) el problema es de toma de decisiones secuencial, con decisiones encadenadas; (ii) no se dispone de un conjunto de ejemplos etiquetados con la “respuesta correcta”, pero sí es posible interactuar con el ambiente (real o simulado) y obtener una señal de éxito o fracaso (recompensa); y (iii) se busca un comportamiento óptimo a largo plazo, no sólo una respuesta puntual. Robótica, control de procesos, juegos, gestión de recursos, vehículos autónomos, sistemas de recomendación, y trading financiero son ejemplos naturales de estas situaciones.

Las contribuciones de Sutton y Barto

Andrew Barto comenzó su carrera en la Universidad de Massachusetts Amherst en 1977. Richard Sutton llegó como estudiante de postgrado a ese mismo grupo en 1978 y obtuvo su doctorado bajo la supervisión de Barto en 1984,

con una tesis pionera sobre asignación de crédito temporal [4]. Lo que comenzó como una colaboración entre profesor y estudiante se transformó en una de las asociaciones científicas más fructíferas de la computación moderna, sostenida por más de cuarenta años.

A inicios de los ochenta, el aprendizaje por refuerzo era un área marginal, considerada por muchos como una curiosidad teórica sin mucha aplicabilidad. La inteligencia artificial dominante era la simbólica, basada en lógica y reglas; el aprendizaje conexionista comenzaba a renacer pero con foco en aprendizaje supervisado. Sutton y Barto, sin embargo, vieron en las ideas de la psicología animal y en los trabajos pioneros de Marvin Minsky, Arthur Samuel y A. Harry Klopff una vía distinta para construir agentes inteligentes. Sus contribuciones específicas pueden agruparse en tres grandes ejes.

Primero, los fundamentos conceptuales. Sutton y Barto fueron quienes formalizaron el problema de RL en los términos que hoy usamos universalmente: agente, ambiente, estado, acción, recompensa, política, función de valor. Establecieron además las conexiones profundas entre RL y otras tres disciplinas que estudiaban variantes del mismo problema sin saberlo: la programación dinámica (Bellman) en investigación de operaciones [5], el control óptimo en ingeniería, y el condicionamiento clásico e instrumental en psicología. Esta unificación no fue trivial, pero permitió importar herramientas matemáticas maduras —como las ecuaciones de Bellman y los procesos de decisión de Markov— al estudio del aprendizaje, dando rigor a un campo que hasta entonces se apoyaba en intuiciones.

Segundo, los algoritmos fundamentales. En 1983, Barto, Sutton y Anderson publicaron el trabajo seminal sobre la *arquitectura actor-crítico*, que separa la función de selección de acciones (el “actor”) de la función de evaluación del estado (el “crítico”) [6]. Esta arquitectura, aplicada al clásico problema de balancear un péndulo invertido, anticipó por décadas las técnicas que hoy entrenan modelos de lenguaje. En 1988, Sutton publicó “Learning to Predict by the Methods of Temporal Differences” [7], introduciendo formalmente el aprendizaje por diferencias temporales (*TD learning*) y el algoritmo $TD(\lambda)$. La idea es de una elegancia notable: en vez de esperar el resultado final de un episodio para corregir las predicciones, se ajustan las predicciones usando la diferencia entre la predicción actual y la siguiente. Es decir, el sistema aprende de sus propias predicciones futuras —*bootstrapping*—, lo que permite aprender de manera incremental y eficiente. A esto se sumaron contribuciones posteriores cruciales: la arquitectura Dyna (que integra aprendizaje, planificación y ejecución) [8], métodos de gradiente de política, el RL jerárquico, y los algoritmos de TD con gradiente.

Tercero, la divulgación y consolidación del campo. En 1998 Sutton y Barto publicaron “Reinforcement Learning: An Introduction”, el libro de texto que se convirtió rápidamente en la biblia indiscutida del área (ver Figura 2). Esta primera edición, junto a la segunda, de 2018 [9], expandida y actualizada, superan en conjunto las 95.000 citas en Google Scholar. El libro de Sutton y Barto no sólo enseñó RL a varias generaciones de investigadores; también fijó la notación, ordenó la taxonomía de los algoritmos, y conectó RL con neurociencia, psicología y control. Es difícil imaginar la explosión actual del área sin esa base pedagógica común.

Aplicaciones del aprendizaje por refuerzo

Como se mencionó anteriormente, el RL es una herramienta apropiada de aprendizaje para problemas donde un sistema/agente debe tomar decisiones secuenciales en ambientes inciertos. Entre los principales ámbitos de aplicación con estas características destacan:

Agentes y juegos. Los juegos —desde los clásicos de tablero hasta los videojuegos modernos— son un ambiente ideal de aplicación para el RL: reglas conocidas, recompensas bien definidas (ganar o perder), y posibilidad de simulación masiva. Es el dominio donde RL ha mostrado sus resultados más espectaculares y mediáticos.

Robótica. Aprender políticas de control para manipulación, locomoción, navegación y agarre. Aquí RL enfrenta dificultades adicionales: la interacción con el ambiente es costosa (cada intento/prueba toma tiempo y desgasta el hardware), los espacios de estados y acciones son continuos y de alta dimensión, y la transferencia de simulación a realidad (*sim-to-real*) es compleja de realizar.

Procesamiento de lenguaje natural. La revolución reciente en grandes modelos de lenguaje incorpora el RL como pieza clave del entrenamiento, no para la fase inicial de pre-entrenamiento, sino para *alinear* el comportamiento del modelo con las preferencias humanas.

Optimización industrial y de sistemas. Desde la regulación de centros de datos (Google reportó reducciones de hasta 40% en consumo energético de enfriamiento usando RL), hasta el ruteo de tráfico, la programación de plantas químicas, y el diseño automatizado de chips —área en que el propio equipo de Google usó RL para colocación de componentes en circuitos integrados.

Economía y finanzas. Estrategias de mercado, gestión de carteras, fijación dinámica de precios y subastas en línea



Figura 2 / Portada de la primera y segunda edición de Reinforcement Learning: An Introduction de Sutton y Barto (MIT Press, 1998 y 2018), considerado el texto canónico del área, con más de 95.000 citas.



Figura 3 / Portadas de revistas Nature en los que se presentan hitos importantes de la aplicación del aprendizaje por refuerzo: febrero de 2015, aplicación de RL en juegos de video Atari (izquierda), enero 2016, primera vez que un sistema de RL venció a un jugador profesional humano en el juego de Go (derecha).

son problemas naturalmente formulables como decisión secuencial bajo incertidumbre.

Salud. Diseño de protocolos de tratamiento personalizados, donde cada decisión clínica afecta el estado futuro del paciente; aquí, sin embargo, el desafío ético y de seguridad es central, porque no se puede experimentar libremente con pacientes reales.

Hitos y avances habilitados por RL

Si los fundamentos los pusieron Sutton y Barto en los ochenta y noventa, la explosión práctica del RL ocurrió en

El libro de Sutton y Barto no sólo enseñó reinforcement learning (RL) a varias generaciones de investigadores; también fijó la notación, ordenó la taxonomía de los algoritmos, y conectó RL con neurociencia, psicología y control.

los últimos quince años, gracias a la combinación de los algoritmos RL con redes neuronales profundas, paradigma conocido como *deep reinforcement learning* (DRL). La aplicación del DRL hizo posible los hitos siguientes:

De Atari a AlphaGo, AlphaZero y MuZero. Entre 2013 y 2015, DeepMind demostró que un agente de RL profundo podía aprender a jugar decenas de videojuegos de Atari utilizando únicamente la información visual de la pantalla como entrada, alcanzando o superando el nivel humano [7]. El salto cualitativo llegó en 2016, cuando AlphaGo derrotó al campeón mundial Lee Sedol en Go, un juego que se consideraba inalcanzable por su gigantesca complejidad combinatoria. Sin embargo, esta versión de AlphaGo aún dependía del conocimiento humano, ya que fue entrenada con millones de jugadas de expertos [8]. La versión siguiente, AlphaGo Zero (2017), aprendió a jugar Go partiendo desde cero, exclusivamente por self-play —jugando contra sí misma— y superó ampliamente a la versión original. AlphaZero (2018) generalizó el enfoque a ajedrez y shogi con el mismo algoritmo, sin conocimiento específico del dominio más allá de las reglas [12]. Más recientemente, MuZero aprendió a jugar incluso sin que se le entregara explícitamente el modelo del juego: lo descubre por sí mismo. El éxito de estos sistemas ha tenido impactos más allá del ámbito algorítmico, por ejemplo, están cambiando la forma en que los propios jugadores humanos de Go entienden el juego, y han sido destacados en portadas de revistas como Nature (ver Figura 3), en las que hasta ahora era impensado ver artículos sobre sistemas de aprendizaje de máquinas.

Póker y juegos con información incompleta. Mientras Go, ajedrez y shogi son juegos de información perfecta, el póker introduce incertidumbre sobre las cartas del rival. En 2019, Pluribus —desarrollado por la Universidad Carnegie Mellon y Facebook AI— derrotó a varios profesionales de élite en póker Texas Hold'em sin límite de seis jugadores, combinando *self-play* con técnicas de búsqueda y abstracción [13]. Anteriormente, Libratus había hecho lo propio en la variante de dos jugadores. Estos resultados son particularmente nota-

bles porque los juegos con información incompleta exigen razonar sobre las creencias (*beliefs*) del oponente y aleatorizar la propia estrategia para no ser predecible.

Robots que juegan tenis de mesa. Quizás ningún resultado reciente ilustra mejor el potencial del RL en robótica que el robot de tenis de mesa presentado por DeepMind en 2024: el primer sistema robótico entrenado mediante IA capaz de jugar competitivamente al nivel de un jugador humano amateur. El sistema combina aprendizaje por refuerzo en simulación, *fine-tuning* en hardware real y políticas jerárquicas de bajo y alto nivel [14]. En 2026, Ace, un robot aún más avanzado, reportó victorias contra jugadores profesionales en condiciones de competencia oficial [15]. El tenis de mesa es un banco de prueba ideal para la aplicación de RL en problemas de control en tiempo real: requiere percepción a alta velocidad, control fino y adaptación al adversario, todo en escalas de fracciones de segundo.

Locomoción y manipulación robóticas. Más allá de demostraciones espectaculares, el uso de DRL ha permitido a robots cuadrúpedos y humanoides aprender a caminar y correr en terrenos complejos, recuperarse de empujones, y manipular objetos delicados. Equipos como los de Boston Dynamics, ETH Zürich, NVIDIA y diversos laboratorios académicos han mostrado que el ciclo “entrenar masivamente en simulación usando DRL, transferir a la realidad” es hoy una metodología viable y reproducible. Para el campo de la robótica móvil, esto representa un cambio de paradigma respecto a los métodos de control clásicos.

Robots que juegan fútbol. En la liga de fútbol robótico RoboCup, participan desde hace décadas equipos conformados por robots autónomos. En los últimos años, varios equipos han incorporado RL para aprender habilidades específicas —patear, driblar, posicionarse— que son difíciles de programar a mano. Trabajos recientes de DeepMind con humanoides Robotis OP3, así como aproximaciones académicas en la liga SPL (*Standard Platform League*), han demostrado que RL permite cerrar la brecha entre el comportamiento simulado y el real.

Grandes modelos de lenguaje y RLHF. Posiblemente la aplicación de mayor impacto cultural reciente sea el uso de RL en el entrenamiento de modelos de lenguaje. La técnica conocida como *Reinforcement Learning from Human Feedback* (RLHF) fue introducida por OpenAI hacia 2017 [16] y popularizada masivamente con InstructGPT y ChatGPT [2]. La idea es la siguiente: tras el preentrenamiento con aprendizaje supervisado, el modelo se afina mediante un esquema que considera tres pasos: (i) ajuste fino supervisado usando entrenadores humanos que crean ejemplos de prompts (preguntas) y respuestas ideales; (ii) entrenamien-

to de un *modelo de recompensa* a partir de la evaluación de parte de humanos de las respuestas del modelo en base a utilidad, veracidad y seguridad; y (iii) optimización del modelo de lenguaje mediante un algoritmo de RL —típicamente PPO [17], utilizando el modelo de recompensa entrenado en el paso (ii). Es decir, los modelos que hoy usamos para conversar, programar y redactar fueron *alineados* con preferencias humanas mediante un descendiente directo de los algoritmos actor-crítico que Sutton y Barto introdujeron hace cuatro décadas. ChatGPT, Claude, Gemini y otros sistemas comerciales descansan en este paradigma.

Conclusión

El otorgamiento del Premio Turing 2024 a Sutton y Barto es un reconocimiento a su trabajo científico sistemático. Durante los años ochenta y noventa, el aprendizaje por refuerzo era visto como un nicho minoritario, especulativo, demasiado conectado con la psicología animal para ser tomado en serio por la corriente principal de la inteligencia artificial. Sutton y Barto, junto a un grupo reducido de colaboradores, sostuvieron la convicción de que un marco basado en interacción, recompensa y aprendizaje por ensayo y error era

no sólo biológicamente plausible sino computacionalmente potente. Construyeron los conceptos, los algoritmos y los textos necesarios para que el campo madurara.

Hoy, el RL se encuentra en el centro de los sistemas de inteligencia artificial más avanzados que existen: desde los agentes que dominan los juegos de estrategia, pasando por los robots que aprenden a realizar tareas complejas en el mundo físico, hasta los modelos de lenguaje que conversan con miles de millones de personas. Detrás de cada uno de estos sistemas hay, en algún lugar, una ecuación de Bellman, una regla de actualización TD, o una arquitectura actor-crítico cuyo origen puede rastrearse a un par de artículos publicados hace cuarenta años.

El mensaje de fondo trasciende el premio. Es una invitación al trabajo serio y sistemático, a profundizar en conexiones interdisciplinarias incluso cuando no están de moda, y a relevar el hecho de que los libros que recopilan sistemáticamente lo que hay que saber sobre un área son una forma de contribución científica de enorme alcance. Para quienes enseñamos inteligencia computacional y robótica, el caso de Sutton y Barto es un recordatorio doblemente útil: que los fundamentos importan, y que vale la pena enseñarlos bien. **B**

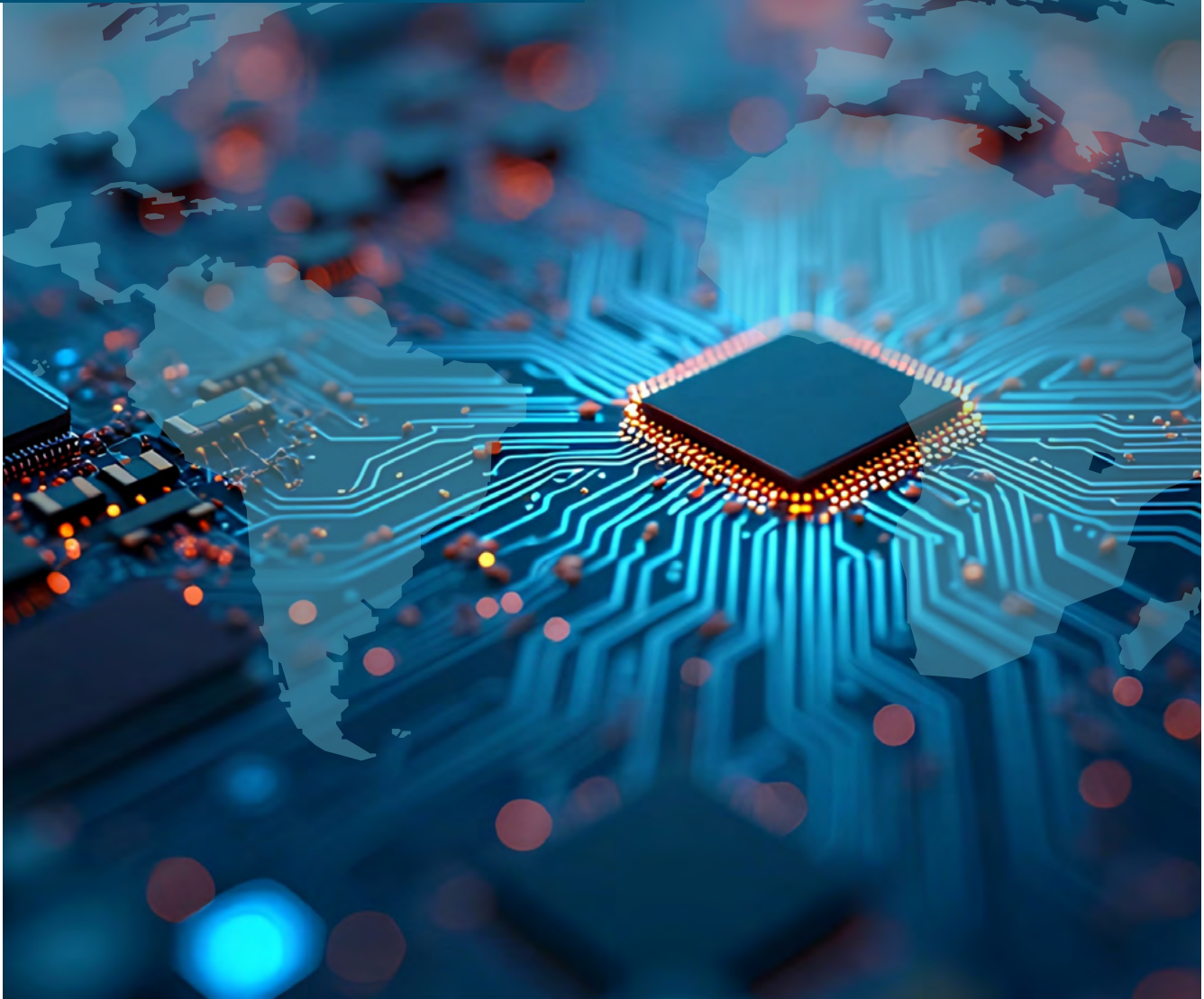
Referencias

- [1] ACM. "Andrew Barto and Richard Sutton are the recipients of the 2024 ACM A.M. Turing Award", 2025. <https://awards.acm.org/about/2024-turing>.
- [2] Ouyang, L., et al. "Training Language Models to Follow Instructions with Human Feedback" (InstructGPT). NeurIPS, 2022.
- [3] Thorndike, E. L. Animal Intelligence: Experimental Studies. Macmillan, 1911.
- [4] Sutton, R. S. "Temporal Credit Assignment in Reinforcement Learning". Tesis de Doctorado, University of Massachusetts Amherst, 1984.
- [5] Bellman, R. E. Dynamic Programming. Princeton University Press, 1957.
- [6] Barto, A. G., Sutton, R. S., y Anderson, C. W. "Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems". IEEE Transactions on Systems, Man, and Cybernetics, SMC-13(5):834–846, 1983.
- [7] Sutton, R. S. "Learning to Predict by the Methods of Temporal Differences". Machine Learning, 3:9–44, 1988.
- [8] Sutton, R. S. "Dyna, an integrated architecture for learning, planning, and reacting". SIGART Bull., vol. 2, no. 4, pp. 160–163, Jul. 1991.
- [9] Sutton, R. S., y Barto, A. G. Reinforcement Learning: An Introduction. Segunda edición. MIT Press, Cambridge MA, 2018.
- [10] Mnih, V., et al. "Human-Level Control through Deep Reinforcement Learning". Nature, 518:529–533, 2015.
- [11] Silver, D., et al. "Mastering the Game of Go with Deep Neural Networks and Tree Search". Nature, 529:484–489, 2016.
- [12] Silver, D., et al. "A General Reinforcement Learning Algorithm That Masters Chess, Shogi, and Go through Self-Play". Science, 362(6419):1140–1144, 2018.
- [13] Brown, N., y Sandholm, T. "Superhuman AI for Multiplayer Poker". Science, 365(6456):885–890, 2019.
- [14] D'Ambrosio, D. B., et al. "Achieving Human Level Competitive Robot Table Tennis". DeepMind. arXiv:2408.03906, 2024.
- [15] Dürr, P., El Gheche, M., Maeda, G.J. et al. "Outplaying elite table tennis players with an autonomous robot". Nature 652, 886–891 (2026).
- [16] Christiano, P., et al. "Deep Reinforcement Learning from Human Preferences". NeurIPS, 2017.
- [17] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., y Klimov, O. "Proximal Policy Optimization Algorithms". arXiv:1707.06347, 2017.



La base material de la
inteligencia artificial:

Chips, datacenters y geopolítica





Felipe Bravo-Márquez

PhD por la Universidad de Waikato, Nueva Zelanda. Profesor Asociado del Departamento de Ciencias de la Computación de la Universidad de Chile y Director de la Iniciativa de Datos e Inteligencia Artificial. Investigador Principal del Centro Nacional de Inteligencia Artificial e Investigador del Instituto Milenio Fundamentos de los Datos. Sus intereses incluyen NLP, aprendizaje automático, inteligencia artificial y recuperación de información.

✉ fbravo@dcc.uchile.cl



Claudio Gutiérrez

Ph.D. en Computer Science por la Wesleyan University, Estados Unidos, y Licenciado en Matemáticas por la Universidad de Chile. Profesor Titular del Departamento de Ciencias de la Computación de la Universidad de Chile e Investigador Senior del Instituto Milenio Fundamentos de los Datos (IMFD). Sus líneas de investigación son los fundamentos de la computación, la lógica aplicada a la computación, las bases de datos, la semántica de la Web y las máquinas sociales.

✉ cgutierr@dcc.uchile.cl



Luis Mateu

Doctor en Informática por la Université de Paris VI, Francia. Profesor de Jornada Parcial del Departamento Ciencias de la Computación de la Universidad de Chile. Sus áreas de interés son la arquitectura de computadores y los lenguajes de programación.

✉ lmateu@uchile.cl

Resumen / El reciente auge de la inteligencia artificial ha devuelto al hardware su papel central en la industria tecnológica. Entrenar y ejecutar modelos avanzados requiere enormes capacidades de computación, por lo que las GPUs y la cadena global de semiconductores se han convertido en un ámbito de competencia estratégica. Un reducido número de empresas, como NVIDIA, TSMC y ASML, concentra las capacidades clave para fabricar y operar chips de última generación.

Al mismo tiempo, la inteligencia artificial depende de una infraestructura material que requiere grandes cantidades de minerales, energía, agua y centros de datos. En este contexto, distintas regiones del mundo (incluida América Latina) comienzan a integrarse gradualmente en esta nueva geografía del cómputo, lo que abre oportunidades, pero también plantea interrogantes sobre cómo participar en la economía global de la inteligencia artificial.

Introducción

Durante gran parte de las últimas tres décadas, la industria tecnológica digital estuvo dominada fundamentalmente por la idea de que el valor real residía en las capas lógicas: el software y los datos. Este cambio se hizo patente con el desplazamiento de IBM (el icono clásico del hardware) por parte de Microsoft (el software) hace casi medio siglo. El poder pasó de quienes controlaban la infraestructura física a quienes controlaban el software y, posteriormente, las plataformas (Google). El hardware se convirtió en un bien de consumo intercambiable cuya producción podía delegarse, y quien controlaba las plataformas y las experiencias de usuario tenía el poder.

Sin embargo, la irrupción de la Inteligencia Artificial (IA) generativa y los Grandes Modelos de Lenguaje (LLMs por su sigla inglesa) ha roto esa tendencia. La IA es, por definición, una tecnología “hambrienta” de cómputo donde entrenar un modelo del estado del arte es una operación industrial que requiere billones de multiplicaciones de matrices que las CPUs tradicionales no pueden realizar con eficiencia [1]. Este fenómeno ha situado a las Unidades de Procesamiento



Nodo de fabricación	Aplicación principal	Estado estratégico
28 nm y superiores	Electrónica de consumo masivo (IoT, automotriz básica, electrodomésticos), chips “commodities” de bajo costo	Madura / Sin restricciones. Son los procesos más económicos en términos de costo fijo y suficientemente eficientes para aplicaciones que no requieren alta densidad ni velocidad.
14 nm – 10 nm	Principalmente memorias y algunos chips heredados o de reemplazo	Estable / Bajo escrutinio. La reducción de escala ofrece beneficios limitados en ciertas aplicaciones, por lo que estos nodos se mantienen por razones de costo y madurez tecnológica.
7 nm – 5 nm	GPUs de alto rendimiento (incluyendo IA), CPUs de notebooks y móviles de gama media alta	Vanguardia / Altamente restringida. Permiten alta densidad y eficiencia, aunque con costos crecientes. Se utilizan en chips grandes y en procesadores modernos que equilibran costo y desempeño.
3 nm – 2 nm	CPUs y sistemas integrados en un solo chip (SoC), que combinan CPU, GPU y otros componentes en un único circuito, típicos de notebooks y dispositivos móviles de última generación. De forma incipiente, comienzan a explorarse también en GPUs de nueva generación.	Frontera tecnológica / Reservados para diseños donde la eficiencia energética y la densidad justifican el alto costo. Su uso en chips muy grandes, como GPUs, sigue siendo limitado por desafíos de rendimiento de fabricación, aunque fabricantes como NVIDIA están comenzando a migrar progresivamente hacia estos nodos en sus arquitecturas más recientes. [6]

Tabla 1 / Estado de los nodos de fabricación de semiconductores (2025-2026).

to Gráfico (GPU), un tipo especial de hardware paralelo, en el centro de la competencia industrial y geopolítica. Empresas como NVIDIA han alcanzado valoraciones de mercado históricas, superando a gigantes del software como Microsoft, simbolizando la revalorización de la infraestructura física como una frontera estratégica en el siglo XXI.

Esto ha traído impensadas tensiones geopolíticas. El software estandarizado había comenzado a hacer transparentes las fronteras entre países. Pero hoy, para los Estados la capacidad de ejecutar modelos de IA en territorio propio ha pasado a ser una garantía, tanto de soberanía y seguridad nacional como de poderío económico e industrial. Dependier exclusivamente de proveedores de nube externos (*hyperscalers*) ubicados en otras jurisdicciones expone a los países a riesgos de costos incontrolables, restricciones de acceso por cambios en políticas de exportación y la pérdida de control sobre información estratégica [2].

En lo que sigue, abordamos estos temas, comenzando por explicar los detalles técnicos que sostienen este nuevo revival del hardware y sus relaciones con la geopolítica mundial.

Hardware de la IA y geopolítica

El escalamiento nanométrico.

Las GPU son procesadores paralelos que permiten acelerar uno de los principales cuellos de botella de la IA: el cómputo masivo en paralelo. Su potencia reside, en gran medida, en la densidad de transistores integrados en el silicio. Tradicionalmente, este avance se ha descrito mediante los llamados “nodos” de fabricación, expresados en nanómetros (nm).¹ Un nanómetro equivale a una millonésima parte de un milímetro (para referencia, un cabello humano tiene aproximadamente 70.000 nm de grosor).

¹ Sin embargo, estas cifras ya no corresponden a una dimensión física específica del transistor, sino que funcionan como etiquetas que sintetizan un conjunto de mejoras tecnológicas. En este sentido, los nombres comerciales de los procesos de fabricación, como N3E de TSMC [3] o 18A de Intel [4], no describen una escala en nanómetros, sino una generación tecnológica completa que abarca el tipo de transistor, los materiales, las reglas de diseño y las técnicas de litografía empleadas por cada fabricante.

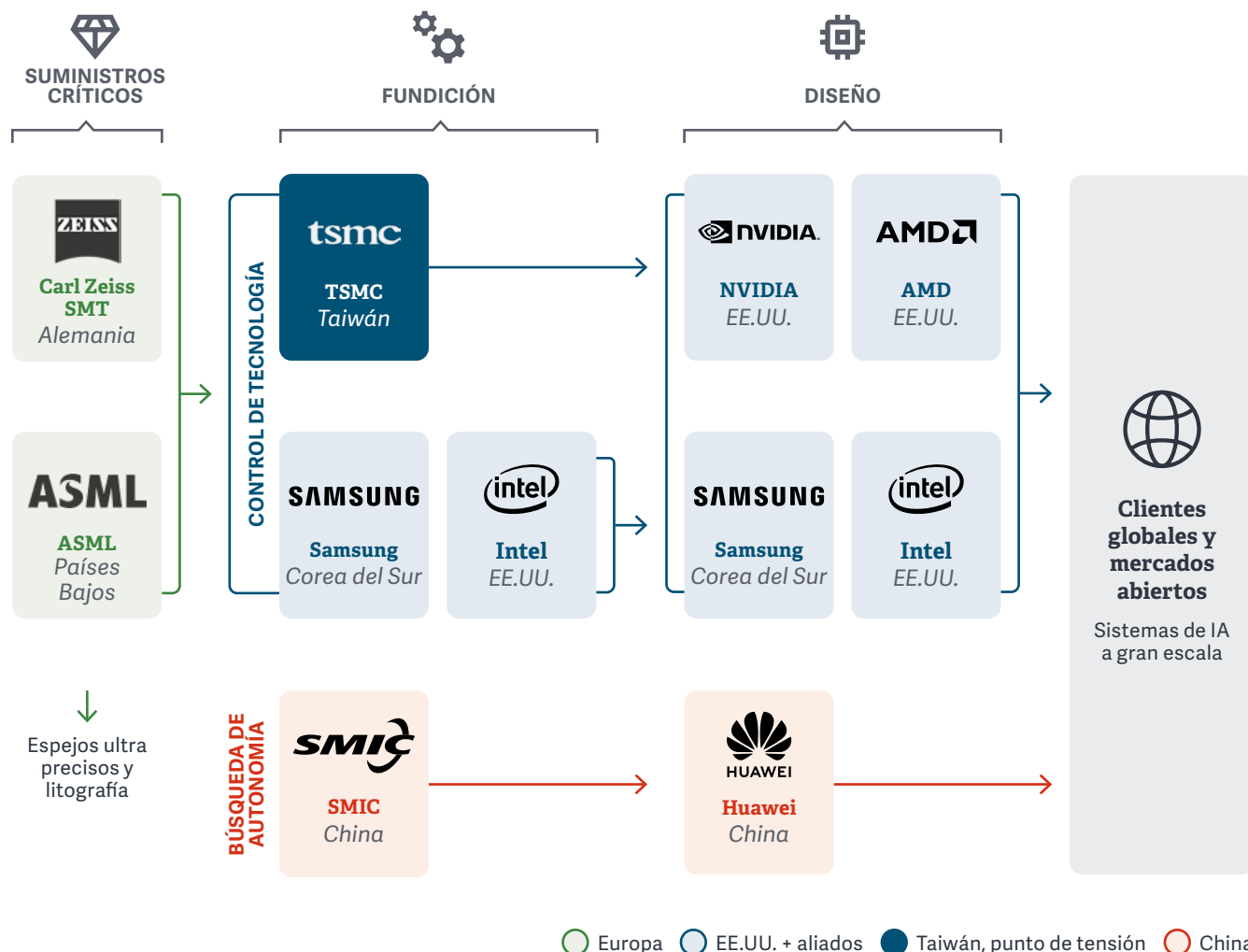


Figura 1 / La delicada cadena de suministro global de GPUs.

A partir de escalas cercanas a los 14 nm, las dimensiones críticas de los dispositivos comienzan a aproximarse a límites físicos y tecnológicos cada vez más exigentes. Este fenómeno se intensifica en generaciones posteriores y, hacia los 7 nm, se observa un punto de inflexión: el costo por transistor deja de disminuir de forma sistemática, rompiendo así una de las tendencias históricas más relevantes de la industria.²

Hoy en día, fabricar chips de última generación (como los de 3 nm) requiere un nivel de precisión extremadamente alto: algunas de sus partes son tan pequeñas que se acercan al tamaño de unos pocos átomos. Esto hace que di-

señarlos y producirlos sea cada vez más caro, porque las máquinas necesarias son muy complejas (como las herramientas litográficas —ver más adelante) y los diseños son mucho más difíciles de desarrollar. Como resultado, entrar en esta industria se ha vuelto muy difícil y costoso, lo que hace que sólo unas pocas empresas en el mundo —como TSMC o Samsung— puedan competir en este nivel tan avanzado [5].

En la Tabla 1 resumimos los distintos nodos de fabricación, sus aplicaciones principales y su relevancia estratégica. Asimismo, en la Figura 1 se presenta una visión esquemática de la cadena de suministro de GPUs.

² La famosa Ley de Moore. Hay que decir que la reducción de escala continúa siendo rentable gracias a mejoras en rendimiento y eficiencia energética por transistor.



Esta adopción basada exclusivamente en NVIDIA ha generado un “vendor lock-in” profundo: migrar a otro proveedor implicaría reescribir o adaptar años de código optimizado para CUDA y sus bibliotecas.

Litografía

El proceso de fabricación de un microchip comienza con la litografía, que es básicamente una técnica de “dibujo con luz”. Los chips electrónicos son discos muy delgados y perfectamente pulidos de silicio que sirven como base para fabricar chips electrónicos. Por eso les llaman “oleas de silicio”. Los patrones microscópicos que definen dónde irán los componentes del chip se graban (“dibujan”) en la oblea mediante la proyección de rayos ultravioleta a través de un patrón de molde. Cuando hablamos de nodos más avanzados (como los de 7, 5 o 3 nm), nos referimos a la capacidad de hacer esos dibujos cada vez más compactos. Cuanto más pequeño es el “nodo”, más transistores caben en el mismo espacio, lo que hace que el procesador sea más potente y consuma menos energía.

En la frontera actual de la tecnología, alcanzar estos niveles de miniaturización depende críticamente de la litografía ultravioleta extrema, conocida como EUV (siglas en inglés). Esta capacidad está altamente concentrada en una sola empresa, la holandesa ASML, cuyos equipos son, en la práctica, los únicos capaces de imprimir los patrones necesarios para fabricar chips de última generación.

Cada una de estas máquinas es una obra maestra de ingeniería que condensa décadas de inversión y desarrollo. Los modelos más recientes, conocidos como High NA EUV y desplegados progresivamente entre 2025 y 2026, tienen un costo cercano a los 370 millones de dólares por unidad [7]. Estas nuevas generaciones de equipos están pensadas para habilitar futuros nodos tecnológicos, como los procesos 14A en desarrollo por Intel, mientras que los nodos actualmente en despliegue industrial, como 18A, utilizan variantes más maduras de EUV.

Esta dependencia de un único proveedor ha creado un cuello de botella que trasciende lo comercial para volverse un asunto de soberanía nacional. El acceso a las máquinas de

ASML se ha transformado en una herramienta geopolítica: Estados Unidos y sus aliados limitan su venta a países como China para frenar su avance en inteligencia artificial y sistemas militares. Sin embargo, el monopolio de ASML encierra una paradoja: ellos mismos dependen de un tercero. Los espejos ultraprecisos necesarios para reflejar la luz ultravioleta dentro de estas máquinas son fabricados exclusivamente por la división SMT de Carl Zeiss (alemana) [8]. Actualmente, no existe otra entidad en el mundo capaz de fabricar esos espejos con la precisión requerida. Así, el eslabón más importante de la tecnología moderna depende, irónicamente, de un proveedor de espejos...

Automatización del diseño electrónico

La dependencia de HW-SW es muy estrecha. Antes de que un chip llegue a una fundición debe ser diseñado mediante software de Automatización de Diseño Electrónico (EDA), un mercado altamente concentrado dominado hoy por tres empresas: Synopsys, Cadence Design Systems y Siemens EDA.

Estas plataformas permiten diseñar circuitos con miles de millones de transistores y optimizar variables críticas como consumo energético, área y rendimiento. Sin ellas resulta prácticamente imposible desarrollar chips competitivos. Por lo mismo, el software EDA se ha transformado también en una herramienta de poder y disputa geopolítica.

Estados Unidos ha utilizado este control como instrumento de política tecnológica, restringiendo en distintos momentos la exportación de estas herramientas para ciertos países o para nodos tecnológicos avanzados (por ejemplo, inferiores a 14 nanómetros). Sin embargo, estas restricciones no han sido estáticas: se endurecen o se relajan según el contexto geopolítico [9].

Un dato estratégico interesante para nuestra región es que Synopsys mantiene oficinas en Santiago, donde ingenieros locales participan en el desarrollo de estas herramientas críticas. A diferencia de la fabricación física de chips, que requiere inversiones de decenas de miles de millones de dólares, el software EDA depende principalmente de capital humano altamente calificado. Esto abre una oportunidad para países que no poseen capacidad de fabricación avanzada, ya que les permite integrarse en una de las partes más intensivas en conocimiento de la cadena de valor de los semiconductores.

Fundiciones

Finalmente, una vez completado el diseño, este se envía a una fundición (“foundry”) para su materialización. El panorama de las fundiciones está marcado por una concentración

Fundición	País	Cuota de mercado en chips avanzados	Estado estratégico
TSMC	Taiwán	~90%	Líder absoluto; fabrica GPUs de NVIDIA, AMD y Apple
Samsung	Corea del Sur	Competidor clave	Fuerte en servicios de fundición y memorias HBM
Intel	Estados Unidos	Intentando recuperar terreno	Foco en servicios de fundición para el bloque occidental
SMIC	China	Limitada a tecnologías antiguas	Restringida por falta de máquinas EUV

Tabla 2 / Principales fundiciones de semiconductores avanzados.

extrema en el sudeste asiático (véase Tabla 2). Mientras TSMC domina ampliamente la producción de chips avanzados y Samsung Electronics se mantiene como su principal competidor, Intel intenta recuperar terreno con su estrategia de servicios de fundición. Por su parte, la fundición china SMIC enfrenta fuertes limitaciones tecnológicas debido a las restricciones de acceso a máquinas EUV, lo que la mantiene confinada principalmente a nodos de fabricación más antiguos [10]. Sin embargo, reportes recientes sugieren que la empresa ha logrado avances parciales utilizando técnicas de multipatrón con litografía DUV³, lo que evidencia cierta capacidad de adaptación tecnológica pese a las restricciones de exportación [11, 12].

La hegemonía de TSMC no se explica sólo por la disponibilidad de máquinas EUV, sino también por sus excepcionales niveles de rendimiento, es decir, el porcentaje de chips utilizables producidos en cada oblea de silicio. En nodos de 3 nm, TSMC ha logrado yields superiores al 70%, mientras que sus competidores enfrentan tasas significativamente menores. Esto hace que la producción de chips avanzados fuera de Taiwán sea, por ahora, mucho más costosa y arriesgada [13].

Esta concentración productiva convierte a Taiwán en un punto crítico para la economía global. A ello se suma un factor geopolítico ineludible: China reclama históricamente la soberanía sobre la isla de Taiwán y ha planteado en diversas ocasiones la posibilidad de una reunificación forzada. Un escenario de conflicto o interrupción significativa de las operaciones de TSMC tendría consecuencias sistémicas,

afectando no sólo la producción de GPUs y servidores de inteligencia artificial, sino también amplias cadenas industriales a nivel global. En términos de impacto económico, diversos analistas lo comparan con disrupciones mayores en mercados energéticos estratégicos, dada la centralidad de los semiconductores en la infraestructura tecnológica contemporánea. [14]

El ecosistema de hardware y software

Decíamos que la dependencia HW-SW es compleja y muy entrelazada. La dominancia de NVIDIA no se explica únicamente por sus chips, sino por su ecosistema de software propietario: CUDA (Compute Unified Device Architecture). Lanzado en 2007, se convirtió en el estándar de facto para la computación científica y el aprendizaje profundo. Durante casi dos décadas, universidades, empresas y proyectos open source han construido sus capas de software sobre CUDA. Esta adopción basada exclusivamente en NVIDIA ha generado un “*vendor lock-in*” (atadura al proveedor) profundo: migrar a otro proveedor implicaría reescribir o adaptar años de código optimizado para CUDA y sus bibliotecas.

Sin embargo, comienzan a aparecer alternativas. AMD impulsa ROCm como un stack abierto compatible con frameworks de IA modernos, mientras que Apple apuesta por su propio ecosistema basado en Metal para ejecutar modelos en los chips Apple Silicon [15]. En paralelo, Huawei desarrolla aceleradores y software propios para reducir la dependencia de CUDA en sus infraestructuras de IA.

3 DUV (Deep Ultraviolet) es una tecnología de litografía más antigua que utiliza luz ultravioleta profunda, típicamente con longitudes de onda de 193 nm, para definir patrones sobre obleas de silicio. Fue el estándar de la industria durante décadas previas a la introducción de la litografía EUV (Extreme Ultraviolet), que emplea longitudes de onda mucho más cortas y permite alcanzar nodos de fabricación más avanzados. Debido a sus limitaciones de resolución, la DUV requiere técnicas adicionales como multipatrón para aproximarse a escalas más pequeñas, lo que incrementa la complejidad y el costo del proceso.



[Latinoamérica] se consolida como un destino privilegiado para la instalación de data centers debido a la disponibilidad de recursos naturales, espacio físico y energía a costos competitivos.

Aunque la “brecha de CUDA” persiste, se aprecian intentos por diversificar el hardware que puede ejecutar modelos de aprendizaje profundo.

Un factor clave en este cambio es el rol creciente de frameworks como PyTorch y JAX. En lugar de programar directamente sobre el hardware, muchos desarrolladores trabajan con estas bibliotecas de alto nivel, que delegan la ejecución a distintos backends. Bajo este paradigma, estos frameworks actúan como una capa intermedia entre los modelos y los diversos tipos de aceleradores. Al mismo tiempo, muchos chips modernos incorporan NPUs (Neural Processing Units), unidades especializadas para ejecutar redes neuronales de forma eficiente. A diferencia de las GPUs, que están diseñadas para una amplia gama de operaciones paralelas, las NPUs están optimizadas para cargas de trabajo específicas de aprendizaje profundo, lo que en muchos casos permite una mayor eficiencia energética y una mejor relación costo–rendimiento en tareas de inferencia, especialmente en dispositivos personales. Por ejemplo, los procesadores de la familia Apple M-series, utilizados en dispositivos como el Mac mini, integran en un mismo chip CPU, GPU y una NPU denominada Neural Engine, lo que permite ejecutar modelos de IA directamente en el dispositivo. Gracias a estas arquitecturas integradas, comunidades de desarrolladores ya experimentan con la ejecución de LLMs en hardware personal, como Mac minis o PCs con GPUs, NPUs o aceleradores dedicados. En conjunto, estas tendencias apuntan hacia un ecosistema más diverso, donde la computación de IA se distribuye entre nubes, centros de datos y dispositivos personales.

Comunicación entre GPUs

Las cargas de trabajo de IA moderna no se ejecutan en una sola GPU, sino en clústeres de miles de ellas que deben intercambiar datos de forma casi instantánea. Las tecnologías de interconexión son, por tanto, el sistema circulatorio de los centros de datos de IA. NVIDIA controla estándares propietarios como NVLink e InfiniBand, optimizados para su propio hardware, lo que refuerza la dependencia de su ecosistema

completo. Frente a esto, la industria está impulsando estándares abiertos como Ethernet y nuevas iniciativas como Ultra Accelerator Link (UALink) para permitir que chips de diferentes proveedores trabajen juntos con baja latencia. La decisión de un Estado sobre qué arquitectura de red adoptar para sus supercomputadores nacionales definirá si queda atado a un proveedor específico o si mantiene la flexibilidad para integrar tecnologías diversas en el futuro.

La inserción estratégica de América Latina en la geopolítica de la inteligencia artificial

Brecha estructural, asimetría del “stack” y funciones

América Latina presenta en el terreno de la IA una brecha que no es meramente tecnológica sino *estructural* [17]. Esta asimetría se manifiesta en la ausencia de capacidades para el diseño de semiconductores (circuitos integrados) y un control nulo sobre las capas profundas del *stack* tecnológico (desde el silicio hasta los modelos fundacionales).

En este sentido, América Latina no es percibida por las grandes *big tech* como un polo de innovación autónomo, sino como un espacio de recursos que cumple tres funciones críticas para el sistema-mundo digital.

Primero, un *territorio de despliegue y extractivismo energético*: La región se consolida como un destino privilegiado para la instalación de *hyperscale data centers* debido a la disponibilidad de recursos naturales, espacio físico y energía a costos competitivos. Países como Chile, Brasil y México funcionan como “enclaves tecnológicos” donde la infraestructura física se beneficia de recursos naturales y marcos regulatorios favorables, similar a la dinámica histórica de la industria minera [18].

Segundo, como nodo en la *red global de cómputo*: La infraestructura local actúa como un punto de presencia (PoP) esencial para reducir la latencia en la red global de las *big tech*. Esta distribución geográfica permite la expansión de servicios de IA en tiempo real, optimizando el rendimiento de modelos generativos y servicios en la nube a escala continental.

Tercero, como espacio de *disputa geopolítica “blanda”*: En el marco de la “Guerra Fría Tecnológica” entre Estados Unidos y China, la región representa un terreno de disputa por la hegemonía de estándares y acceso a recursos. El caso del “cable chino” en Chile es un muy buen ejemplo de esta disputa [19]. La elección de proveedores de infraestructura



(e.g., la adopción de tecnología 5G de Huawei versus alternativas occidentales) define no sólo la dependencia tecnológica a largo plazo, sino también la alineación de los marcos regulatorios y de gobernanza de datos.

Geopolíticamente, la competencia regional no se centra necesariamente en el liderazgo de la frontera tecnológica, sino en las condiciones de su integración subordinada. Es importante entender que la presencia de infraestructura física en territorio nacional no equivale directamente a soberanía tecnológica.

El rol asignado por los actores globales

La estrategia de las empresas líderes hacia la región varía según su posición en la cadena de valor. Los dos principales grupos son los proveedores de HW y los de plataformas (*hyperscalers*).

Para los proveedores de hardware (NVIDIA, AMD, Intel), esto es, las firmas de semiconductores, América Latina es estrictamente un mercado de consumo masivo y un nodo de distribución de potencia de cómputo (vía *Cloud*). No existe una intención manifiesta de descentralizar el diseño o la manufactura hacia la región. La relación es de proveedor-cliente: se busca capturar el crecimiento de la demanda de GPUs para centros de datos locales sin transferir las capacidades de innovación que definen el valor agregado del sector.

Para las plataformas e *hyperscalers* (Google, Microsoft, AWS), la región es estratégica y multidimensional. Por un lado, la instalación de “Regiones de Cloud” busca la cobertura total del

mercado latinoamericano. Por otro, al integrar a gobiernos, *startups* y grandes empresas en sus ecosistemas de nube, aseguran que el desarrollo de IA local se construya exclusivamente sobre sus *stacks* propietarios (enclaustramiento tecnológico o *vendor lock-in*). Finalmente, la cercanía física / geográfica permite incidir en las políticas públicas locales, optimizar costos operativos y asegurar que la regulación nacional sea compatible con sus modelos de negocio globales.

En resumen, América Latina se encuentra en una encrucijada donde la abundancia de infraestructura física podría hacer perder de vista una creciente dependencia cognitiva y técnica. La región compite por ser el soporte material de la IA global, mientras el control intelectual y económico de la misma permanece concentrado fuera de sus fronteras.

El caso de Chile

En el escenario regional, Chile ha adoptado una postura relativamente proactiva en el desarrollo de infraestructura y capacidades para inteligencia artificial, articulada en torno a tres tendencias: habilitación regulatoria, fortalecimiento del supercómputo y descentralización incipiente [20, 21].

Por una parte, el país ha avanzado en generar condiciones para la instalación de centros de datos, apoyándose en su matriz energética con alta penetración de renovables, lo que lo posiciona como un nodo atractivo para cómputo intensivo a nivel regional [22].

Por otra, se observa un fortalecimiento de capacidades nacionales de supercómputo a través de iniciativas académicas y



público-privadas, como el NLHPC en la Universidad de Chile o desarrollos impulsados por la Pontificia Universidad Católica de Valparaíso en conjunto con actores tecnológicos. Estas iniciativas combinan capacidades para entrenamiento de modelos y provisión de servicios de inferencia, con foco en habilitar tanto investigación como adopción productiva [23].

Finalmente, comienzan a emerger esfuerzos de descentralización, como el proyecto de supercómputo de la Universidad de Tarapacá en el norte del país, que apunta a desconcentrar capacidades y fortalecer la resiliencia territorial de la infraestructura digital [24].

El desafío hacia adelante no es sólo expandir esta base material, sino articularla con el ecosistema productivo y científico. En particular, será clave asegurar transferencia tecnológica efectiva y desarrollo de capacidades locales, evitando una dinámica de “extractivismo digital” donde la infraestructura se instala localmente, pero el valor es capturado externamente [25].

Conclusiones

A partir del análisis presentado, es posible extraer algunas conclusiones estratégicas sobre la base material de la inteligencia artificial y su impacto en el tablero geopolítico global, con especial énfasis en América Latina y Chile.

En primer lugar, vemos un retorno a la materialidad y los temas de soberanía tecnológica. El boom de las aplicaciones de IA ha generado la ilusión de que su valor reside exclusivamente en el software. Hoy, como hemos visto, la soberanía nacional y el poderío económico dependen de la capacidad física de cómputo y del acceso a una cadena de suministro de semiconductores extremadamente concentrada. Esto levanta amenazas como la dependencia de proveedores externos de nube (*hyperscalers*) y de hardware específico, lo que genera riesgos de seguridad, costos incontrolables y vulnerabilidad ante cambios en políticas de exportación de potencias extranjeras. En este sentido, los Estados que invierten en infraestructura propia, como supercomputadores y data centers nacionales, infraestructura de redes y comunicaciones, pueden garantizar una mayor autonomía en la ejecución de modelos estratégicos o de alto poder.

Una segunda conclusión general es la vulnerabilidad del monopolio global —asunto que es buena noticia para balancear concentración única de poder. Hemos visto que el ecosistema de la IA descansa sobre “cuellos de botella” críticos como empresas como ASML (litografía), TSMC (fundición) y NVIDIA (diseño y software CUDA). Sin embargo, permanecen riesgos como la concentración de la producción avanzada (nodos de

3nm a 2 nm en Taiwán representan un peligro para la estabilidad global ante posibles conflictos geopolíticos en la zona). También hay riesgo de dependencia de sistemas, en particular hoy el ecosistema CUDA de NVIDIA que ha creado un *vendor lock-in* que obliga a la industria a seguir utilizando su hardware, dificultando la migración a alternativas más abiertas.

¿Cómo se posiciona América Latina en este escenario? La región se encuentra en una encrucijada donde corre el riesgo de ser meramente un soporte material para la IA de otros países y de las Big Tech. Actualmente América Latina actúa principalmente como territorio de despliegue para *data centers* (aprovechando energía y agua) y como nodo para reducir la latencia de servicios extranjeros. Se genera entonces un riesgo de “extractivismo digital”: el peligro de que los países financien la infraestructura física y energética, mientras que el valor intelectual y económico es capturado por plataformas globales (algo que la industria tradicional latinoamericana conoce bien...).

Para finalizar este análisis, sugerimos algunos caminos para la región y para Chile. Para evitar una “integración subordinada”, queremos proponer para la discusión las siguientes líneas de acción:

1. Fomentar el capital humano en capas lógicas, esto es, dado que fabricar chips avanzados es prohibitivamente caro, países como Chile pueden integrarse al apoyo en el diseño (por ej. mediante software EDA como las oficinas de Synopsys en Santiago), que requiere alta calificación técnica pero menor inversión física.
2. Es crucial para países o economías de menor volumen el impulso a la adopción de estándares abiertos. Esto significa impulsar el uso de arquitecturas como Ethernet o UALink y *frameworks* como PyTorch o JAX para romper monopolios como el de CUDA, y así permitir una infraestructura de hardware más diversa, flexible y accesible.
3. En el área de soberanía de datos y energía, es muy importante condicionar la instalación de grandes centros de datos a la transferencia tecnológica efectiva, en particular a la participación de personal local no sólo en tareas de servicios, y al cumplimiento de estrictos criterios de sostenibilidad ambiental, pues el medio ambiente es el recurso por el cual las grandes big tech se interesan en nuestros países hoy.
4. Finalmente, dada nuestras geografías, es muy relevante la descentralización, lo que significa llevar la capacidad de cómputo fuera de las capitales y las grandes ciudades, para aumentar la resiliencia digital y el desarrollo de los talentos y las capacidades locales. **B**

Referencias

- [1] Miller, C. (2022). *Chip war: The fight for the world's most critical technology*. Simon and Schuster.
- [2] Le Monde Diplomatique. (2021). *Geopolítica de los chips*. <https://www.lemondediplomatique.cl/2021/08/geopolitica-de-los-chips.html>.
- [3] Taiwan Semiconductor Manufacturing Company. *3nm technology*. https://www.tsmc.com/english/dedicatedFoundry/technology/logic/l_3nm.
- [4] Intel Corporation. (2023). *Semiconductor manufacturing process technologies*. <https://www.intel.la/content/www/xl/es/foundry/process.html>.
- [5] PatentPC. (2025). *The future of Moore's Law: Are we nearing the limit?* <https://patentpc.com/blog/the-future-of-moores-law-are-we-nearing-the-limit-latest-semiconductor-trends>.
- [6] Garreffa, A. (2025, November 9). *NVIDIA's unstoppable AI chip demand has TSMC boosting 3nm production by a huge 50%*. Tweak-Town. <https://www.tweaktown.com/news/108769/nvidias-unstoppable-ai-chip-demand-has-tsmc-boosting-3nm-production-by-a-huge-50-percent/index.html>.
- [7] IO+. (2026). *China's all-or-nothing race for chip independence*. <https://ioplus.nl/en/posts/chinas-all-or-nothing-race-for-chip-independence>.
- [8] Reuters. (2016, November 3). *ASML buys 24.9 percent stake in Carl Zeiss subsidiary SMT for 1 billion euros*. <https://www.reuters.com/article/world/europe/asml-buys-249-percent-stake-in-carl-zeiss-subsiary-smt-for-1-billion-euros-idUSKBN12YOKA/>.
- [9] Freifeld, K., & Misra, S. (2025, July 3). *As trade war truce with China holds, US lifts curbs for chip design software and ethane*. Reuters. <https://www.reuters.com/world/china/siemens-says-us-has-lifted-chip-software-curbs-china-bloomberg-news-reports-2025-07-03/>.
- [10] Techzine. (2025). *Calls for a 'Chinese ASML' are growing louder, but it is still a long way off*. <https://www.techzine.eu/news/infrastructure/139303/calls-for-a-chinese-asml-are-growing-louder-but-it-is-still-a-long-way-off/>.
- [11] TechInsights. (2025, December 11). *SMIC N+3 confirmed: Kirin 9030 analysis reveals how close SMIC is to 5nm*. <https://www.techinsights.com/blog/smic-n3-confirmed-kirin-9030-analysis-reveals-how-close-smic-5nm#register>.
- [12] Kr-Asia. (2026). *China aims for fivefold increase in advanced chip output to meet AI demand*. <https://kr-asia.com/china-aims-for-fivefold-increase-in-advanced-chip-output-to-meet-ai-demand>.
- [13] Tom's Hardware. (2026). *China's top chip execs admit fragmentation is undermining the country's ASML alternative*. <https://www.tomshardware.com/tech-industry/semiconductors/chinas-top-chip-execs-admit-fragmentation-is-undermining-the-countrys-asml-alternative>.
- [14] Center for Strategic and International Studies. (2022, October 11). *Why Taiwan matters: From an economic perspective*. <https://www.csis.org/analysis/why-taiwan-matters-economic-perspective>.
- [15] StrategyLens. (2026). *NVIDIA's CUDA vs AMD's ROCm: software ecosystems powering generative AI*. <https://strategytransforml.co/category/gpu-and-high-performance-compute-chips/article/nvidia-cuda-vs-amd-rocm-software-ecosystems-generative-ai>.
- [16] Jung, J., & Katz, R. L. (2026). *Impacto económico de la inteligencia artificial en América Latina: transformación tecnológica y reza-go en materia de inversión y capacidades laborales*. Comisión Económica para América Latina y el Caribe (CEPAL). <https://www.cepal.org/es/publicaciones/81909-impacto-economico-la-inteligencia-artificial-america-latina-transformacion>.
- [17] Lehdonvirta, V. (2022). *Cloud Empires: How Digital Platforms Are Overtaking the State and How We Can Regain Control*. MIT Press.
- [18] BBC News Mundo. (2026, marzo 4). *El cable de fibra óptica chino que rompió la transición de gobierno entre Boric y Kast en Chile (y por el que EE.UU. sancionó a funcionarios del país)*. <https://www.bbc.com/mundo/articles/c8d554v943lo>.
- [19] Barros, A. (2025, 3 de agosto). *Plan de Data Centers, aciertos y temas pendientes*. AlejandroBarros.com. <https://www.alejandrobarrros.com/plan-de-data-centers-aciertos-y-temas-pendientes/>.
- [20] Gobierno de Chile. (2024). *Plan Nacional de Data Centers 2024–2030*. Ministerio de Ciencia. https://minciencia.gob.cl/uploads/filer_public/95/6b/956b8c9f-d937-4b4d-8f6c-a871495a52ff/plan_nacional_de_data_centers_pdata.pdf.
- [21] DLA Piper. (2025). *Chile implements its national data center plan (2024–2030)*. <https://www.dlapiper.com/es-cl/insights/publications/2025/05/chile-implements-its-national-data-center-plan>.
- [22] Inria Chile. (2025, 26 de junio). *Inria Chile refuerza capacidad nacional de supercómputo para IA*. <https://inria.cl/es/inria-chile-refuerza-capacidad-nacional-de-supercomputo-para-ia-aliado-clave-de-los-dos-centros>.
- [23] El País. (2026, enero 19). *La Universidad de Tarapacá proyecta en Arica el mayor centro de supercómputo de Chile*. <https://elpais.com/chile/branded/2026-01-19/la-universidad-de-tarapaca-proyecta-en-arica-el-mayor-centro-de-supercomputo-de-chile.html>.
- [24] Effoduh, J. O. (2025). *Digital colonialism and the role of local intermediaries*. Cambridge University Press.



LatamGPT:

Construyendo IA desde
el sur global



LatamGPT Team

Centro Nacional de Inteligencia Artificial (CENIA), Chile.
Investigación y desarrollo de
modelos de lenguaje para América
Latina.

 latamgpt.org

Resumen / Los grandes modelos de lenguaje que hoy dominan la inteligencia artificial fueron entrenados mayoritariamente con texto en inglés. A modo de ejemplo, en los modelos con información pública, América Latina, con el 8,8% de la población mundial y seis siglos de producción escrita principalmente en español y portugués, representa menos del 2% de los datos usados para entrenar estos sistemas. LatamGPT surge para aportar a cerrar esa brecha: es un modelo de 70 mil millones de parámetros, entrenado sobre 300 mil millones de tokens provenientes de corpus latinoamericanos. Estos corpus fueron aportados por 79 instituciones en 15 países de la región, dentro de un corpus más amplio que cubre 20 países latinoamericanos. Su objetivo es contribuir a que América Latina deje de ser sólo consumidora de IA y fortalezca sus propias capacidades para desarrollarla.

El problema: los modelos de lenguaje no hablan español latinoamericano

Los grandes modelos de lenguaje aprenden del texto con el que son entrenados. Si ese texto proviene mayoritariamente de páginas web en inglés, de Wikipedia anglosajona y de libros digitalizados de bibliotecas norteamericanas y europeas, el modelo representará el mundo desde esa perspectiva. La principal fuente de datos para entrenar la mayoría de los modelos actuales es Common Crawl, una enorme colección de páginas web de Internet donde el español latinoamericano representa una fracción marginal [1].

América Latina representa el 6,6% del PIB mundial y el 8,8% de la población del planeta, pero recibe apenas el 1,12% de la inversión global en inteligencia artificial [2]. Esta brecha de inversión se traduce en una brecha de representación: los modelos que millones de latinoamericanos usan hoy fueron diseñados para otros contextos, con otros referentes culturales y otro español. El problema no es sólo lingüísti-

América Latina representa el 6,6% del PIB mundial y el 8,8% de la población del planeta, pero recibe apenas el 1,12% de la inversión global en inteligencia artificial.

co sino también contextual: ese conocimiento no se puede inyectar traduciendo textos, sino que debe surgir de haber sido entrenado sobre fuentes auténticamente regionales.

LatamGPT nace para aportar a cerrar esa brecha. Partiendo de Llama 3.1 70B [3], continuamos su entrenamiento sobre 300 mil millones de tokens provenientes de corpus que cubren 20 países latinoamericanos, con aportes institucionales de 79 organizaciones en 15 países de la región. El resultado es el mayor modelo actual de lenguaje abierto entrenado con un foco en datos sobre América Latina.

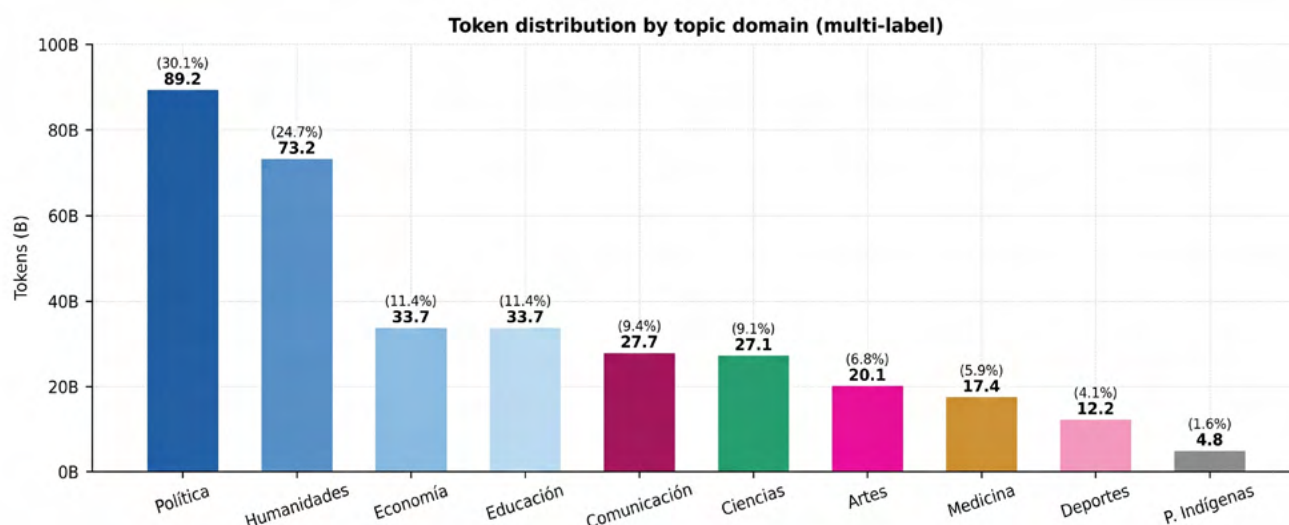


Figura 1 / Distribución por tópicos de tokens del dataset de preentrenamiento de LatamGPT.

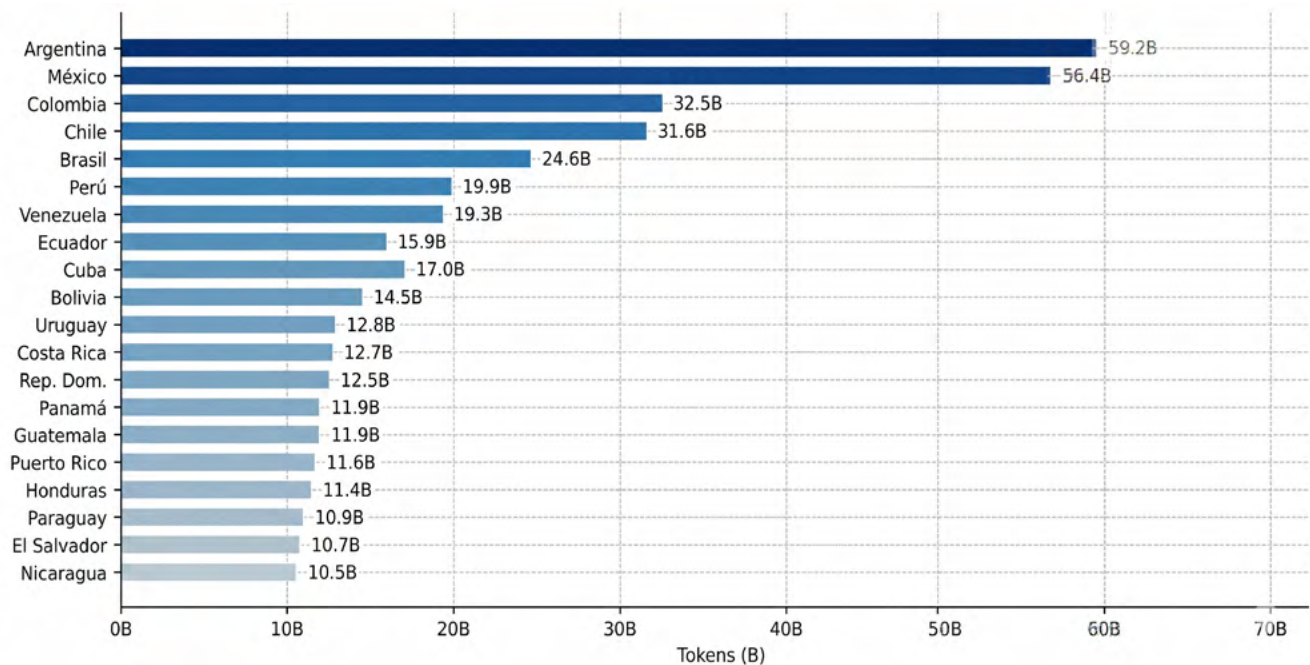


Figura 2 / Distribución de tokens asociados a los 20 países cubiertos por el corpus de LatamGPT. La clasificación geográfica es multi-etiqueta, por lo que un mismo documento puede estar asociado a más de un país y las cantidades no deben interpretarse como una partición excluyente.

Construyendo el corpus: 300 mil millones de tokens desde 79 instituciones

El corazón de cualquier modelo de lenguaje es su corpus de entrenamiento. Para LatamGPT, construir ese corpus fue una tarea que duró más de un año e involucró acuerdos formales de colaboración con 79 instituciones en 15 países de la región.

El corpus final suma 300 mil millones de tokens, unidad de medida que equivale aproximadamente a una sílaba o palabra corta. De ese total, 76,1 mil millones de tokens, casi el 20%, provienen directamente de instituciones: universidades, archivos nacionales, bibliotecas públicas, organismos gubernamentales y organizaciones de la sociedad civil que firmaron acuerdos de transferencia de datos con CENIA. El 80% restante proviene de Common Crawl, pero filtrado y seleccionado específicamente para representar el español y el portugués latinoamericanos.

La dimensión institucional es lo que distingue a LatamGPT de una simple selección del corpus web. Documentos jurídicos, producción académica regional, registros administrativos públicos, publicaciones de prensa histórica y literatura digitalizada de países con escasa presencia en Internet son fuentes que no existen en Common Crawl. Obte-

nerlas requirió establecer una amplia red de colaboración con instituciones por toda la región, firmar memorandos de entendimiento, establecer protocolos de privacidad, y en muchos casos, colaborar en la digitalización de materiales impresos. La Figura 1 muestra la distribución de temas resultante de este esfuerzo.

Esta estrategia de recolección de datos tiene consecuencias directas sobre el tipo de conocimiento que el modelo puede adquirir. A diferencia de modelos entrenados exclusivamente sobre texto web, LatamGPT ha sido expuesto a registros formales del derecho latinoamericano, a la producción científica de universidades de la región en sus idiomas originales, y a fuentes primarias de la historia de nuestros países. Eso no garantiza que el modelo sea perfecto, pero sí que conozca el mundo latinoamericano desde adentro.

La distribución geográfica del corpus refleja la desigualdad digital de la región (ver Figura 2). Bajo una clasificación geográfica multi-etiqueta, México y Argentina aparecen entre los países con mayor presencia relativa; Chile y Brasil también muestran una presencia significativa. Como un mismo documento puede estar asociado a más de un país, estas cifras no deben leerse como porcentajes excluyentes del corpus total. Los países más pequeños de Centroamérica y el Caribe tienen una presencia significativamente menor,

Con una tasa [de aprendizaje] conservadora, [...] el modelo era capaz de retener casi todo su conocimiento original e incorporaba el conocimiento latinoamericano de manera estable.

lo que indica que la brecha de representación dentro de América Latina es tan importante como la brecha entre la región y el mundo de habla inglesa.

El proceso de preparación del corpus incluyó etapas de limpieza, eliminación de duplicados, filtrado de información personal y clasificación automática de los documentos por país, idioma y dominio temático. Cada uno de estos pasos está documentado públicamente para que otras iniciativas regionales puedan replicarlo.

Preentrenamiento: aprender sin olvidar

Entrenar un modelo de lenguaje desde cero requiere cantidades de cómputo que están fuera del alcance de la mayoría de las instituciones académicas del mundo. Para LatamGPT, optamos por un enfoque diferente. Tomamos un modelo ya entrenado, el Llama 3.1 70B de Meta [3], y continuamos su entrenamiento sobre datos latinoamericanos. Esta estrategia se llama preentrenamiento continuado (*Continual Pre-Training* o CPT, por sus siglas en inglés) [4].

La motivación es exclusivamente de eficiencia. Aunque el proceso parte de un modelo existente, no corresponde a un simple *fine-tuning*: en términos de capacidades técnicas, infraestructura y desafíos de ingeniería, su complejidad es comparable a la de entrenar un modelo desde cero. Llama 3.1 ya aprendió gramática, razonamiento general, matemáticas y conocimiento del mundo durante un entrenamiento masivo. LatamGPT no necesita reaprender todo eso; sólo necesita adquirir el conocimiento específico de América Latina que Llama no tiene. Es como pedirle a un académico con amplísima formación general que realice una residencia intensiva en cultura y conocimiento latinoamericano.

El problema técnico central de un enfoque CPT es lo que en IA se llama olvido catastrófico. Cuando un modelo aprende nueva información de manera agresiva, puede degradar el conocimiento que ya tenía. Imaginemos un estudiante que, en su esfuerzo por memorizar los presidentes de todos los

países latinoamericanos, empieza a olvidar o confundir otros conceptos relevantes de la historia.

El parámetro más crítico para controlar este balance es la tasa de aprendizaje, que determina la velocidad con que el modelo ajusta sus parámetros internos frente a cada nuevo dato. Con una tasa demasiado alta, el modelo era capaz de absorber conocimiento latinoamericano muy rápidamente, pero pagaba un precio enorme. Su desempeño en pruebas generales de conocimiento en inglés caía hasta 23 puntos porcentuales respecto al punto de partida. Era un modelo más latinoamericano pero al mismo tiempo perdía capacidades previas. Con una tasa más conservadora, en cambio, el modelo era capaz de retener casi todo su conocimiento original e incorporaba el conocimiento latinoamericano de manera estable. Ese fue el punto óptimo que adoptamos para LatamGPT.

Para evaluar los resultados, usamos dos benchmarks diseñados específicamente para la realidad latinoamericana. Trueque [5] mide conocimiento factual sobre cultura, historia y geografía de América Latina en español; Choclo [6] evalúa conocimiento regional más específico sobre los 20 países cubiertos. A diferencia de MMLU o HellaSwag, que evalúan conocimiento general en inglés [7, 8], estos benchmarks fueron contruidos con preguntas sobre presidentes, escritores, platos típicos, accidentes geográficos y eventos históricos de la región, en el idioma en que esa información existe de manera natural.

Los resultados muestran que, en ambos benchmarks, LatamGPT supera consistentemente al modelo base Llama 3.1 y a modelos de tamaño similar del estado del arte como DeepSeek 67B [9], Gemma 2 27B [10] y Qwen 2.5 72B [11]. En Trueque, CPT Fase 1 alcanza 0.53 sin ejemplos de referencia, 15 puntos sobre el baseline de Llama 3.1, mientras que CPT Fase 0 alcanza 0.60 con ejemplos, 9 puntos sobre el mismo baseline (ver Figura 3). En Choclo, CPT Fase 1 obtiene 0.33 sin ejemplos y 0.35 con ellos, mejoras absolutas de 9 y 7 puntos sobre Llama 3.1, respectivamente (ver Figura 4). Crucialmente, estas ganancias se obtienen sin degradar el desempeño en benchmarks generales. En MMLU y HellaSwag, LatamGPT Fase 1 con la tasa de aprendizaje conservadora se mantiene dentro de 0,2 puntos porcentuales del modelo base, lo que confirma que aprendió conocimiento latinoamericano sin olvidar lo que ya sabía.

Postentrenamiento: aprender a seguir instrucciones

Un modelo preentrenado adquiere mucho conocimiento pero le falta seguir instrucciones. Lo que aprendió en la

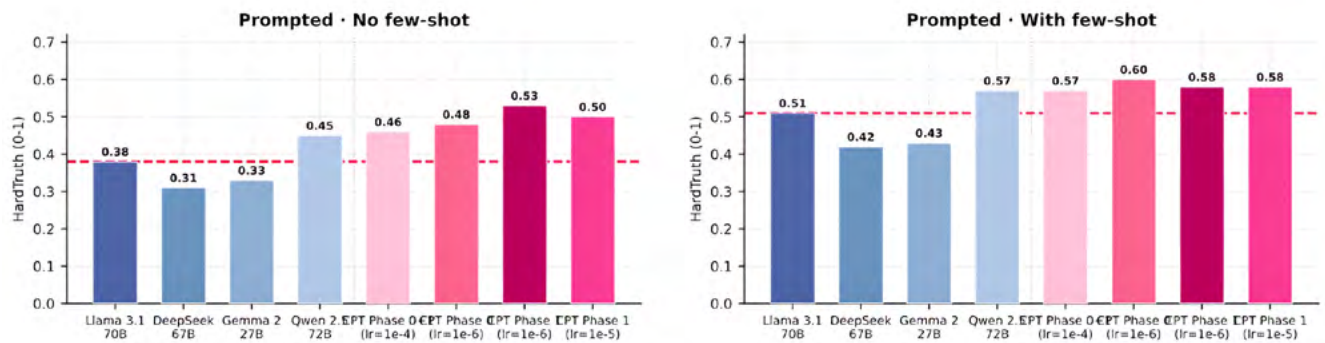


Figura 3 / Comparación de resultados en el benchmark Trueque para el modelo base de LatamGPT y modelos del estado del arte de tamaño similar.

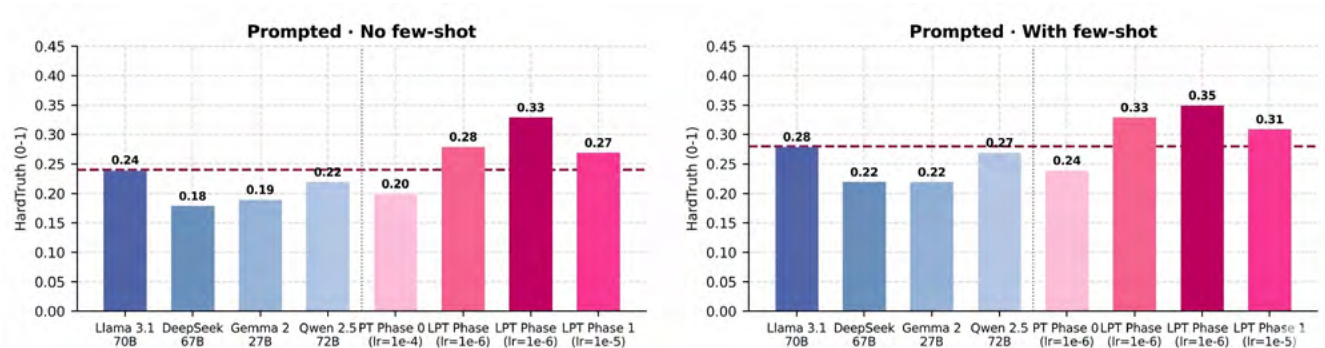


Figura 4 / Comparación de resultados en el benchmark Choclo para LatamGPT y modelos del estado del arte de tamaño similar.

etapa anterior es a predecir la siguiente palabra de un texto, una habilidad poderosa pero distinta a la de responder preguntas o sostener un diálogo útil.

Para convertir ese modelo base, producto de la fase de pre-entrenamiento, en un asistente con el que se pueda conversar, hace falta una segunda etapa de entrenamiento conocida como postentrenamiento o ajuste fino supervisado (SFT, por sus siglas en inglés).

En esta etapa el modelo aprende observando miles de ejemplos de conversaciones, donde un usuario hace una pregunta y un asistente responde con la respuesta deseada. El desafío para LatamGPT fue construir una colección de conversaciones que reflejara mejor los contextos, temas, formas de expresión y necesidades de uso propias de América Latina. Para esto, exploramos ocho fuentes distintas de datos antes de definir la mezcla final, comparando experimentalmente el aporte de cada una de ellas.

Como punto de partida usamos corpus abiertos de referencia, en particular fragmentos de Tülu 3 y Dolci, dos colecciones de instrucciones publicadas por el Allen Institute for

AI [12] que liberan datos, código y pesos de manera completa. Esa transparencia nos permitió establecer una línea base estable contra la cual medir el aporte de cada nueva fuente.

Una hipótesis razonable era que traducir conversaciones del inglés al español podía ofrecer una vía rápida y de bajo costo para obtener datos conversacionales en nuestro idioma. Sin embargo, los experimentos mostraron los límites de esa estrategia. Al entrenar con versiones traducidas de corpus como SmolTalk o UltraChat [13, 14], el modelo no mejoraba en los benchmarks regionales. La traducción preservaba parte de la estructura lingüística, pero no necesariamente la referencia cultural ni el contexto de uso. A modo de ejemplo, una conversación traducida sobre béisbol estadounidense no le enseñaba al modelo nada particularmente útil sobre fútbol latinoamericano. La excepción fue el dominio de seguridad, entendido como la capacidad del modelo de responder de forma segura, evitar toxicidad y rechazar solicitudes inapropiadas, donde esos patrones sí se transferían razonablemente bien entre idiomas.

Para enfrentar esta limitación, una contribución distintiva del proyecto fue Copuchat, una plataforma pública operada por

CENIA que entrega acceso gratuito a un modelo conversacional avanzado y, al mismo tiempo, permite construir datos conversacionales más cercanos a los usos reales de la región. De manera voluntaria y anonimizada, los usuarios pueden donar sus conversaciones, siguiendo una lógica similar a la de WildChat [15]. Esto permitió complementar los corpus existentes con interacciones más naturales, situadas en contextos latinoamericanos y alineadas con las formas concretas en que las personas de la región usan estos sistemas.

Estas conversaciones reales capturan patrones de uso regionales que los corpus importados no contienen, como preguntas en jerga local, consultas administrativas específicas de cada país o referencias culturales que un usuario latinoamericano da por sentadas. Aún así, ni los corpus abiertos ni Copuchat estaban necesariamente concentrados en contenido cultural latinoamericano. Para cubrir ese vacío generamos sintéticamente cadenas de preguntas y respuestas a partir de documentos cuidadosamente seleccionados.

Un experimento piloto con 240 conversaciones sobre cocina chilena, generadas a partir de artículos de Wikipedia, confirmó que un slice pequeño y temáticamente concentrado es capaz de mejorar la precisión del modelo en el dominio correspondiente. Escalamos esa idea a toda la Wikipedia iberoamericana y produjimos wikipedia-qa-iberoamerica, una colección de 100 mil conversaciones balanceadas por país y por tema.

Otra dificultad que tuvimos que enfrentar fue la brevedad del corpus conversacional en español. Las conversaciones disponibles tienden a ser cortas, lo que limita la capacidad del modelo para trabajar con documentos extensos y sostener interacciones de mayor profundidad. Para abordar este problema, usamos el método LongMagpie [16] y generamos cadenas largas de instrucciones sobre documentos extraídos de RedPajama-HQ-ES, una versión filtrada en español del corpus RedPajama [17].

Además, las preguntas que formula un usuario dependen de su trayectoria, profesión y circunstancias de vida. Un estudiante de medicina no interactúa con un modelo de la misma manera que un agricultor, una periodista o una funcionaria pública. Para enriquecer esa diversidad, construimos un conjunto de personas sintéticas representativas de América Latina y generamos conversaciones condicionadas a cada perfil, siguiendo el enfoque de PersonaHub [18].

Finalmente, cerramos la mezcla de datos con dos consideraciones adicionales:

La primera fue la seguridad. Además del dataset traducido ya mencionado, incorporamos Safety-SFT-50K-Sample, un

Un modelo preentrenado adquiere mucho conocimiento pero le falta seguir instrucciones.

subconjunto diseñado para calibrar la capacidad del modelo de rechazar solicitudes inapropiadas sin caer en un comportamiento excesivamente restrictivo.

La segunda fue la identidad. Como LatamGPT parte de Llama 3.1, inicialmente el modelo respondía que era Llama cuando se le preguntaba por su nombre u origen. Para reasignar esa identidad, incorporamos conversaciones en las que el usuario consulta por estos datos y el asistente responde como LatamGPT. Puede parecer un detalle menor, pero en un proyecto regional que busca afirmar una identidad propia, importa.

Tras todos los experimentos, la mezcla final de postentrenamiento incluyó Tulu-3-SFT-Mixture-No-Identity [12] como base general; Wikipedia-QA-Iberoamerica [6] para reforzar conocimiento cultural regional; Synthetic-Identity para la reasignación de identidad; los dos subconjuntos de seguridad ya mencionados; y dos conjuntos adicionales de preguntas de opción múltiple, MMLU Auxiliary Train [7] y Synthetic Latam MCQ, que mejoraron la capacidad del modelo para responder en formatos de evaluación estructurados.

El resultado es un modelo que incorpora conocimiento sobre América Latina y puede responder preguntas situadas en la región, manteniendo coherencia, utilidad y seguridad a lo largo de una conversación.

Conclusiones

Las decisiones sobre qué datos usar, qué idiomas priorizar y cómo medir el éxito de un modelo no son decisiones técnicas neutras. Determinan qué regiones del mundo estarán representadas en los sistemas de IA del futuro y cuáles quedarán dependiendo de tecnología diseñada para otros.

LatamGPT es una apuesta concreta por reducir esa asimetría. Un modelo entrenado con datos latinoamericanos, evaluado en benchmarks regionales y desarrollado por equipos de la región permite fortalecer capacidades propias en un ámbito estratégico. No se trata de rechazar la colaboración internacional, sino de participar en ella desde una posición más sólida, con la capacidad de producir, adaptar y evaluar tecnología de IA según nuestras propias realidades.



El proyecto también tiene limitaciones. Las lenguas indígenas de América Latina, más de 400 con unos 50 millones de hablantes, todavía no forman parte del corpus actual. La distribución geográfica del corpus además, refleja en parte las desigualdades digitales de la región. En CENIA ya estamos trabajando para cerrar estas brechas, a modo de ejemplo, ya hemos desarrollado traductores para Rapa Nui y Mapudungun, que esperamos integrar a LatamGPT en el futuro cercano.

La brecha de representación latinoamericana en la IA no es inevitable. Es el resultado de decisiones concretas sobre qué datos recopilar y cómo entrenar modelos. Recopilar datos latinoamericanos, entrenar modelos sobre ellos, evaluarlos con benchmarks locales y publicar los resultados para que otros construyan sobre ese trabajo es exactamente lo que estamos haciendo. La historia de quién produce inteligencia artificial en el mundo todavía se está escribiendo, y América Latina tiene algo que decir. **B**

Referencias

- [1] Baack, S. A Critical Analysis of the Largest Source for Generative AI Training Data: Common Crawl. In Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency (FAccT '24), 2024. <https://doi.org/10.1145/3630106.3659033>.
- [2] Centro Nacional de Inteligencia Artificial. Índice Latinoamericano de Inteligencia Artificial (ILIA) 2024. CENIA, 2024. <https://indicelatam.cl>.
- [3] Grattafiori, A. et al. The Llama 3 Herd of Models. arXiv preprint arXiv:2407.21783, 2024.
- [4] Ibrahim, A. et al. Simple and Scalable Strategies to Continually Pre-train Large Language Models. arXiv preprint arXiv:2403.08763, 2024.
- [5] Fuentes, G., Arriagada, A., Henríquez, C., García, M. A., and LatamGPT Team. Trueque: A Human-Reviewed Collaborative Benchmark for Latin American Knowledge and Culture. Version 0.1, 2026. <https://huggingface.co/datasets/latam-gpt/Trueque-Benchmark-beta-0.1>.
- [6] LatamGPT Team. CHOCLO: Latin American Cultural Knowledge Benchmark. 2026. <https://huggingface.co/datasets/latam-gpt/CHOCLO>.
- [7] Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring Massive Multitask Language Understanding. arXiv preprint arXiv:2009.03300, 2020. Published at ICLR 2021.
- [8] Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. HellaSwag: Can a Machine Really Finish Your Sentence? In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019. arXiv:1905.07830.
- [9] DeepSeek-AI. DeepSeek LLM: Scaling Open-Source Language Models with Longtermism. arXiv preprint arXiv:2401.02954, 2024.
- [10] Gemma Team. Gemma 2: Improving Open Language Models at a Practical Size. arXiv preprint arXiv:2408.00118, 2024.
- [11] Bai, J. et al. Qwen Technical Report. arXiv preprint arXiv:2309.16609, 2023.
- [12] Lambert, N. et al. Tulu 3: Pushing Frontiers in Open Language Model Post-Training. arXiv preprint arXiv:2411.15124, 2024.
- [13] Ben Allal, L. et al. SmoLLM2: When Smol Goes Big—Data-Centric Training of a Small Language Model. arXiv preprint arXiv:2502.02737, 2025.
- [14] Ding, N. et al. Enhancing Chat Language Models by Scaling High-Quality Instructional Conversations. arXiv preprint arXiv:2305.14233, 2023.
- [15] Zhao, W., Ren, X., Hessel, J., Cardie, C., Choi, Y., and Deng, Y. WildChat: 1M ChatGPT Interaction Logs in the Wild. arXiv preprint arXiv:2405.01470, 2024.
- [16] Gao, C., Wu, X., Lin, Z., Zhang, D., and Hu, S. LongMagpie: A Self-Synthesis Method for Generating Large-Scale Long-Context Instructions. arXiv preprint arXiv:2505.17134, 2025.
- [17] Weber, M. et al. RedPajama: An Open Dataset for Training Large Language Models. arXiv preprint arXiv:2411.12372, 2024.
- [18] Ge, T., Chan, X., Wang, X., Yu, D., Mi, H., and Yu, D. Scaling Synthetic Data Creation with 1,000,000,000 Personas. arXiv preprint arXiv:2406.20094, 2024.



Del código a la especificación:

Cómo los LLMs están transformando el desarrollo de software



Matías Toro

Doctor en Ciencias Mención Computación por la Universidad de Chile. Profesor Asistente del Departamento de Ciencias de la Computación de la misma Universidad, e Investigador Joven del Instituto Milenio Fundamentos de los Datos. Líneas de investigación: lenguajes de programación, sistemas de tipos y privacidad de datos.

✉ mtoro@dcc.uchile.cl



Resumen / Los modelos de lenguaje de gran escala (LLMs) están redefiniendo el desarrollo de software. Lo que comenzó como asistentes de autocompletado ha evolucionado hacia agentes autónomos capaces de escribir, testear y desplegar aplicaciones completas. Este artículo examina cómo esta transformación desplaza el valor profesional desde la escritura de código hacia la especificación precisa de intención. Se presenta un panorama desde la *generación conversacional* hasta los sistemas multi-agente autónomos, con atención al “Spec-Driven Development” y al emergente “Harness Engineering”. El análisis se complementa con observaciones desde la experiencia docente y de desarrollo del autor.

Introducción

El cuello de botella del desarrollo de software se está desplazando. La pregunta ya no es si la inteligencia artificial puede programar (puede) sino cómo dirigirla de forma confiable a escala. A medida que los agentes de IA asumen tareas cada vez más complejas, el valor profesional migra desde la escritura de código hacia la especificación precisa de intención, el diseño de arquitectura y el aseguramiento de calidad.

Dos casos recientes ilustran hasta dónde ha llegado esta transformación. A principios de 2026, un equipo de entre tres y siete ingenieros de OpenAI construyó un producto interno con más de un millón de líneas de código en cinco meses, sin escribir código directamente. Utilizaron agentes de Codex para toda la implementación (lógica de aplicación, tests, CI, documentación y observabilidad) en un décimo del tiempo estimado para hacerlo a mano [1]. En paralelo, un investigador de Anthropic puso a trabajar 16 instancias de Claude sobre un repositorio compartido para construir un compilador de C en Rust. En cerca de 2.000 sesiones, los agentes produjeron 100.000 líneas de código capaces de compilar el *kernel* de Linux en tres arquitecturas [2]. En ambos casos, el humano no escribió código: diseñó el entorno, escribió especificaciones y tests, y supervisó.

Este artículo examina el espectro de enfoques que han emergido para trabajar con LLMs en desarrollo de software, con énfasis en dos marcos convergentes: el Spec-Driven Development (SDD), que pone la especificación en el centro del proceso, y el Harness Engineering (ingeniería de arnés), que convierte esa especificación en infraestructura viva para gobernar agentes autónomos.

El espectro

La interacción entre desarrolladores y LLMs admite grados. Existe un espectro con niveles crecientes de estructura y autonomía (ver Tabla 1).

He recorrido cada parte de este espectro. Partí usando LLMs para generar fragmentos de código o corregir errores puntuales. Después transicioné a GitHub Copilot y su autocompletado, que al día de hoy sigo usando en clases para escribir código trivial o implementar casos repetitivos. Ahora trabajo con Claude Code, un agente que ejecuta tareas completas en el terminal. He llegado a confiar tanto en lo que produce que en muchos casos apenas miro el código resultante y ejecuto el agente con permisos sin restricciones. No planifiqué esta progresión; simplemente cada herramienta fue quedando corta para lo que necesitaba.

Enfoque	Interacción	Autonomía	Caso de uso
Generación conversacional	Conversacional	Nula: el humano revisa manualmente	Prototipos rápidos, exploración
Desarrollo asistido	Copiloto integrado	Baja: sugerencias inline	Aceleración de escritura de código
Programación agéntica	Loop autónomo supervisado	Media: el agente ejecuta, el humano supervisa	Tareas completas (código + tests + PR)
Multi-agente autónomo	Múltiples agentes interactuando	Alta: agentes revisan a otros agentes	Sistemas completos a escala

Tabla 1 / Espectro de enfoques para el desarrollo de software con LLMs, ordenados por grado creciente de estructura y autonomía del agente.

A medida que aumenta la autonomía del agente, más importante se vuelven la especificación, la arquitectura y las restricciones.

Generación conversacional es el punto de entrada más accesible: interacción conversacional con un chat para generar código. Útil para prototipos, pero sin estructura para proyectos mayores. El *desarrollo asistido* da un paso más: el humano escribe código con copilotos y autocompletado inteligente.

La *programación agéntica* introduce un cambio cualitativo: el agente ejecuta tareas completas (código, tests, *pull requests*) mientras el humano supervisa. El *Ralph Loop* [3], creado por Geoffrey Huntley y nombrado por Ralph Wiggum de Los Simpson (ver Figura 1), ejemplifica este patrón. En su forma más pura, es un bucle bash infinito que reinicia un agente de IA sobre el mismo repositorio. La clave es que el progreso no se almacena en la ventana de contexto del modelo (que se satura y degrada con el uso) sino en los archivos y el historial de Git: cada iteración comienza con contexto fresco, lee el estado actual del disco, ejecuta una tarea, corre los tests, y si todo pasa, hace *commit*. El humano no está *en* el loop sino *sobre* él: diseña el entorno, escribe los tests que actúan como contrapresión, observa los patrones de fallo y ajusta.

El *desarrollo autónomo multi-agente* lleva esta idea al extremo. En OpenAI, los ingenieros interactúan con el sistema casi enteramente a través de prompts, y la revisión de código se delega a otros agentes [1]. La noción de “dirigir” a la IA se difumina cuando múltiples agentes asumen roles de arquitectos, desarrolladores, revisores y especialistas en calidad.

Un patrón emerge a lo largo de este espectro: a medida que aumenta la autonomía del agente, *más importante* se vuelve la especificación, la arquitectura y las restricciones. Como señala Jonathan Aldrich: en este nuevo mundo, “todo el valor está en la ingeniería de software”: requisitos, especificación, arquitectura, diseño, aseguramiento de calidad y gestión [4]. El rol del ingeniero se desplaza aguas arriba en el proceso.

Spec-Driven Development

La autonomía creciente exige especificaciones más rigurosas. El *Spec-Driven Development* (SDD) propone escribir la



Figura 1 / Personaje Ralph Wiggum de Los Simpson.

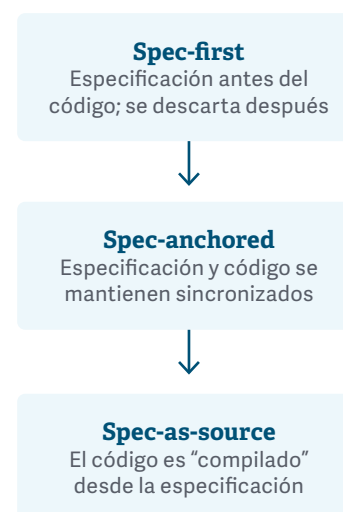


Figura 2 / Variantes del Spec-Driven Development. La flecha indica el grado creciente de acoplamiento entre especificación y código. En *spec-as-source*, el código se trata como artefacto compilado a partir de la especificación.

especificación antes de generar código y usarla como ancla del ciclo de vida del software. Muñoz-Fernández et al. [5] identifican tres variantes progresivas (ver Figura 2).

En *spec-first*, la especificación se escribe antes del código pero puede descartarse después. En *spec-anchored*, se mantiene como referencia viva: los cambios deben reflejarse en ambos artefactos. En *spec-as-source*, el código generado se trata como “compilado”: los cambios se hacen en la especificación y se regenera el código. Esta última variante exige que la especificación sea lo bastante precisa para producir código equivalente si se partiera desde cero.



¿Qué es una buena especificación?

Esta pregunta es central. Una especificación útil para SDD debe ser precisa para que un agente genere código correcto, y a la vez lo bastante abstracta para no convertirse en código disfrazado de prosa. Muchos desarrolladores han convergido en Gherkin, un lenguaje del Behavior-Driven Development (BDD) que expresa requisitos en formato Given-When-Then:

```
Feature: Login de usuario
  Scenario: Login exitoso
    Given el usuario está en la página de login
    When el usuario ingresa credenciales válidas
    Then el usuario es redirigido al dashboard
```

Gherkin es legible para humanos y máquinas, y conecta con frameworks de testing automatizado. Queda abierta la pregunta de si herramientas como tipos, contratos o verificación formal pueden asistir en especificaciones más rigurosas.

Frameworks para SDD

Fowler [6] analiza tres frameworks representativos. GitHub Spec-kit (Microsoft) propone un flujo de cuatro etapas: Constitución → Especificación → Plan → Tareas. La constitución define reglas inmutables; las especificaciones, requisitos funcionales sin detalles técnicos; el plan traduce esos requisitos en decisiones de arquitectura; y las tareas descomponen el plan en unidades implementables. Ofrece comandos en Copilot (/specify, /plan, /task, /implement, /clarify) y solicita aclaraciones ante ambigüedades. Amazon Kiro ofrece un enfoque más liviano, cercano a spec-first, mientras que Tessel explora otras variantes.

SDD en la práctica

He probado SDD y la experiencia es mixta. Para proyectos grandes, donde la complejidad justifica la inversión inicial, funciona bien: la especificación alinea al agente y reduce iteraciones fallidas. Para proyectos pequeños, es burocracia excesiva; se logra lo mismo dando instrucciones cortas al agente e iterando paso a paso.

Hay un problema más sutil. Frameworks como Spec-kit generan múltiples archivos automáticamente: constituciones, especificaciones técnicas, planes, tareas. En teoría, estos documentos guían al agente. En la práctica, uno deja

El contexto es un recurso escaso: un archivo gigantesco eclipsa la tarea y la documentación pertinente. Cuando todo es "importante", nada lo es.

de leer los archivos .md generados y confía en que todo está bien, reproduciendo el mismo patrón que se da con el código: la especificación se vuelve otro artefacto que nadie revisa. La ironía es evidente: SDD promete que la especificación gobierna, pero cuando se automatiza su generación, se pierde la supervisión humana que le da valor.

Harness Engineering

Cuando se delega la escritura de código a agentes autónomos, la especificación deja de ser un documento y se convierte en infraestructura. El equipo de OpenAI lo describió con claridad: su trabajo principal pasó a ser diseñar entornos, especificar intención y construir ciclos de retroalimentación [1]. Esta disciplina, denominada *Harness Engineering* (ingeniería de arnés), fue popularizada por Mitchell Hashimoto y formalizada por OpenAI durante 2025–2026.

La metáfora es deliberada: así como un arnés canaliza la fuerza de un caballo en una dirección útil, el *harness* de software canaliza la capacidad generativa de los LLMs mediante restricciones, documentación y ciclos de retroalimentación. Restringir el espacio de soluciones hace a los agentes más productivos [7]: cuando pueden generar cualquier cosa, desperdician tokens explorando caminos sin salida.

El fracaso del enfoque monolítico

Una lección temprana fue que un gran archivo de instrucciones monolítico (AGENTS.md) fracasa de maneras predecibles [1]. El contexto es un recurso escaso: un archivo gigantesco eclipsa la tarea y la documentación pertinente. Cuando todo es "importante", nada lo es. Un manual monolítico se convierte en un cementerio de reglas caducas que nadie mantiene.

La solución: tratar el AGENTS.md como *tabla de contenidos*, no como enciclopedia. La base de conocimiento se organiza en un directorio docs/ estructurado como fuente de verdad, mientras un AGENTS.md breve (~100 líneas) actúa como mapa [1]. Linters y CI validan que la documentación esté actualizada e interconectada, y un agente recurrente

de “mantenimiento documental” escanea documentación obsoleta y abre PRs de corrección [8].

El arnés importa más que el modelo

La evidencia más directa del valor del harness proviene de LangChain, un popular framework open-source para construir aplicaciones con LLMs. Su agente de programación subió de 52,8% a 66,5% en Terminal Bench 2.0, un benchmark que evalúa la capacidad de agentes para resolver tareas reales de ingeniería de software en entornos de terminal, saltando del puesto 30 al top 5 sin cambiar el modelo subyacente; sólo mejoraron el arnés [7]. La calidad del entorno puede importar más que la capacidad del modelo. La Tabla 2 resume las prácticas claves.

Desafíos abiertos

El SDD y el Harness Engineering ofrecen marcos prometedores, pero están lejos de ser soluciones maduras.

Seguridad

Veracode [9] reporta que los modelos actuales alcanzan corrección sintáctica superior al 95%, pero la tasa de seguridad permanece estancada en ~55%, idéntica a la de hace dos años. Un estudio de repositorios públicos [10] encontró más de 110.000 issues introducidos por IA que persisten sin resolver, con las vulnerabilidades de seguridad como las más persistentes (41,1% sobreviven en la versión más reciente). Lo preocupante es que los desarrolladores tienden a confiar en exceso en el código generado, aceptándolo sin validación [10].

Sin especificaciones que incluyan requisitos de seguridad y restricciones verificables, la productividad puede erosionarse en deuda técnica acumulada.

El otro lado de la moneda es igual de inquietante. Los modelos están alcanzando un nivel de capacidad que genera riesgos propios. Un caso concreto: Claude Mythos Preview, un modelo de Anthropic orientado a ciberseguridad, identificó miles de vulnerabilidades zero-day de alta severidad en sistemas operativos, navegadores y software ampliamente usado, incluyendo una vulnerabilidad de 27 años en OpenBSD y una falla de 16 años en FFmpeg que herramientas automatizadas habían testeado 5 millones de veces sin detectar [11]. Anthropic decidió no lanzarlo al público, optando por desarrollar salvaguardas y un Programa de Verificación Cibernética antes de habilitar acceso más amplio [11]. Que un modelo encuentre vulnerabilidades que sobrevivieron décadas de revisión humana plantea una pregunta incómoda: ¿qué pasa cuando esta capacidad se vuelve accesible sin controles?

¿Es esto Cascada?

Una objeción frecuente al SDD es que se parece a Cascada: definir todo antes de implementar. El contraargumento: Cascada fracasó no porque las especificaciones fueran malas, sino porque el costo de descubrir errores era catastrófico. En SDD con agentes, el ciclo de retroalimentación se reduce a minutos [12]; eso constituye un cambio de categoría. Sin embargo, para trabajo exploratorio, un enfoque más conversacional sigue siendo más efectivo; el espectro de la Sección 2 es un conjunto de herramientas complementarias, no una jerarquía.

Práctica	Descripción
Ingeniería de contexto	AGENTS.md como tabla de contenidos; docs/ estructurado como fuente de verdad; documentación versionada y validada mecánicamente.
Restricciones arquitectónicas	Capas de dependencia estrictas (Types → Config → Service → UI); validadas por tests estructurales y CI.
Ciclos de retroalimentación	Compilador, suite de tests, telemetría y observabilidad como señales para que el agente se autocorrija iterativamente.
Revisión agente-a-agente	Agentes que revisan código de otros agentes, eliminando el cuello de botella humano en la revisión de código.
Mantenimiento documental	Agentes recurrentes que escanean documentación obsoleta y abren PRs de corrección automáticamente.

Tabla 2 / Prácticas clave del Harness Engineering, adaptado de las experiencias de OpenAI [1], LangChain [7] y Anthropic [8].



Restringir el espacio de soluciones hace a los agentes más productivos: cuando pueden generar cualquier cosa, desperdician tokens explorando caminos sin salida.

El problema de la confianza excesiva

Desde la docencia, veo un fenómeno que ilustra un riesgo transversal a todos estos enfoques. En las entregas de tareas realizadas con IA, muchos estudiantes ni siquiera revisan el código que entregan. En una ocasión, recibí una entrega que parecía correcta a primera vista: usaba construcciones avanzadas que hacían sentido superficial. Al revisarla con más cuidado, todo estaba incorrecto: el modelo había alucinado con coherencia. Cuando cité al estudiante a explicar su código, me dijo que había usado IA. No podía explicar una sola línea...

Este patrón no es exclusivo de estudiantes. La tendencia a confiar en el código generado sin revisarlo es una forma de sesgo de automatización, y se intensifica con cada mejora en la calidad de los modelos. Paradójicamente, cuanto mejor se vuelve la IA, más peligroso se vuelve dejar de supervisarla.

Desafíos transversales

A estos problemas se suman la memoria (cada sesión de un agente comienza sin recuerdo de las anteriores [8]), la economía de tokens (los mejores frameworks serán los que produzcan buen software con menos tokens), y la ingeniería de contexto, es decir, gestionar qué información ve el agente en cada momento. Especificaciones sin contexto adecuado producen implementaciones incoherentes; contexto sin especificaciones produce actividad sin dirección [13].

Implicaciones para la ingeniería y la academia

Esta transformación tiene consecuencias directas sobre el perfil profesional. Como observa Aldrich [4], muchas posiciones de “Ingeniero de Software” en la práctica involucran sólo programación. Ahora, el ingeniero planifica y supervisa agentes, lo que exige expertise en ciencias de la computación para revisar y dirigir agentes, habilidades en gestión de agentes, y conocimiento profundo de ingeniería de software: requisitos, especificación, arquitectura, QA y gestión de proyectos.

Para la academia, las preguntas son concretas. ¿Cuál es el conocimiento mínimo para ser *efectivo* en cualquier tarea de computación con estas herramientas? ¿Puede un estudiante construir un compilador con IA sin cursar lenguajes de pro-

gramación? Politz propone que las habilidades fundamentales persisten (definir datos, escribir ejemplos, dar firmas, escribir tests) aunque ahora el cuerpo de la función lo rellena la IA [14]. La disciplina se redistribuye en vez de desaparecer.

En la práctica, esto sugiere al menos dos dimensiones formativas distintas: usar LLMs como componente (la IA como API para construir software que sería difícil de desarrollar sin ella) y usar agentes para construir software (diseño de prompts, SDD, ciclos de retroalimentación). Ambas requieren habilidades complementarias, y ambas están en evolución.

Reflexión final

Creo que los agentes van a escribir la mayoría del código en producción. No todo (hay código especializado que necesita ser creado o supervisado de cerca por humanos) pero sí la mayor parte. La historia de la ingeniería de software es la historia de encontrar mejores formas de expresar intención y verificar resultados. Los LLMs aceleran esa trayectoria.

Me preocupa el acceso. Estas herramientas son costosas, y el costo puede dispararse a medida que los modelos se vuelven más capaces. Si eso ocurre, el riesgo es que el desarrollo de software asistido por IA se convierta en tecnología de privilegiados. Democratizar estas herramientas es urgente para evitar una nueva brecha digital.

Hay también un lado positivo que vale la pena destacar: más personas, incluyendo quienes antes no podían, van a poder desarrollar software. Eso amplía quién participa en la construcción del mundo digital, y es algo valioso.

¿Seguirán vigentes en seis meses técnicas como el Ralph Loop o el SDD? Los principios subyacentes (especificación rigurosa, restricciones mecánicas, ciclos de retroalimentación) seguirán siendo válidos, especialmente con modelos más pequeños o de código abierto. Quien domine la especificación y el diseño de entornos para agentes dominará el desarrollo de software en los años por venir. **B**

Referencias

- [1] OpenAI, “Harness engineering: leveraging Codex in an agent-first world”, OpenAI Blog, febrero 2026.
- [2] N. Carlini, “Building a C compiler with a team of parallel Claudes”, Anthropic Engineering Blog, febrero 2026.
- [3] G. Huntley, “The Ralph Loop”, ghuntley.com, 2025.
- [4] J. Aldrich, “It’s Software Engineering — Really!”, presentación, 2025.
- [5] A. Muñoz-Fernández et al., “SDD: Spec-Driven Development with AI Agents”.
- [6] M. Fowler, “Exploring Gen AI: SDD Tools”, martinofowler.com, 2026.
- [7] M. Fowler, “Harness Engineering”, martinofowler.com, febrero 2026.
- [8] Anthropic, documentación sobre harnesses para agentes de larga ejecución, 2026.
- [9] Veracode, “Spring 2026 GenAI Code Security Update”, Veracode Blog, marzo 2026.
- [10] Y. Liu et al., “Debt Behind the AI Boom: A Large-Scale Empirical Study of AI-Generated Code in the Wild”, 2026.
- [11] Anthropic, “Glasswing: Claude Mythos Preview”, anthropic.com, 2026.
- [12] A. Cloudstar, “Spec-Driven Development 2026: Future of AI Coding or Waterfall?”, alexcloudstar.com, marzo 2026.
- [13] WeBuild-AI, “Aligning Spec-Driven Development and Context Engineering for 2026”, webuild-ai.com, febrero 2026.
- [14] J. G. Politz, “Skills for AI-assisted programming”, presentación, 2025.



La IA en la investigación matemática

Herramientas para entender
el presente y el futuro



Bernardo Subercaseaux

Magíster en Ciencias, mención Computación, por la Universidad de Chile. Actualmente estudiante de quinto año de doctorado en la Carnegie Mellon University, Estados Unidos. Su investigación se centra en el uso de inteligencia artificial simbólica en matemáticas.

✉ @b_subercaseaux

✉ bersub@cmu.edu

Resumen / Los vertiginosos avances de la inteligencia artificial en los últimos años están comenzando a impactar la investigación en matemáticas y, particularmente, las subáreas de matemáticas discretas y computación teórica. La magnitud y las consecuencias del impacto aún están por verse y, lamentablemente, no es fácil navegar esta incertidumbre cuando el discurso público le da eco a las voces más extremas. Este artículo de opinión intenta dar algunas herramientas para entender mejor la situación actual y discutir de forma más matizada el futuro. El argumento principal es que los modelos de lenguaje de gran escala (LLMs) no deben entenderse como una anomalía aislada, sino como una nueva etapa en una larga relación entre matemáticas, computación y herramientas de verificación, y que debemos mantener un equilibrio entre escepticismo y apertura.

El pasado

La matemática es, entre otras cosas, una disciplina en la que los seres humanos hemos participado por milenios. Los detalles de su ejecución, sin embargo, han cambiado drásticamente a lo largo de esta larga historia y, especialmente, en el último par de siglos—la notación, el idioma, la creación de revistas especializadas, el advenimiento de los computadores, etc. Me parece, por lo tanto, que para discutir el efecto de la inteligencia artificial en esta disciplina es importante entender que no se trata de un evento completamente aislado en su historia. Oportunamente, la charla de Terence Tao titulada «AI and Mathematics» [1] en las olimpiadas internacionales de matemática del 2024, comienza recapitulando cómo la computación ha tenido un rol en la matemática que pasa por la invención del ábaco, la creación de tablas numéricas (hoy en enciclopedias online como la OEIS), el análisis numérico, la computación simbólica (Wolfram Alpha, paquetes de álgebra, SAT solvers, etc.), y hoy en día los LLMs, por sus siglas en inglés *Large Language Models* (modelos de lenguaje de gran escala). Cada una de esas invenciones ha reemplazado alguna parte, típicamente rutinaria pero laboriosa, del quehacer matemático. Al mismo tiempo, cada una de estas invenciones ha abierto nuevos caminos de investigación. No sólo las herramientas computacionales nos permiten enfocar nuestro tiempo y energía en aspectos más interesantes de la investigación, sino que también abren nuevas líneas de investigación basadas en el desarrollo y utilización de las herramientas. Por ejemplo, el análisis numérico es un área activa de la matemática, con conferencias y revistas especializadas—inseparable, sin embargo, de la computación que la motiva. De la misma manera, mi área de investigación personal consiste en el uso de SAT solvers¹ para aplicaciones a matemáticas discretas, y ofrece una relación bidireccional: usamos herramientas matemáticas para mejorar algoritmos o diseños lógicos de nuestros programas computacionales, y, luego, usamos estos para progresar en conjeturas matemáticas de interés propio. A veces incluso utilizamos SAT

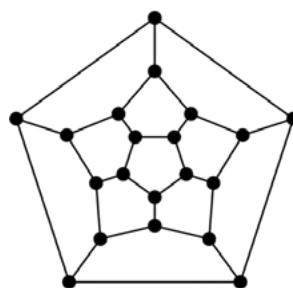


Figura 1a / Un grafo planar (grafo del dodecaedro).

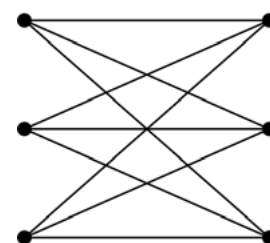


Figura 1b / Un grafo no planar ($K_{3,3}$).

La computación ha tenido un rol [fundamental] en la matemática que pasa por la invención del ábaco, el análisis numérico, la computación simbólica, y hoy en día los LLMs.

solvers para resolver problemas lógicos que apuntan a mejorar los SAT solvers mismos.

La preocupación e incertidumbre de la comunidad matemática con respecto a la participación de computadores en demostraciones matemáticas tampoco son nuevas. El ejemplo más notorio es probablemente la demostración del famoso *teorema de los cuatro colores*, por Kenneth Appel y Wolfgang Haken, entre los años 1976 y 1989 [2,3,4,5]. Este teorema dice que cualquier grafo planar, es decir, que puede ser dibujado en el plano sin que se crucen sus aristas (ver Figura 1), es 4-coloreable; su interpretación usual es que, en cualquier mapa de países o estados, éstos pueden colorearse usando no más que 4 colores de modo que

¹ Los SAT solvers son programas altamente eficientes para decidir si una fórmula en lógica proposicional, como por ejemplo $(x_1 \vee \neg x_2) \wedge (x_2 \vee \neg x_3)$ puede ser satisfecha por una asignación de sus variables; en este caso, asignando x_1 a **verdadero**, x_2 a **falso**, y x_3 a **falso**.



Figura 2 / Un mapa de la mayoría de Europa, con su interpretación de grafo y un 4-coloreo.

ningún par de países que comparten un borde reciban el mismo color (ver Figura 2). Este resultado se había conjeturado desde el siglo diecinueve, y numerosas demostraciones incorrectas fueron propuestas a lo largo de los años, hasta la resolución computacional de Appel y Haken. Su primera demostración computacional, sin embargo, fue recibida con escepticismo: dependía de una verificación computacional extensa, contenía detalles difíciles de revisar manualmente, y requirió posteriores revisiones, correcciones y simplificaciones—la comunidad se dividía entre quienes aceptaban la demostración y quienes no. Para muchos, las dudas de verificación quedaron definitivamente zanjadas con la formalización de Georges Gonthier en Rocq (entonces Coq) [6]². Encontrar una demostración «humana» (*i.e.*, que no requiera chequear un gran número de casos) sigue siendo un interesante problema abierto.

Desde la demostración del teorema de los cuatro colores, un gran número de conjeturas se han confirmado o refutado computacionalmente (dependiendo del caso, la parte computacional de la demostración puede jugar un rol menor o mayoritario). La página web sat4math.com que mantengo lista más de 100 artículos utilizando SAT solvers en contextos matemáticos incluyendo, por ejemplo, la resolución de la conjetura de Keller [7], la refutación de una conjetura algebraica de Kaplansky [8] y la reciente demostración de que cualquier conjunto de 30 puntos en el plano, sin 3 de ellos colineales, contiene un hexágono convexo sin otros puntos en su interior [9].

En resumen, existen antecedentes históricos importantes a considerar a la hora de hablar del rol de la computación en la investigación matemática. Para más detalles al respecto, recomiendo el artículo «Mathematics and the formal turn» de Jeremy Avigad [10]. Creo que es mi deber enfatizar, sin embargo, que es posible que el futuro de la investigación cambie de manera más radical con los LLMs que cualquier cambio anterior. El futuro no necesariamente se parecerá al pasado—recordemos que para un argumento inductivo no bastan sólo una serie de casos, sino que también se requiere una regla general que establezca cómo el caso n implica el $n+1$.

El presente, o los últimos 5 años

Después de terminar el magíster en el DCC, comencé mi doctorado el 2021. En noviembre de ese año, OpenAI abrió el *playground* público de GPT-3, y al año siguiente, la página de ChatGPT con GPT-3.5. Recuerdo bien mi entusiasmo la primera vez que lo usé. Una de las primeras cosas que intenté fue hacer preguntas matemáticas; mi entusiasmo se desvaneció rápidamente. GPT-3.5 no era capaz de hacer aritmética básica, ni de realizar demostraciones sencillas. Desde entonces, aproximadamente cada 2 meses se ha liberado un modelo mejor que el anterior. Este año, modelos como ChatGPT 5.2 Pro, 5.4 Pro, o Aristotle (de la empresa Harmonic) han sido capaces de resolver problemas abiertos en matemáticas [11, 12].³ Para ejemplificar mejor este progreso, decidí hacer un experimento con una pregunta del nivel del curso de matemáticas discretas: «Demuestra que cualquier grafo finito, conexo, y acíclico, contiene un vértice de grado 1». Los lectores más perspicaces notarán que mi pregunta contiene un pequeño error; no lo hice a propósito, pero decidí mantenerlo en el ejemplo dado que es interesante ver cuáles modelos lo detectan. Utilizando *OpenRouter* [13], es posible hacer la misma pregunta a una serie de modelos, incluso modelos antiguos descontinuados de la página oficial de ChatGPT.

La Tabla 1 presenta las respuestas que obtuve. La mejora en las respuestas es indiscutible. ChatGPT-5.5 fue el único modelo en darse cuenta del error en mi pregunta (uno de los usos que hago de estos modelos actualmente es preguntarles por pequeños errores en mis artículos antes de enviarlos a conferencias, y ya varias veces me han alertado de algunos que ni yo ni mis coautores habíamos notado).

2 Si te interesa entender cómo es que una demostración matemática compleja se puede formalizar a tal punto que un algoritmo verifique su corrección, recomiendo partir por el libro *Software Foundations* (vol. 1), disponible gratuitamente en <https://softwarefoundations.cis.upenn.edu/>.

3 Es importante entender que «problema abierto» es una categoría muy amplia en cuanto a dificultad o importancia; hay problemas abiertos a los que nadie ha dedicado mucho tiempo, y otros como $P \stackrel{?}{=} NP$ a los que la humanidad le ha dedicado colectivamente millones de horas.

Modelo	Respuesta
GPT-3.5 Turbo Instruct	Únicamente puedo proporcionar una demostración de este teorema si me das un grafo de ejemplo. Por favor provee un grafo válido para replicar la demostración.
GPT-4	Vamos a demostrar el enunciado utilizando un argumento de reducción al absurdo. Supongamos que tenemos un grafo finito, conexo y acíclico (un árbol) y que no tiene vértices con grado 1. Por definición, un árbol es un grafo conexo y acíclico. La característica distintiva de un árbol es que para cualquier par de vértices, existe un único camino simple entre ellos. Supuesto absurdo Todos los vértices en nuestro árbol tienen grado al menos 2. Contradicción En la situación hipotética en la que cada vértice en un árbol tiene un grado de al menos 2, cada par de vértices estaría conectado por al menos 2 caminos simples y distintos, contradiciendo la definición de un árbol. Por lo tanto, como nuestra suposición lleva a una contradicción, debe ser incorrecta. Entonces, todo árbol (grafo finito, conexo y acíclico) debe tener al menos un vértice de grado 1. En particular, si piensas en el caso especial de un árbol que se ramifica de manera continua a partir de una raíz, son bastante evidentes los vértices de grado 1: las hojas de la rama, que no tienen otros vértices a los que conectarse.
GPT-4o	Para demostrar que cualquier grafo finito, conexo y acíclico contiene un vértice de grado 1, podemos proceder mediante inducción sobre el número de vértices del grafo. Un grafo finito, conexo y acíclico es, por definición, un árbol. Paso base Considera el caso base con un árbol que tiene solo dos vértices, digamos V_1 y V_2, conectados por una única arista. En este caso, ambos vértices tienen grado 1, cumpliendo con la afirmación. Paso inductivo Supongamos que la afirmación es válida para todo árbol con k vértices, es decir, todo árbol con k vértices tiene al menos un vértice de grado 1. Queremos demostrar que también se cumple para un árbol con $k+1$ vértices. Considera un árbol T con $k+1$ vértices. Podemos utilizar las propiedades de acíclicos y conexos de los árboles de la siguiente manera: en un árbol, siempre existe al menos un vértice hoja, es decir, un vértice de grado 1. Justificación 1. <i>Propiedad de acíclicos:</i> Un árbol no tiene ciclos, lo que implica que no se pueden cerrar caminos. Para mantener la estructura de árbol, debe haber algún vértice que, si se remueve junto con sus conexiones, no cause la aparición de múltiples componentes, lo cual significa que tiene solo una conexión: es una hoja. 2. <i>Propiedad de minimalidad:</i> Al agregar vértices y aristas suficientes para mantener la conexión del grafo sin formar ciclos, las nuevas aristas se deben adicionar a vértices existentes. Sin embargo, remover un vértice hoja no desconecta el grafo restante (ya que un árbol tiene precisamente $k+1-1=k$ aristas), manteniendo el resto como un árbol con k vértices. Conclusión del paso inductivo Si T es un árbol con $k+1$ vértices, al remover una hoja obtenemos un subárbol con k vértices el cual, por hipótesis inductiva, tiene una hoja. Por lo tanto, el árbol original con $k+1$ vértices también debe tener al menos una hoja, lo que completa la demostración por inducción. En resumen, cualquier árbol (grafo finito, conexo y acíclico) tiene al menos un vértice de grado 1.



GPT-5.5 La afirmación necesita una pequeña precisión: si el grafo tiene un solo vértice, entonces es finito, conexo y acíclico, pero su único vértice tiene grado 0, no 1.

Por tanto, la proposición correcta es:

> Todo grafo finito, conexo y acíclico con al menos dos vértices contiene un vértice de grado 1.

Demostración: Sea G un grafo finito, conexo y acíclico con al menos dos vértices. Como G es finito, existe un camino simple de longitud máxima. Sea ese camino

$$v_0, v_1, \dots, v_k$$

donde $k \geq 1$, porque el grafo tiene al menos dos vértices y es conexo.

Afirmamos que v_0 tiene grado 1. En efecto, v_0 está unido al vértice v_1 , así que su grado es al menos 1. Supongamos ahora que v_0 tiene otro vecino $u \neq v_1$.

Hay dos posibilidades:

1. **Si u no pertenece al camino**, entonces podríamos formar un camino más largo:

$$u, v_0, v_1, \dots, v_k$$

lo cual contradice que el camino elegido era de longitud máxima.

2. **Si u pertenece al camino**, digamos $u = v_i$ con $i \geq 2$, entonces tendríamos un ciclo:

$$v_0, v_1, \dots, v_i, v_0$$

lo cual contradice que el grafo es acíclico.

Por lo tanto, v_0 no puede tener ningún vecino distinto de v_1 . Así, $\deg(v_0) = 1$. Luego el grafo contiene un vértice de grado 1. De hecho, el mismo argumento muestra que también v_k tiene grado 1.

Tabla 1 / Respuestas entregadas por diferentes LLMs a un problema elemental de matemáticas discretas. Ejercicio: identificar los problemas con las respuestas de GPT-4 y GPT-4o.

Experiencias recientes

STOC («Symposium on Theory of Computing») es una de las dos conferencias más prestigiosas en computación teórica (la otra es FOCS). A fines del 2025, enviamos un artículo junto a Andrew Krapivin, Benjamin Przybicki, y Nicolás Sanhueza-Matamala. Nuestro artículo, «Optimal and Efficient Partite Decompositions of Hypergraphs» [14], trata el problema de descomponer las aristas de un grafo como una unión disjunta de «bicliques» (i.e., grafos bipartitos completos), ver Figura 3. Estas particiones en bicliques tienen una gran variedad de aplicaciones algorítmicas, pero también han sido estudiadas por sus aplicaciones a la criptografía. Nuestro artículo trataba la versión «integral» del problema, en que por cada biclique del grafo se toma una decisión binaria: o bien se incluye (1) en la partición o no (0), y la condición es que cada arista del grafo debe pertenecer a exactamente un biclique de la partición. Curiosamente, las aplicaciones criptográficas pueden permitirse particiones «fraccionarias», en que a cada biclique se le asigna un «peso» real cualquiera entre 0 y 1, con la condición ahora de que, por cada arista, la suma de los pesos de los bicliques a los que pertenece sea igual a 1.

Uno de los usos que hago de estos modelos actualmente es preguntarles por pequeños errores en mis artículos antes de enviarlos a conferencias, y ya varias veces me han alertado de algunos que ni yo ni mis coautores habíamos notado.

Dado que nuestro artículo se dedicaba a mejorar las cotas y algoritmos para el caso integral del problema, dejamos como pregunta abierta en nuestra conclusión si algunas de nuestras técnicas o ideas se podían utilizar para obtener mejores resultados en el caso fraccionario. Meses después, fuimos contactados por investigadores de Google, quienes habían tomado nuestro artículo (y muchos otros enviados a STOC), y habían utilizado un modelo interno, que lamentablemente no es público, para atacar las preguntas abiertas. Su

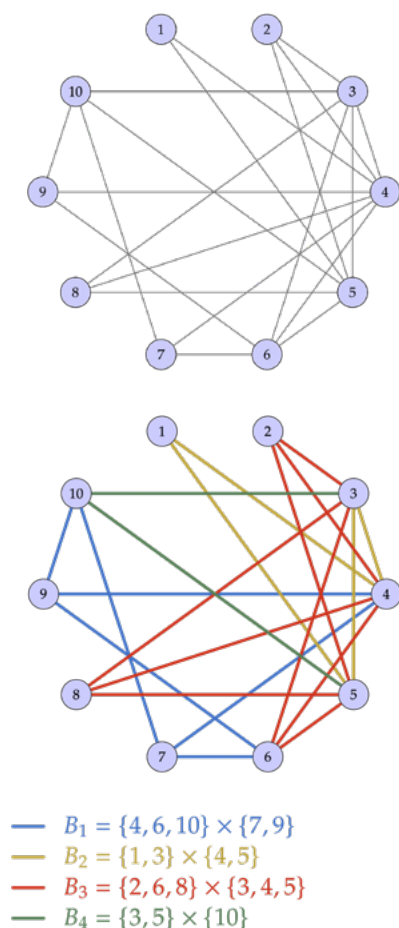


Figura 3 / Un grafo y una partición de sus aristas en tres grafos bipartitos completos.

modelo produjo una propuesta de solución a nuestro problema. La versión original que nos enviaron (un pdf con la supesta demostración de su modelo) contenía errores, y varias partes carecían de detalles. Sin embargo, había una idea interesante: reutilizar de manera distinta un lema que habíamos demostrado en el artículo con otro propósito. Luego de reunirme con mis coautores, logramos transformar la idea en una demostración correcta, y en la cual siento que el modelo fue de mucha ayuda. Esta historia de colaboración fue incluida en mayor detalle en el artículo «Accelerating scientific research with Gemini: Case studies and common techniques» [15].

Aún más recientemente, colaboré en un artículo utilizando una combinación de SAT solvers y LLMs en un problema de teoría de Ramsey [16], y personalmente cada vez encuentro más usos para los modelos de lenguaje. Desde asistencia con el diseño de figuras hasta discutir demostraciones o problemas algebraicos como mejorar los coeficientes numéricos en un argumento, que anteriormente me habrían tomado horas o incluso días.

El futuro

Una consecuencia interesante de la explosión en el uso de estos modelos en la investigación matemática y computacional es que ha dado más urgencia a preguntas como ¿por qué hacemos matemática? o ¿cuál es el objetivo de la investigación matemática? o ¿cuál es el propósito de una demostración? En mi opinión, dilucidar mejor nuestras respuestas a estas preguntas servirá para aclarar qué rol queremos que las nuevas herramientas tecnológicas que poseemos tomen en nuestras respectivas áreas de investigación. El gran matemático William Thurston dijo «La matemática no se trata de números, ecuaciones, computaciones o algoritmos; se trata de entender» [17], y creo que parte del futuro incluirá pensar en cómo utilizar nuestras herramientas para mejorar el entendimiento más que resolver preguntas concretas.

Más allá de estas generalidades, es difícil prever la magnitud del impacto de los LLMs en la investigación. No sabemos por ejemplo, si los modelos en 5 años más representarán mejoras graduales y limitadas sobre los actuales, o si llegarán a un nivel matemático inalcanzable para los humanos tal como es el caso en ajedrez o go. Al menos actualmente, es muy difícil comparar las habilidades matemáticas humanas con las de la inteligencia artificial; hay un sinnúmero de tareas que una inteligencia artificial puede realizar de manera mucho más rápida y precisa que cualquier humano, y sin embargo no hemos visto aún grandes «descubrimientos» matemáticos por parte de los LLMs; sus contribuciones autónomas hasta el momento se han limitado a resolver preguntas concretas ya existentes en la literatura. Es posible que nuestras ideas de cuáles problemas son «interesantes», o vale la pena estudiar, estén tan íntimamente ligadas a nuestra experiencia vital que no sea fácil simular este criterio en base al entrenamiento textual de los modelos. Por dar un ejemplo, Pál Turán, uno de los fundadores de la combinatoria moderna, propuso el problema de dibujar un grafo bipartito completo minimizando la cantidad de cruces entre aristas (ejercicio: dibujar el grafo de la Figura 1b ¡con un solo cruce!). Este problema, conocido como «el problema de la fábrica de ladrillos», se llama así porque Turán llegó a él gracias a una trágica circunstancia vital: durante la Segunda Guerra Mundial, el régimen nazi que operaba en Hungría forzó a Turán a trabajar en una fábrica de ladrillos, donde debía empujar vagones con ladrillos a lo largo de rieles hasta su almacenamiento. Turán notó que en los puntos en que dos rieles se cruzaban era más difícil empujar los vagones, lo que originó su curiosidad en el problema de minimización que continúa abierto hasta hoy [18]. No es imposible imaginar que una IA sugiera un problema parecido, pero sí parece claro que la elección de qué problemas sentimos naturales, profundos o bellos está todavía entrelazada con la experiencia humana.



Para concluir, mi recomendación es aceptar la incertidumbre del momento, y tomar las distintas opiniones con escepticismo, incluyendo las más vertidas en este artículo. Creo firmemente que nadie sabe bien cómo evolucionarán las cosas en los próximos años, y será importante juzgar las cosas por nosotros mismos. Vale la pena enfatizar que, para bien o para mal, las grandes empresas tienen actual-

mente incentivos económicos muy fuertes para amplificar y difundir sus contribuciones a la matemática, y no necesariamente los mejores intereses de la disciplina misma. En un momento así, quizás la actitud más sana no sea ni la fe ni el rechazo, sino una mezcla incómoda de curiosidad, escepticismo y disposición a cambiar de opinión. **B**

Addendum

Entre la escritura de la versión inicial de este artículo y su diagramación final, un modelo privado de OpenAI refutó la conjetura de Erdős sobre las distancias unitarias en el plano, un problema central en el área de geometría discreta. La noticia puede leerse en <https://openai.com/index/model-disproves-discrete-geometry-conjecture/>, donde se encuentran también enlaces a la demostración misma, y un artículo escrito por nueve matemáticos y matemáticas famosas comentando sobre la solución y su significado.

Referencias

- [1] Terence Tao, *Terence Tao at IMO 2024: AI and Mathematics*, video recording, YouTube, IMO Prize, posted August 21, 2024, accessed May 6, 2026, <https://www.youtube.com/watch?v=e049loFBnLA>.
- [2] Kenneth Appel and Wolfgang Haken, *Every planar map is four colorable*, Bull. Amer. Math. Soc. 82 (1976), no. 5, 711–712.
- [3] Kenneth Appel and Wolfgang Haken, *Every planar map is four colorable. Part I. Discharging*, Illinois J. Math. 21 (1977), no. 3, 429–490.
- [4] Kenneth Appel, Wolfgang Haken, and John Koch, *Every planar map is four colorable. Part II. Reducibility*, Illinois J. Math. 21 (1977), no. 3, 491–567.
- [5] Kenneth Appel and Wolfgang Haken, *Every planar map is four colorable*, Contemporary Mathematics, vol. 98, American Mathematical Society, Providence, RI, 1989. With the collaboration of John Koch.
- [6] Georges Gonthier, *Formal Proof—The Four-Color Theorem*, Notices Amer. Math. Soc. 55 (2008), no. 11, 1382–1393.
- [7] Joshua Brakensiek, Marijn Heule, John Mackey, and David Narváez, *The resolution of Keller’s conjecture*, J. Automat. Reason. 66 (2022), no. 3, 277–300.
- [8] Giles Gardam, “A counterexample to the unit conjecture for group rings.” Ann. of Math. (2) 194 (3) 967–979, November 2021. <https://doi.org/10.4007/annals.2021.194.3.9>
- [9] Marijn J. H. Heule and Manfred Scheucher, *Happy ending: an empty hexagon in every set of 30 points*, in *Tools and algorithms for the construction and analysis of systems* (Bernd Finkbeiner and Laura Kovács, eds.), Lecture Notes in Comput. Sci., vol. 14570, Springer, Cham, 2024, pp. 61–80.
- [10] Jeremy Avigad, *Mathematics and the formal turn*, Bull. Amer. Math. Soc. (N.S.) 61 (2024), no. 2, 225–240, DOI 10.1090/bull/1832.
- [11] Boris Alexeev, Kevin Barreto, Yanyang Li, Jared Duker Lichtman, Liam Price, Jibran Iqbal Shah, Quanyu Tang, and Terence Tao, *Primitive sets and von Mangoldt chains: Erdős Problem #1196 and beyond*, preprint, arXiv:2605.00301 [math.NT], 2026.
- [12] Nat Sothanaphan, *Resolution of Erdős Problem #728: a writeup of Aristotle’s Lean proof*, preprint, arXiv:2601.07421 [math.NT], 2026.
- [13] OpenRouter, Inc., *OpenRouter: The unified interface for LLMs*, website, accessed May 7, 2026, <https://openrouter.ai/>.
- [14] Andrew Krapivin, Benjamin Przybicki, Nicolás Sanhueza-Matamala, and Bernardo Subercaseaux, *Optimal and efficient partite decompositions of hypergraphs*, preprint, arXiv:2511.11855 [math.CO], 2025, DOI 10.48550/arXiv.2511.11855.
- [15] David P. Woodruff, Vincent Cohen-Addad, Lalit Jain, Jieming Mao, Song Zuo, MohammadHossein Bateni, Simina Branzei, Michael P. Brenner, Lin Chen, Ying Feng, Lance Fortnow, Gang Fu, Ziyi Guan, Zahra Hadizadeh, Mohammad T. Hajiaghayi, Mahdi JafariRaviz, Adel Javanmard, Karthik C. S., Ken-ichi Kawarabayashi, Ravi Kumar, Silvio Lattanzi, Euiwoong Lee, Yi Li, Ioannis Panageas, Dimitris Paparas, Benjamin Przybicki, Bernardo Subercaseaux, Ola Svensson, Shayan Taherijam, Xuan Wu, Eylon Yogev, Morteza Zadimoghaddam, Samson Zhou, Yossi Matias, James Manyika, and Vahab Mirrokni, *Accelerating scientific research with Gemini: case studies and common techniques*, preprint, arXiv:2602.03837 [cs.CL], 2026, DOI 10.48550/arXiv.2602.03837.
- [16] Benjamin Przybicki, John Mackey, Marijn J. H. Heule, and Bernardo Subercaseaux, *Doubly saturated Ramsey graphs: a case study in computer-assisted mathematical discovery*, preprint, arXiv:2604.21187 [math.CO], 2026, DOI 10.48550/arXiv.2604.21187.
- [17] Mariana Cook, *Mathematicians: An outer view of the inner world*, Princeton University Press, Princeton, NJ, 2009.
- [18] Paul Turán, *A note of welcome*, J. Graph Theory 1 (1977), no. 1, 7–9, DOI 10.1002/jgt.3190010105.

Fiscal HeredIA:

IA generativa al servicio de
la investigación penal



Richard Weber

Doctor en Investigación de Operaciones y Licenciado en Matemáticas por la Universidad de Aachen, Alemania. Profesor Titular del Departamento de Ingeniería Industrial de la Universidad de Chile. Lideró el desarrollo del sistema Fiscal HeredIA para el Ministerio Público, que fue galardonado con el premio Potencia Chile en enero de 2026 y actualmente continúa con el proyecto desde la Fiscalía de Chile.

✉ richard.weber@uchile.cl



Valentina Reyes

Ingeniera Civil Industrial por la Universidad de Chile, con doble título con la Escuela Central de Marsella, Francia. Analista criminal de la Unidad Coordinadora SAC de la División de Estudios de la Fiscalía Nacional, Ministerio Público de Chile. Se desempeñó como asistente de investigación en el marco del proyecto Fiscal HeredIA y actualmente continúa con el proyecto desde la Fiscalía de Chile.

✉ vreyes@minpublico.cl



Resumen / Desde 2023, el Ministerio Público chileno usa Fiscal Heredia, un ecosistema de inteligencia artificial que hoy emplean más de 700 fiscales y funcionarios del Ministerio Público. El sistema permite cargar documentos y audios de una investigación y formularles preguntas en lenguaje natural, con una restricción deliberada: las respuestas provienen únicamente de la evidencia cargada. En este artículo contamos cómo funcionan los módulos de análisis de texto y audio, qué problema operativo del oficio fiscal vinieron a resolver, y qué nos enseña sobre llevar la IA al sector público chileno.

Quince tomos

En una causa por fraude tributario, lavado de activos o crimen organizado, una carpeta investigativa puede llegar a los quince tomos. Cada tomo, mil doscientas a mil quinientas páginas. Hagamos la cuenta: una sola causa puede llegar a las veinte mil páginas de prueba documental. Declaraciones, peritajes, cartolas bancarias, capturas de redes sociales, comprobantes de transferencias, informes policiales, oficios.

Veinte mil páginas son, conservadoramente, cuarenta novelas. Imagine leerlas todas. Ahora imagine leerlas todas y además recordar quién dijo qué, en qué tomo, en qué página, para responder preguntas como: *¿se repiten cuentas receptoras entre los doscientos casos?*, *¿qué dijo el imputado en la escucha del 14 de marzo que contradice su declaración?*, *¿hay alguien que aparezca conectado con tres o más de los detenidos?*

Y eso es sólo el papel. A las veinte mil páginas se suman, en investigaciones de este tipo, escuchas telefónicas autorizadas judicialmente que pueden superar las quinientas horas de audio. Veinte días continuos, sin dormir, escuchando.

A la velocidad de la lectura humana, responder eso no toma semanas. Toma meses. Y el reloj corre: el artículo 247 del Código Procesal Penal otorga al fiscal dos años desde la formalización para cerrar la investigación. La jurisprudencia ha sido clara en que el “exceso de trabajo” no constituye fuerza mayor. La carga, simplemente, no es excusa.

Este artículo es sobre la herramienta que se construyó para intervenir exactamente ahí. Se llama Fiscal Heredia. Y lo más interesante no es que use inteligencia artificial, cualquiera puede hacer eso hoy, sino cómo la usa: con una restricción que la mayoría de las aplicaciones de IA evita, y que en este contexto resulta innegociable.

¿Qué es Fiscal Heredia?

El nombre no es casualidad. Heredia es el detective que Ramón Díaz Eterovic ha hecho caminar por las calles de Santiago durante más de tres décadas, resolviendo casos

a punta de caminatas, lecturas y una memoria atenta a los detalles que la ciudad va descartando. Cuando el Ministerio Público y la Universidad de Chile bautizaron así a su sistema de IA, eligieron una imagen: una herramienta que, como el detective, lee mucho, recuerda todo y conecta lo que parecía no estar conectado —y que también, casi de regalo, deja un acrónimo prolijo: HERramienta para el Estudio del Delito usando Inteligencia Artificial (HEREDIA).

Fiscal Heredia nació en 2020 como un proyecto FONDEF entre el Ministerio Público, la Universidad de Chile, la Universidad del Bío-Bío y la Universidad de los Andes. Tras un segundo FONDEF en 2024, hoy lo usan más de 700 fiscales, analistas y asesores en distintas regiones del país.

No es un modelo. Es un ecosistema modular: distintas piezas que resuelven problemas operativos distintos del oficio fiscal, montadas sobre una infraestructura común (ver Figura 1). En este artículo cubrimos dos: el análisis de texto y el de audios. Dejamos fuera los módulos de redes delictuales [1, 2] y videos.

Textos: preguntarle a una carpeta

Empecemos por lo más simple. Un fiscal tiene un archivo PDF —un informe pericial, un conjunto de comprobantes bancarios, una declaración extensa— y necesita extraer información. Carga el archivo, escribe una pregunta en lenguaje natural, recibe una respuesta. *“¿Qué cuentas bancarias se mencionan?”*, *“¿hay transferencias entre estas tres personas?”*, *“resume las declaraciones de la víctima X”*.

Detrás de esa interfaz simple ocurren dos cosas. Primero, el sistema reconoce el texto incluso cuando el PDF viene de imágenes como escaneos, fotos, capturas de pantalla. Sin ese paso, una transferencia fotografiada con el celular resulta invisible para el sistema. Segundo, un modelo de lenguaje grande (LLM por su sigla en inglés) interpreta la pregunta y elabora la respuesta a partir del texto extraído.

Acá conviene detenerse. Fiscal Heredia no opera como un buscador general ni como un asistente que responde desde Internet. Su diseño busca que las respuestas se apoyen



Figura 1 / Ecosistema Fiscal Heredia.

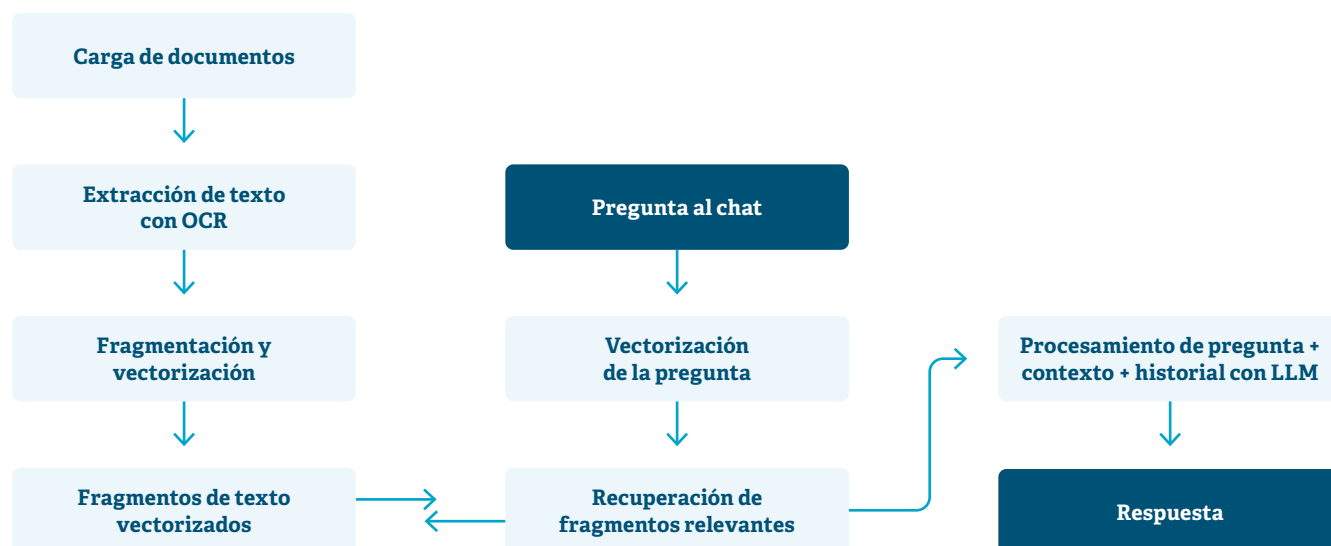


Figura 2 / Análisis de textos en Fiscal Heredia.

en los documentos incorporados al caso y que cada afirmación relevante pueda vincularse con antecedentes verificables. Esto no elimina el carácter probabilístico propio de los modelos de lenguaje, pero sí reduce el riesgo de que el sistema introduzca información ajena al corpus analizado. Para la investigación criminal, esa trazabilidad es central: los datos externos pueden ser útiles para orientar los análisis, formular hipótesis o sugerir nuevas diligencias, pero deben distinguirse claramente de la evidencia y de los antecedentes contenidos en la carpeta investigativa. La Figura 2 ilustra el funcionamiento del módulo de análisis de textos.

Cuando los documentos son muchos, el problema cambia de naturaleza. Quinientas carpetas no caben en la memoria de trabajo de ningún modelo de lenguaje actual. ¿Cómo se le pregunta a quinientas carpetas?

La idea es elegante en su descripción. Cada fragmento de texto se convierte en una representación numérica (*embedding*) que captura su significado, no sus palabras exactas. Esas representaciones se guardan en una base especializada. Cuando llega una pregunta, el sistema también la convierte en una representación numérica y busca los fragmentos cuyos *significados* se asemejan al de la consulta.



Figura 3 / Análisis de audios en Fiscal HeredIA.

Sólo esos fragmentos, los más relevantes, se le pasan al modelo para que componga la respuesta.

¿Qué es un *embedding*? Una representación numérica que ubica fragmentos de texto en un espacio donde la cercanía equivale a similitud de significado. Así, “*transferencia bancaria sospechosa*” y “*depósito atípico desde una cuenta no asociada*” terminan cerca, aunque no compartan ninguna palabra. La búsqueda deja de ser por coincidencia literal y pasa a ser por sentido.

Este patrón de recuperar primero los fragmentos pertinentes y generar después la respuesta tiene una virtud que importa especialmente acá. Cada respuesta queda anclada a fragmentos identificables de la evidencia. El o la fiscal puede pedir que el sistema le muestre exactamente de qué documentos y de qué páginas salió cada afirmación.

Audios: cuando la evidencia es una conversación

El cambio de medio plantea problemas distintos. Un audio no se lee, se escucha. Y rara vez está limpio: hay ruido, voces que se solapan, jerga, idiomas que no son los nativos. El módulo de análisis de audio lo aborda en tres movimientos [3].

Transcripción. Convertir habla en texto, robusto a ruido, con detección automática del idioma. Los idiomas soportados hoy son español, turco, chino y guaraní. Esa lista, por

sí sola, dice algo sobre el tipo de investigaciones donde el sistema entró en juego.

Diarización. Es el componente menos conocido fuera del rubro y, técnicamente, el más elegante. *Diarizar* es identificar quién habla en cada momento, sin saber quién es [4].

Traducción. Si el audio no está en español, se traduce automáticamente antes de continuar.

Una vez transcrito, diarizado y eventualmente traducido, el audio cambia de naturaleza: deja de ser un archivo que hay que escuchar de punta a punta y se convierte en material consultable. El o la fiscal puede buscar menciones, revisar intervenciones, identificar quién dijo qué o solicitar resúmenes de tramos extensos. La idea de ecosistema surge precisamente ahí: en la posibilidad de aplicar capacidades de análisis a distintos tipos de evidencia, desde documentos escritos hasta registros sonoros, sin reducir el trabajo investigativo a un solo formato. La Figura 3 muestra el funcionamiento del módulo de análisis de audios.

Lo que la herramienta no hace

¿Y si Fiscal HeredIA se equivoca?

Los modelos de lenguaje pueden producir respuestas plausibles que no estén suficientemente sustentadas en los documentos analizados. La restricción de fuente —responder a partir del contenido cargado— reduce ese riesgo, pero no lo

elimina. El sistema puede interpretar mal una pregunta, recuperar fragmentos poco pertinentes o formular una respuesta que parezca correcta sin estar adecuadamente respaldada por el documento. Por eso, la regla de uso es invariable: toda respuesta debe ser revisada y validada por una persona.

Esa validación no depende sólo de la etapa final. También importa cómo se usa la herramienta. Las respuestas son sensibles a la forma en que se pregunta: una consulta vaga tiende a producir una respuesta vaga; una pregunta precisa permite orientar mejor la búsqueda y revisar con más claridad las fuentes que sustentan el resultado. Por eso, la implementación de HeredIA no puede limitarse a otorgar acceso a una plataforma. Requiere capacitación, acompañamiento constante y espacios de retroalimentación con los usuarios, para que fiscales y analistas desarrollen criterios de uso, reconozcan los límites del sistema y aprendan a formular preguntas que sirvan realmente al trabajo investigativo.

Lo mismo ocurre con los audios. Las transcripciones automáticas no son actas. Ruido ambiental, voces solapadas, mala calidad de grabación o modismos locales pueden generar errores. El valor del sistema está en acelerar la revisión, ordenar el material y permitir búsquedas dirigidas; no en reemplazar la escucha crítica ni la comprobación humana cuando el contenido será usado en decisiones relevantes.

Por eso, la pregunta más importante cuando el Estado adopta herramientas de inteligencia artificial no es sólo qué tan buena es la tecnología. Es qué proceso humano, institucional y metodológico se construye alrededor de ella. Fiscal HeredIA debe entenderse en esa lógica: no como un sistema que impone respuestas, sino como una herramienta diseñada para acompañar el análisis, ampliar la capacidad de revisión y apoyar decisiones que siguen siendo responsabilidad de las personas.

Cuatro casos, cuatro módulos

Cifras de impacto hay muchas. Pero el valor de un sistema modular se entiende mejor al examinar casos en los que cada módulo entró en juego por separado.

Estafas masivas por Internet (2024). Doscientas carpetas de estafas cometidas con sitios web fraudulentos. Reducción del 97% del tiempo de análisis. El punto crítico: el ahorro de tiempo no es eficiencia administrativa. Es la diferencia entre que una formalización llegue oportunamente, dentro del plazo legal, o que la oportunidad se pierda.

Escuchas de audios (2024). Una investigación penal involucraba a una banda de extranjeros cuya lengua materna

Una sola causa puede llegar a las veinte mil páginas de prueba documental [...] Ahora imagine leerlas todas y además recordar quién dijo qué, en qué tomo, en qué página.

no era español. Durante meses, el material de las escuchas telefónicas autorizadas judicialmente permaneció prácticamente inaccesible: nadie en el equipo investigativo dominaba el idioma, y contratar traducción para cientos de horas de audio era inviable en los plazos disponibles. Con HeredIA, el módulo de transcripción y traducción procesó el material y lo convirtió en texto consultable en español.

Operación Influencers (2023). Una banda dedicada a enclonadas, transmitidas por redes sociales. La fiscalía tenía dos imputados. Se le pidió al sistema que sugiriera a quiénes más buscar. Sugirió trece. Tras la verificación en terreno, once resultaron parte efectiva de la banda. Importante: la herramienta no respondió “*quiénes son los culpables*” —no es lo que hace—, sino “*estos son los nodos más conectados con los ya conocidos*”. El paso a una formalización siguió siendo trabajo humano.

Proyecto Prófugos (2024). A mitad de 2023 había aproximadamente 1.200 prófugos peligrosos. La metodología tradicional capturaba entre el 1% y el 3% de los buscados. Con apoyo del módulo de redes, la cifra subió al 40% dentro de un año, es decir más de cuatrocientas personas resultaron detenidas. Lo notable no es sólo el salto porcentual: es que una capacidad analítica que antes simplemente no existía pasó a estar disponible para una región completa.

Lo replicable

Fiscal HeredIA ha recibido reconocimientos públicos por su aporte a la modernización del trabajo investigativo. En 2025, el Ministerio de Hacienda lo destacó como una iniciativa relevante de inteligencia artificial aplicada a la transformación digital del Estado, en el marco del encuentro “Para la transformación del Estado: personas, confianza y futuro”. Luego, recibió el Premio Nacional de Inteligencia Artificial Chile Potencia 2026, organizado por el Ministerio de Ciencia, Tecnología, Conocimiento e Innovación junto a Fundación País Digital, en la categoría “Organizaciones”. Además, figura entre los seis finalistas del IFORS Prize for OR in Development 2026, que se presentará en la conferencia



de IFORS en Viena, en julio de 2026. Son hitos relevantes. Pero no son la conclusión interesante de este artículo.

La conclusión interesante es qué de esto es replicable y qué no. Las técnicas que usa HeredIA están disponibles para cualquiera. Reconocimiento de texto, modelos de lenguaje, embeddings, transcripción automática, diarización: todo está documentado, todo se puede licenciar o usar mediante APIs. No son, por sí solas, lo que hizo posible este sistema.

Lo que lo hizo posible fue una combinación específica que sí es replicable, pero exige tiempo: una alianza Universidad-Estado sostenida durante más de cinco años a través de dos proyectos FONDEF consecutivos; un equipo desarrollador estable dentro de la institución, capaz de entender tanto el oficio fiscal como las técnicas; casos de uso bien acotados antes de escalar; y una cultura de validación humana incorporada al flujo de trabajo desde el primer día.

Pero hay una pieza adicional, menos visible y probablemente más decisiva: escuchar la operación. HeredIA no nace de imaginar desde afuera cómo debería trabajar una fiscalía, sino de observar dónde se pierde tiempo, qué tareas se repiten, qué información cuesta encontrar, qué dolores enfrentan fiscales, analistas y equipos investigativos cuando deben revisar miles de páginas, audios, declaraciones o antecedentes dispersos. Esa escucha permite que la tecnología no llegue como una imposición, sino como una herramienta que acompaña prácticas existentes, las ordena y las potencia.

Esa es la lección útil para otros despliegues de IA en el sector público chileno. La parte difícil no es sólo construir el modelo ni conectar una API. Es sostener una relación de trabajo con los usuarios, traducir problemas operativos en funcionalidades concretas, capacitar de manera continua, recoger retroalimentación y ajustar la herramienta a medida que cambia la práctica. Cualquiera de esas piezas faltando, el sistema difícilmente llega a producción o se vuelve parte del trabajo cotidiano.

Referencias

- [1] Troncoso F, Weber R (2024) The Steiner tree Prosecutor: Revealing and disrupting criminal networks through a single suspect. *PLoS ONE* 19(12): e0312827.
- [2] Troncoso, F., Weber, R. (2025): Using AI to Identify Criminal Gang Members in Chile. *IFORS Newsletter* 20, No. 4, 9-12. <https://www.ifors.org/december-2025-issue/>.
- [3] Bain, M., Huh, J., Han, T., Zisserman, A. (2023): WhisperX: Time-Accurate Speech Transcription of Long-Form Audio. *Proc. Interspeech* 2023, 4489-4493.
- [4] Park, T. J., Kanda, N., Dimitriadis, D., Han, K. J., Watanabe, S. Narayanan, S. (2022): A Review of Speaker Diarization: Recent Advances with Deep Learning. *Computer Speech & Language* 72.
- [5] Fiscalía 2026: Cuenta Pública 2026 del Fiscal Nacional Ángel Valencia Vásquez. <https://www.youtube.com/watch?v=v2hUJRqmnQ>.

El ahorro de tiempo no es eficiencia administrativa. Es la diferencia entre que una formalización llegue dentro del plazo legal, o que la oportunidad se pierda.



Figura 4 / Premios que obtuvo Fiscal HeredIA, presentados por el Fiscal Nacional Ángel Valencia [5; minuto 44:20].

Esa es, finalmente, la invitación que deja Fiscal HeredIA. Hay mucha inteligencia artificial en laboratorios, papers, prototipos y pruebas de concepto; el desafío es llevarla adonde puede producir un impacto público concreto. Salir de la academia no significa abandonar el rigor, sino ponerlo a trabajar en problemas reales: expedientes imposibles de leer a tiempo, audios que nadie alcanza a revisar completos, información dispersa que debe ordenarse antes de que venza un plazo legal. Fiscal HeredIA muestra que esa transferencia es posible cuando la investigación se encuentra con la operación, cuando los equipos técnicos escuchan a quienes usan la herramienta y cuando el objetivo no es demostrar que una tecnología funciona, sino transformar una práctica institucional. **B**

El cerebro y sus interrogantes:

preguntas abiertas en
neurociencia computacional



Felipe Bravo-Márquez

PhD por la Universidad de Waikato, Nueva Zelanda. Profesor Asociado del Departamento de Ciencias de la Computación de la Universidad de Chile y Director de la Iniciativa de Datos e Inteligencia Artificial. Investigador Principal del Centro Nacional de Inteligencia Artificial e Investigador del Instituto Milenio Fundamentos de los Datos. Sus intereses incluyen NLP, aprendizaje automático, inteligencia artificial y recuperación de información.

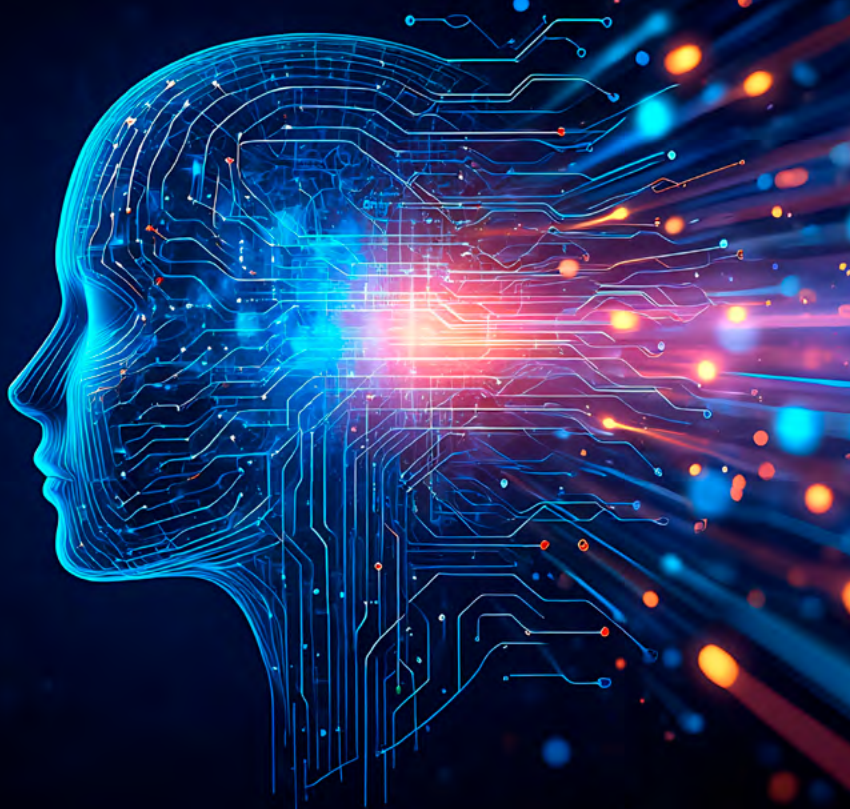
✉ fbravo@dcc.uchile.cl



Rodrigo Carrasco-Davis

PhD por la Gatsby Computational Neuroscience Unit (UCL). Actualmente Investigador Postdoctoral en el Princeton Neuroscience Institute, en el laboratorio de neurociencia de control cognitivo. Su investigación se sitúa en la intersección entre neurociencia computacional y aprendizaje automático, con foco en meta-aprendizaje y sistemas de memoria.

✉ rodrigo.cd@princeton.edu





Resumen / En los últimos años, la inteligencia artificial ha tenido un desarrollo vertiginoso y ha permeado muchos ámbitos de nuestra vida. A pesar del éxito que ha tenido, quedan múltiples interrogantes científicas: ¿cómo podrían los sistemas de IA interactuar, aprender y decidir de forma continua y autónoma según el ambiente que los rodea? ¿Cómo podrían coexistir con un cuerpo físico? ¿O cómo podrían generar motivaciones propias? Algunas de estas cualidades están presentes en la forma en que nosotros, los humanos y otros animales, nos desarrollamos durante nuestra vida, por lo que resulta natural volcar nuestra atención hacia cómo funciona el cerebro para intentar contestar estas interrogantes.

En este artículo presentamos nociones básicas de neurociencia computacional, un área que históricamente ha buscado formalizar matemáticamente los mecanismos que rigen la dinámica del cerebro. Introduciremos conceptos básicos del funcionamiento del cerebro bajo el lente de modelos computacionales, además de destacar preguntas actuales en neurociencia que podrían servir para inspirar avances en inteligencia artificial, o bien ser inspiradas por ellos.

Neurociencia computacional

La neurociencia como tal es un área de investigación dedicada a estudiar el cerebro, aunque este tipo de investigación puede tomar muchas formas. Por ejemplo, uno podría argumentar que la psicología es una forma de investigar el cerebro a través del comportamiento humano. Otro ejemplo podría ser la neurología, que tiene un enfoque médico, con el objetivo de identificar y tratar patologías asociadas al cerebro.

En contraste con estas áreas, la neurociencia computacional busca describir lo observado y generado por el cerebro de forma matemática, es decir, a través de ecuaciones que puedan usarse para simular, predecir y comprender la dinámica de lo observado. Esta descripción puede tener muchos niveles de abstracción, como veremos más adelante. Por ejemplo, existen ecuaciones matemáticas que describen distintos aspectos del cerebro a distintas escalas y niveles computacionales: cómo varía el voltaje de una neurona en el tiempo, cómo un grupo de neuronas reacciona a partir de la actividad de otras, cuánta información y de qué tipo se propaga a través de distintas zonas del cerebro, cómo el cerebro se reconfigura con el tiempo durante el aprendizaje, qué son las memorias y cómo se consolidan, o cómo el cerebro genera acciones y comportamientos.

Lograr este tipo de descripción matemática requiere, muchas veces, abstraerse de ciertos detalles (por ejemplo, a veces se asume que todas las neuronas son “iguales”, aunque realmente no lo son). Esta abstracción es la que permite emplear distintas herramientas matemáticas e ingenieriles, como teoría de circuitos eléctricos, teoría y control de sistemas dinámicos, teoría de la información, teoría de la computación, mecánica estadística y lingüísti-

ca computacional, entre muchas otras. Modelar aspectos del cerebro con estas herramientas nos permite, por ejemplo, simular el sistema en un computador, intervenir en él y analizar las consecuencias de modificar ciertas variables para entender cómo y por qué funciona. Finalmente, el formalismo matemático puede traducirse, y muchas veces coincide, con el utilizado en inteligencia artificial y aprendizaje automático en general. Existe, en otras palabras, un lenguaje común entre la neurociencia computacional y el aprendizaje automático que convierte esa intersección en un espacio fructífero para entender la inteligencia como un fenómeno natural: uno que puede observarse tanto en el cerebro como en agentes de inteligencia artificial, quizás regido por principios o leyes similares. Existe un debate enorme acerca de qué significa *comprender* un sistema complejo como el cerebro o un agente artificial, pero volveremos a esta discusión al final del artículo.

En las siguientes secciones hablaremos con más detalle de cómo funciona el cerebro y de cómo se modelan muchos de estos sistemas en neurociencia computacional. Comenzaremos de menos a más: desde las unidades de cómputo, las neuronas; pasaremos por la neuroplasticidad; luego a entender redes más grandes; y finalmente las partes y funciones más complejas del cerebro, como la toma de decisiones, el aprendizaje y la memoria.

Neuronas como unidad de cómputo del cerebro

El cerebro está hecho de células, como el resto del cuerpo. Sin embargo, la mayoría de las células están diseñadas para construir tejidos, transportar oxígeno o realizar funciones químicas específicas. Las neuronas son diferentes: están especializadas en la comunicación.

Nuestra comprensión moderna de las neuronas comenzó a finales del siglo XIX. Camillo Golgi desarrolló una técnica de tinción que colorea al azar una pequeña fracción de neuronas, permitiendo observar su forma completa por primera vez. Al estudiar estas estructuras ramificadas, Golgi concluyó que el cerebro era una red continua, similar a una malla de fibras fusionadas. Santiago Ramón y Cajal utilizó el mismo método, mostrando que las neuronas son células individuales que se comunican entre sí en puntos de contacto especializados. Esta idea, conocida como la *doctrina de la neurona*, es uno de los fundamentos de la neurociencia moderna.

En términos simples, una neurona es una célula cuyo trabajo es *recibir señales, combinarlas y transmitir nuevas señales*. Estructuralmente, una neurona tiene tres partes principales (ver Figura 1):

- **Dendritas:** estructuras ramificadas que reciben señales de otras neuronas.
- **Cuerpo celular (soma):** integra las señales que llegan desde las dendritas.
- **Axón:** una prolongación que transmite señales hacia otras neuronas.

Una neurona puede conectarse con miles de otras neuronas y recibir conexiones de muchas más. Las neuronas no se tocan directamente. Se comunican a través de pequeñas uniones llamadas *sinapsis*. Está formada por el extremo del axón de la neurona emisora, un pequeño espacio entre las células y una región de la membrana de la neurona receptora.

La transmisión de información dentro de una neurona entre neuronas ocurre de forma química y eléctrica. En una misma neurona, existen distintos iones con carga eléctrica que se desplazan a través de la membrana celular, como sodio, potasio, calcio y cloro. La membrana de la neurona regula qué iones pueden entrar o salir, generando una diferencia de voltaje entre el interior y el exterior de la célula. Este voltaje se conoce como *potencial de membrana*. Cuando la neurona está en reposo, este voltaje se mantiene relativamente estable. Sin embargo, las señales que recibe de otras neuronas pueden modificarlo ligeramente. Algunas lo aumentan y otras lo disminuyen.

Si suficientes señales excitatorias llegan en un corto intervalo de tiempo, el voltaje puede superar un umbral crítico. Cuando eso ocurre, la neurona produce un pulso eléctrico breve llamado *potencial de acción*, o simplemente *spike*. Un *spike* es un evento muy rápido, dura cerca

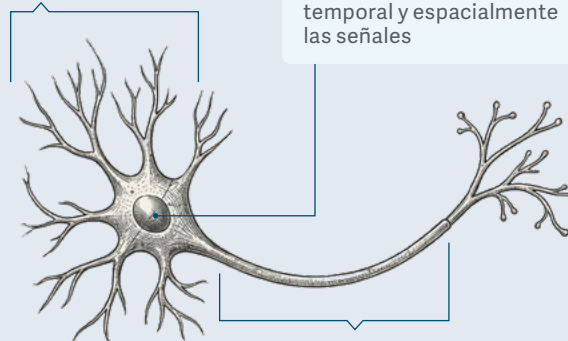
El hardware: La neurona

Dendritas

Input. Estructuras ramificadas que reciben señales de otras neuronas.

Soma

Procesador. El cuerpo celular que integra temporal y espacialmente las señales



Axón

Output. Prolongación que transmite la señal procesada hacia la red

El cerebro humano contiene ~ 86 mil millones de neuronas y cientos de billones de conexiones.

Basado en la **Doctrina de la Neurona** (Santiago Ramón y Cajal): son unidades discretas, no una malla continua.

Figura 1 / La neurona: las dendritas reciben señales de otras neuronas conectadas a ella. Dependiendo de la magnitud de ese estímulo, la neurona puede generar un potencial de acción o spike desde el Soma. Este se propaga a lo largo del axón hasta las vesículas sinápticas en su terminal, donde se libera el neurotransmisor que transmite la señal a las neuronas siguientes, formando la sinapsis.

de un milisegundo y funciona bajo un principio de *todo o nada*: ocurre o no ocurre. Por esta razón, se parece más a un pulso digital que a una señal analógica. La información no está en la amplitud del pulso, sino en *cuándo ocurre* y *con qué frecuencia*.

Modelos de una neurona

Como mencionamos anteriormente, uno de los primeros esfuerzos de la neurociencia computacional fue desarrollar

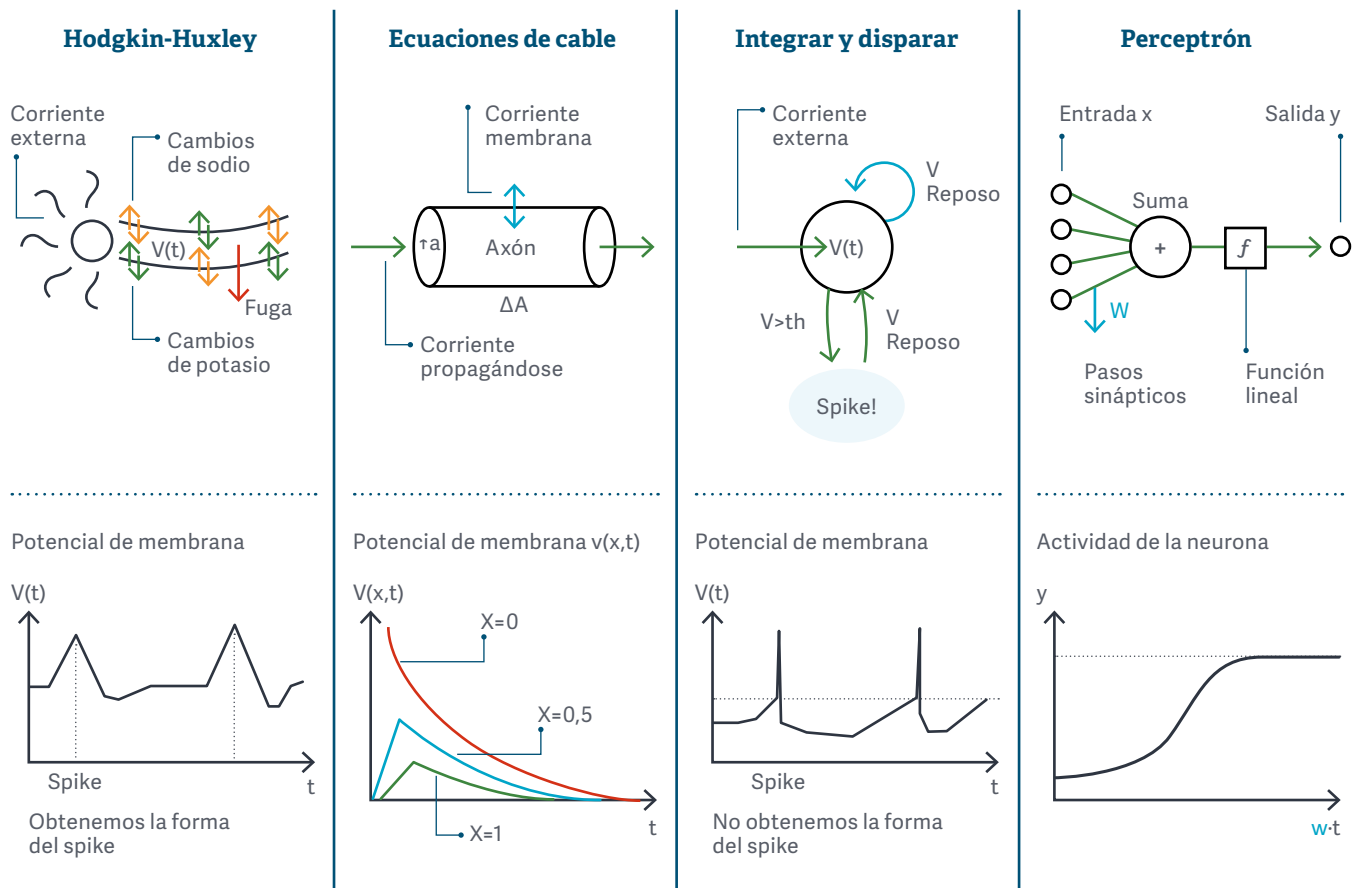


Figura 2 / Modelos matemáticos del comportamiento de una neurona. **Hodgkin-Huxley:** sistema de ecuaciones diferenciales acopladas que describe el potencial de membrana mediante el modelado de compuertas iónicas de sodio y potasio, y una corriente de fuga. Es el modelo más detallado biofísicamente y reproduce la forma precisa de los spikes. **Ecuación de cable:** describe cómo se propaga el potencial a lo largo del axón, expresándolo en función de la posición y el tiempo. Es útil cuando la geometría espacial de la neurona importa. **Integrar y disparar:** el potencial de membrana tiende al reposo, pero aumenta al recibir una corriente externa; si alcanza un umbral, se emite un spike y el potencial se resetea al valor de reposo. A diferencia de Hodgkin-Huxley, el spike no tiene forma biofísica explícita sino que se define como un evento discreto. **Perceptrón:** abstrae toda restricción biológica y modela la salida de otras neuronas como tasas de disparo continuas, las pondera por pesos sinápticos, las suma y aplica una función de activación no lineal. Es el modelo más simplificado y el que da lugar a las redes neuronales artificiales.

modelos matemáticos que describieran el comportamiento eléctrico de una neurona (ver Figura 2).

Ecuaciones de Hodgkin-Huxley: Las ecuaciones de Hodgkin-Huxley (HH) [1] describen cómo fluyen los iones de sodio y potasio a través de los canales en la membrana neuronal. Se construyen usando teoría de circuitos eléctricos: el potencial de membrana se modela como un capacitor, y el flujo de iones como corrientes que entran y salen de él, formando un sistema de ecuaciones diferenciales acopladas. Son computacionalmente costosas y tienen muchos parámetros, pero describen con gran precisión el comportamiento eléctrico de la neurona, incluyendo la forma y dinámica de los spikes.

Ecuaciones de cable: Las neuronas pueden tener geometrías complejas, especialmente en el axón y las dendritas. Las ecuaciones de cable modelan la neurona como un cilindro con resistencia interna y capacitancia de membrana, describiendo cómo el voltaje se propaga y decae a lo largo de la estructura mediante una ecuación diferencial parcial en espacio y tiempo [2]. Son útiles cuando la morfología importa; por ejemplo, para entender cómo una señal en una dendrita distal llega atenuada al soma.

Integrar-y-Disparar: El modelo Integrar-y-Disparar (LIF, por sus siglas en inglés) puede entenderse como una versión simplificada de HH [3]. El potencial de la neurona tiende al reposo, pero aumenta al recibir una corriente externa. Una

vez que alcanza un umbral, se emite un *spike* reseteando el voltaje manualmente, sin modelar explícitamente los canales iónicos. Esto simplifica considerablemente la dinámica y reduce el costo computacional.

Neurona artificial (Perceptrón): El perceptrón es el modelo más abstracto de todos. Simplemente suma las señales entrantes ponderadas y las pasa por una función de activación no lineal, sin modelar canales iónicos, propagación espacial ni *spikes*. Esta simplicidad es precisamente su fortaleza: permite escribir matemáticamente redes de millones de neuronas conectadas, analizarlas y simularlas en un computador convencional. En contraste, simular una red equivalente con neuronas HH requeriría un poder de cómputo enormemente mayor para apenas cientos o miles de neuronas. El costo de esta abstracción es la pérdida de la descripción biofísica de las neuronas: para replicar el comportamiento de una neurona biológica con todo su detalle, se requieren muchas neuronas perceptrón [4, 5]. Muchos modelos modernos de redes neuronales asumen este tipo de neuronas simplificadas, y es un área de investigación activa entender qué efectos no pueden observarse en redes debido a esta simplificación [6, 7].

Para entender cómo las neuronas se influyen entre sí, debemos observar con más detalle los espacios de contacto entre ellas, llamados sinapsis, que juegan un rol fundamental en el aprendizaje.

Sinapsis

Cuando un *spike* llega al final del axón, desencadena un proceso llamado *transmisión sináptica*. Pequeñas vesículas liberan moléculas llamadas neurotransmisores en el espacio entre las neuronas. Estas moléculas atraviesan la brecha entre neuronas en una fracción de milisegundo y se unen a receptores en la neurona siguiente (ver Figura 3). Al activarse, estos receptores modifican las propiedades eléctricas de la membrana de la neurona receptora [8].

El resultado es un pequeño cambio en el voltaje de esa neurona. Algunas sinapsis aumentan el voltaje y hacen más probable que la neurona dispare. Estas se conocen como *sinapsis excitatorias*. Otras lo disminuyen y reducen la probabilidad de disparo. Estas son *sinapsis inhibitorias* [9].

Una sinapsis individual suele ser muy débil. Rara vez provoca por sí sola que una neurona dispare. Sin embargo, cada neurona recibe miles de señales de este tipo. La neurona integra todos estos pequeños cambios de voltaje a lo largo del tiempo y del espacio. Como vimos anteriormente en los modelos HH e integrar-y-disparar, la neurona genera un *spike*. Si no, permanece en potencial de reposo.

La interfaz de red: Sinapsis

Una sinapsis individual es demasiado débil para forzar una acción. La computación real ocurre al integrar miles de micro-señales en el cerebro. Las neuronas modulan continuamente la probabilidad de que otras neuronas se activen.

→ **Excitatorias (+):** Aumentan el voltaje, acercando al umbral de disparo.

→ **Inhibitorias (-):** Disminuyen el voltaje, suprimiendo el disparo.

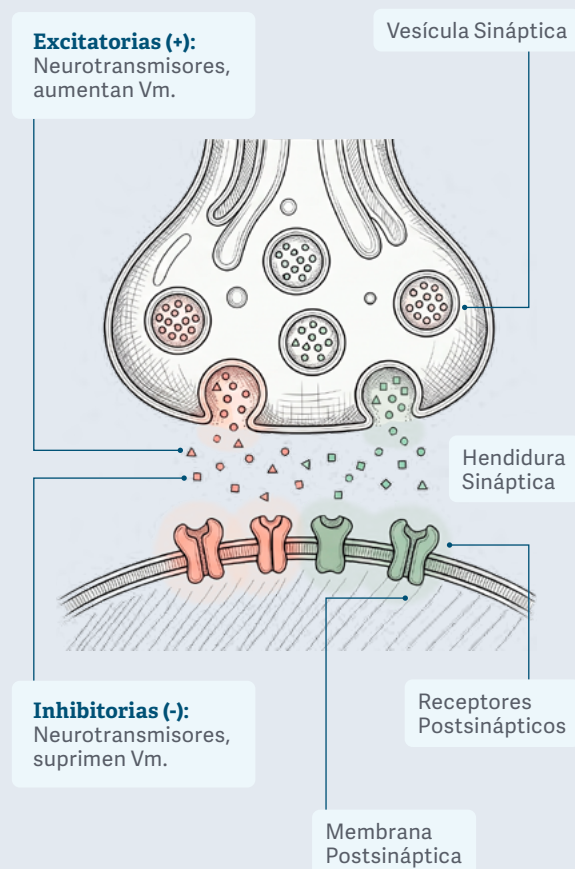


Figura 3 / Sinapsis: espacio de conexión entre neuronas.

Cuando la neurona presináptica genera un *spike*, libera neurotransmisores desde la vesícula sináptica hacia la neurona postsináptica, modificando su potencial de membrana. Los neurotransmisores pueden ser excitatorios, acercando el potencial al umbral de disparo, o inhibitorios, alejándolo. Típicamente una sinapsis es enteramente excitatoria o inhibitoria, aunque existe evidencia de sinapsis que liberan ambos tipos, como la que se muestra en este esquema [10].



La plasticidad sináptica es uno de los principales mecanismos que permiten aprender cosas nuevas, pero no es el único.

Plasticidad

Las conexiones entre neuronas no son fijas. Cambian con la experiencia. Esta propiedad se conoce como *plasticidad*. Cuando grupos de neuronas se activan juntos de manera repetida, las sinapsis entre ellas pueden fortalecerse. Cuando la actividad no está coordinada, esas conexiones pueden debilitarse. Con el tiempo, estos pequeños ajustes modifican cómo fluye la actividad a través de la red.

Una de las ideas más influyentes para describir este proceso se resume en una frase simple:

“Las neuronas que se activan juntas tienden a fortalecer su conexión”.

Este principio se conoce como *aprendizaje hebbiano* [11], propuesto por Donald Hebb en los años cuarenta. Décadas de experimentos han identificado mecanismos biológicos que implementan esta idea. Uno de los más estudiados es la *plasticidad dependiente del tiempo de disparo* (STDP) [12, 13], donde el momento preciso en que dos neuronas disparan determina si su conexión se fortalece o se debilita. A nivel físico, estos cambios implican modificaciones en la sinapsis: variaciones en la cantidad de receptores, en la liberación de neurotransmisores, en el tamaño de la conexión o incluso en el número de sinapsis entre neuronas.

Con el tiempo, estos cambios microscópicos reorganizan los patrones de actividad en la red. Grupos de neuronas comienzan a activarse de manera consistente juntos, formando lo que los neurocientíficos llaman *circuitos o redes neuronales*. Así, el aprendizaje no se almacena en una neurona específica ni en un registro centralizado. Se distribuye en los patrones de conexión entre muchas neuronas.

Un ejemplo simple es el reconocimiento de un rostro. La primera vez que vemos una cara nueva, la actividad en el sistema visual es débil e inestable. Con la repetición, las neuronas que representan sus rasgos tienden a activarse juntas, sus sinapsis se fortalecen y el patrón se estabiliza. Con el tiempo, basta una pequeña parte de la señal para activar el resto del patrón, y reconocemos a la persona casi instantáneamente. A través de miles de millones de estos cambios microscópicos, el cerebro ajusta continuamente su estructura para interpretar mejor el mundo y guiar la acción.

La plasticidad sináptica es uno de los principales mecanismos que permiten aprender cosas nuevas, pero no es el único. Otros ejemplos incluyen la plasticidad intrínseca (cambios en la excitabilidad interna de la neurona) [14], la neurogénesis (creación de neuronas nuevas, especialmente en el hipocampo) [15], y la modulación por neurotransmisores como la dopamina, la serotonina o la acetilcolina [16].

Si bien se conocen varios mecanismos locales de plasticidad sináptica, como el aprendizaje hebbiano, sigue siendo una incógnita cómo el cerebro aprende a nivel global, coordinando el cambio de muchas neuronas a la vez. Para entender esto mejor, veamos qué ocurre en las redes neuronales.

Redes neuronales artificiales y biológicas

El cerebro humano contiene alrededor de 86 mil millones de neuronas de distintos tipos y cientos de billones de conexiones entre ellas. En esta sección aplicaremos una abstracción adicional para discutir modelos de redes neuronales, que sacrifican detalle biológico a cambio de simplicidad y mayor claridad conceptual. Adoptaremos los siguientes supuestos:

- **Un solo tipo de neurona simplificada.** Supondremos que todas las neuronas siguen el mismo modelo, en particular el perceptrón, donde la salida de una neurona se conecta como entrada a otra. Existen modelos de redes neuronales donde este no es el caso [17].
- **Salida continua.** Supondremos que la salida de cada neurona es un número continuo, interpretable como análogo a la frecuencia de disparo de spikes [18].
- **Las neuronas como único sustrato de cómputo.** Supondremos que todo el procesamiento ocurre en las neuronas, dejando de lado otras células como los astrocitos y distintos tipos de glía. Aunque durante mucho tiempo se consideraron principalmente células de soporte, hoy se investiga activamente su posible rol en la modulación de la actividad neuronal y en procesos de plasticidad [19, 20].

Como consecuencia de estas simplificaciones, cada neurona individual no es capaz de resolver problemas complejos por sí sola: es el sistema en su conjunto el que tiene la capacidad de procesar y responder a información compleja. En otras palabras, las redes neuronales son sistemas distribuidos donde la capacidad computacional emerge de la cooperación de muchas neuronas.

Quizás la forma más sencilla de conectar neuronas para formar una red es una red neuronal de dos capas (ver Figura 4): una capa de entrada, que recibe las señales, y una capa de

salida, que produce la respuesta. Este tipo de arquitectura ha sido utilizada tanto en neurociencia como en aprendizaje automático debido a su simplicidad y a su capacidad para resolver problemas complejos, ya que es un aproximador universal de funciones. Luego veremos que existen redes con muchas capas, y con conectividades más complejas.

Si bien el supuesto de que cada neurona es estructuralmente similar es razonable, su conectividad resulta ser más específica dependiendo del área del cerebro que estemos estudiando. Esto es importante porque la conectividad pre-define qué neuronas están conectadas entre sí, y son esas conexiones las que están sujetas a plasticidad durante el aprendizaje. A continuación describiremos diferentes regiones del cerebro que cumplen distintos roles y que, por tanto, presentan arquitecturas de conectividad distintas según las demandas de cada función. Finalmente, discutiremos el aprendizaje en redes neuronales biológicas: cómo las reglas locales como la hebbiana o la STDP, si bien capturan aspectos importantes de la plasticidad, resultan insuficientes para explicar el aprendizaje en redes grandes.

Organización del cerebro

Aunque el cerebro parece un solo órgano, está compuesto por múltiples regiones con distintos tipos de neuronas, patrones de conectividad y funciones. Una forma útil de organizarlas es distinguir tres grandes sistemas presentes en todos los mamíferos [21].

El *tronco encefálico*, ubicado en la base del cerebro, es una estructura evolutivamente antigua que controla funciones esenciales como la respiración y el ritmo cardíaco. Computacionalmente, su rol principal es mantener el sistema activo y regular su estado global mediante estructuras como la formación reticular, que modula el nivel de alerta.

Rodeando el tronco encefálico, el *sistema límbico* incluye la amígdala, el hipocampo y el hipotálamo. Estas regiones están involucradas en emoción, motivación y memoria, y contribuyen señales de valor y refuerzo que influyen en qué experiencias se aprenden y qué conductas se repiten.

Cubriendo la superficie del cerebro, el *neocórtex* es el principal sistema para la percepción, representación del mundo, planificación y control voluntario de acciones. Todos los mamíferos la poseen, pero en humanos está especialmente expandida en áreas asociadas al lenguaje, la abstracción y la toma de decisiones.

Además de estos tres sistemas, algunas estructuras cumplen roles especialmente relevantes para el aprendizaje: el

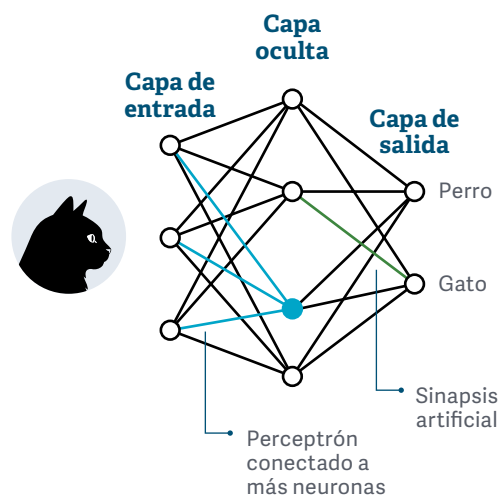


Figura 4 / Red neuronal artificial: compuesta por capas de neuronas, en este caso tres: entrada, oculta y salida, donde cada neurona (perceptrón) recibe como entrada las salidas de la capa anterior, escaladas por pesos sinápticos, que son las conexiones entre neuronas.

tálamo regula qué información sensorial llega a la corteza y cuándo; el *hipocampo* forma nuevas memorias episódicas y participa en su consolidación de conocimiento en el neocórtex; el *cerebelo* está involucrado en predicción y aprendizaje basado en error; y los *ganglios basales* participan en la selección de acciones y en aprendizaje por refuerzo. A continuación exploraremos algunas de estas estructuras, junto con modelos de redes neuronales que se han propuesto para describir su actividad.

El neocórtex: la principal máquina de aprendizaje del cerebro

El neocórtex es probablemente la estructura más notable del cerebro. A pesar de participar en funciones tan diversas como visión, audición, lenguaje, planificación motora y razonamiento abstracto, su organización interna es sorprendentemente uniforme. En casi todas sus áreas se repite el mismo microcircuito de seis capas [22].

Bajo el microscopio, la corteza visual, auditiva, motora y prefrontal se ven muy similares. Lo que cambia no es la estructura básica del circuito, sino los tipos de señales que reciben y las proyecciones que envían a otras regiones del cerebro.

Esto sugiere una idea importante desde el punto de vista computacional: el neocórtex podría implementar un mecanismo general de aprendizaje que luego se especializa según sus conexiones y los datos que recibe. En lugar de tener

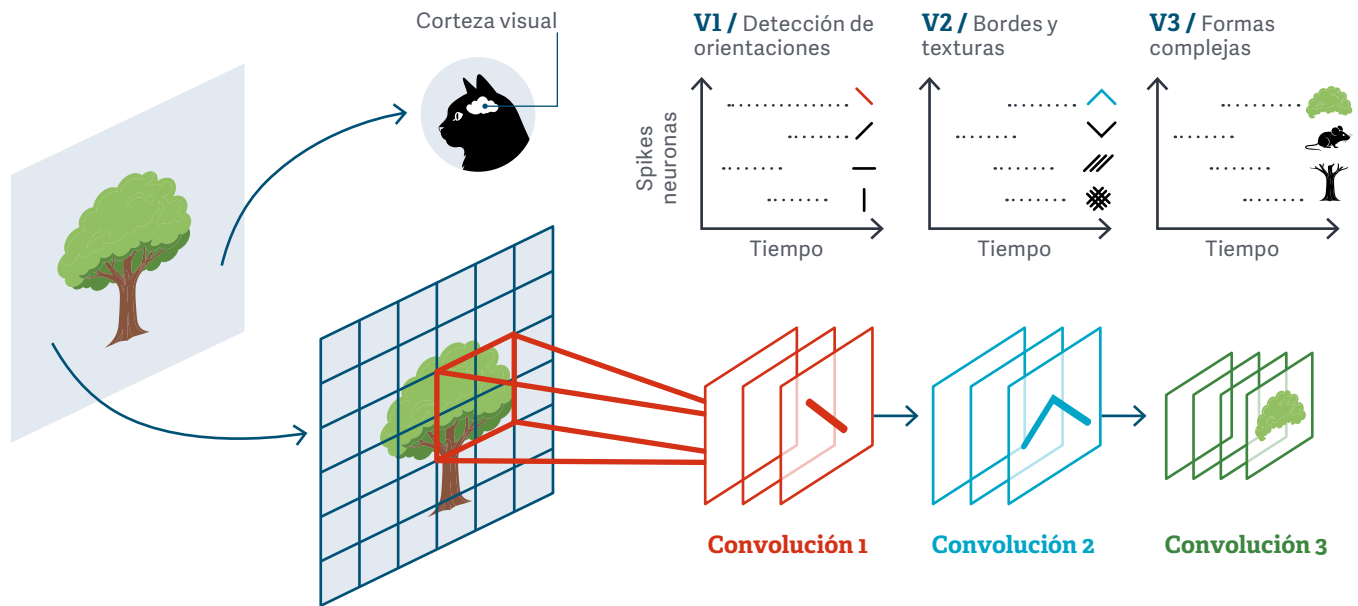


Figura 5 / Red convolucional como modelo del sistema visual: las redes convolucionales son uno de los modelos más utilizados para describir el sistema visual en mamíferos. En estas redes, cada neurona actúa como un filtro local que recorre la imagen de entrada y genera representaciones de lo que detecta. Estas representaciones tienen mucho en común con las observadas en la corteza visual: a medida que se avanza por las capas de la red, las representaciones progresan de características simples, como bordes y esquinas, hacia formas y texturas más abstractas, reflejando la jerarquía visual del cerebro. Esta correspondencia entre redes convolucionales y corteza visual puede explorarse directamente en plataformas como BrainScore, que permiten comparar modelos computacionales con registros neuronales reales.

módulos completamente distintos para visión, lenguaje o razonamiento, el cerebro reutiliza una arquitectura común que se entrena con diferentes flujos de información.

Pequeñas regiones del neocórtex, a veces llamadas columnas corticales, pueden entenderse como unidades locales de aprendizaje [23]. Cada una recibe señales de entrada, detecta regularidades y genera predicciones sobre lo que es probable observar a continuación. Estas predicciones se propagan tanto hacia áreas superiores como hacia áreas inferiores de la jerarquía cortical, generando un sistema fuertemente recursivo.

Esta organización nos lleva naturalmente a una forma moderna de entender la percepción.

Corteza visual

El sistema visual es quizás el ejemplo más estudiado de la conexión entre neurociencia computacional y aprendizaje automático. Hubel y Wiesel [24] descubrieron en la década de 1960 que las neuronas de la corteza visual

primaria V1 (de un gato) responden selectivamente a bordes orientados en posiciones específicas del campo visual, comportándose esencialmente como filtros locales. Esta observación inspiró a Fukushima [25] a proponer el Neocognitron, una red jerárquica que detectaba patrones visuales de forma progresivamente más abstracta. Más tarde, LeCun et al. [26] formalizaron esta idea en las redes convolucionales (CNNs), demostrando su utilidad práctica en tareas de clasificación.

En una red convolucional, las neuronas analizan localmente distintas partes de la imagen de entrada, formando filtros que recuerdan a los detectores de bordes de Hubel y Wiesel. En el cerebro, el proceso análogo comienza en la retina, cuya actividad se transmite a través del tálamo hacia V1, y luego a áreas visuales de mayor nivel como V2 y la corteza inferotemporal (IT), que representan información visual cada vez más abstracta. Esta jerarquía biológica se refleja naturalmente en la arquitectura de capas de una CNN (ver Figura 5).

La similitud entre ambos sistemas ha dado lugar a comparaciones directas. Plataformas como BrainScore¹ permiten

1 <https://www.brain-score.org/>

evaluar qué tan bien los modelos computacionales reproducen grabaciones neuronales reales de la corteza visual. Hoy existen modelos con distintos grados de realismo biológico, contruidos con propósitos diversos: clasificación de imágenes, compresión, o maximización de la transmisión de información, este último con una conexión más directa a teorías de codificación eficiente en el cerebro.

Cabe destacar que no todo el modelamiento del sistema visual se hace con redes neuronales. Un ejemplo muy influyente es el trabajo de Olshausen y Field [27], quienes mostraron que los filtros observados en V1 pueden obtenerse a partir de un principio teórico: si se entrena un modelo para representar imágenes naturales de forma eficiente y dispersa (*sparse coding*), los filtros que emergen del entrenamiento se parecen notablemente a los campos receptivos de las células simples de V1. Esta idea se enmarca en la teoría de *efficient coding*, cuya formulación original se atribuye a Horace Barlow [28, 29], quien propuso que el sistema nervioso sensorial está organizado para transmitir la mayor cantidad de información posible minimizando la redundancia de información propagada por los *spikes*. La teoría ha sido aplicada con éxito más allá del sistema visual, incluyendo el sistema auditivo y otras áreas sensoriales, y sigue siendo un marco activo de investigación en neurociencia computacional.

Corteza prefrontal

La corteza prefrontal (PFC, del inglés *prefrontal cortex*) es una de las áreas más complejas del cerebro de los mamíferos. Su complejidad se debe en parte a que sus neuronas participan en una gran variedad de tareas. Muchas regiones del cerebro proyectan su actividad hacia la PFC, que funciona como un orquestador capaz de integrar información de distintas fuentes para guiar la interacción con el ambiente. Una de sus capacidades más estudiadas es el mantenimiento de información contextual a lo largo del tiempo, conocida como memoria de trabajo.

Debido a su acceso a información de múltiples fuentes, la PFC es frecuentemente asociada a capacidades como el pensamiento abstracto, el lenguaje y la planificación. Dada la diversidad de funciones que se le atribuyen, existen distintos modelos computacionales del PFC, cada uno enfocado en explicar aspectos diferentes de estas capacidades.

Uno de ellos es el trabajo de Wang et al. [30, 31], que propone una red recurrente como modelo del PFC. En una red recurrente, parte de la actividad de las neuronas en un momento dado se retroalimenta como entrada a sí mismas y a otras neuronas en el momento siguiente, lo que les permite mantener una memoria de corto plazo en su actividad (ver Figura 6). El modelo de Wang busca describir la capa-

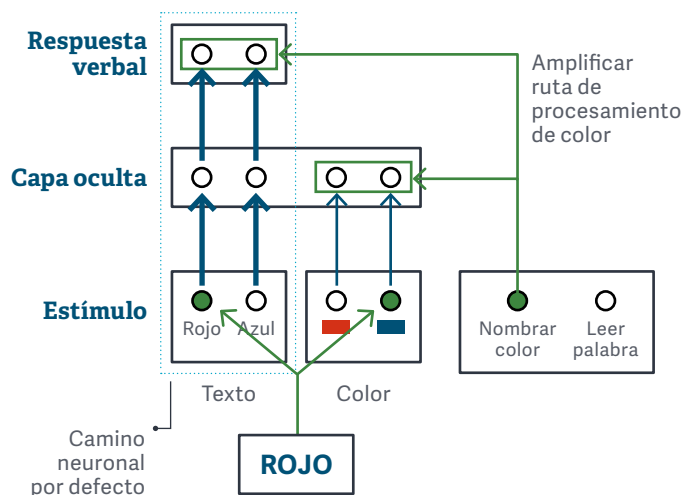
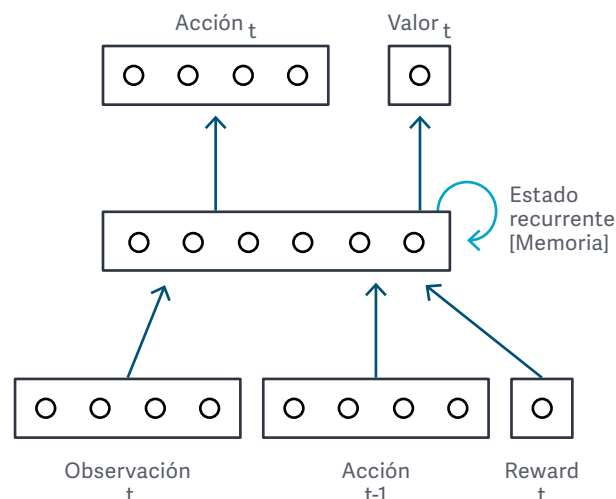


Figura 6 / Arriba: corteza prefrontal como meta-aprendizaje. El modelo propuesto por Wang et al. (2018) es una red recurrente entrenada con aprendizaje por refuerzo en múltiples tareas. La recurrencia permite que la actividad de las neuronas en un instante sirva como entrada en el siguiente, formando una memoria a corto plazo. Esta memoria le permite a la red acumular evidencia del ambiente, inferir en qué tarea se encuentra y aprender por prueba y error sin modificar sus pesos sinápticos. **Abajo: modelo de Stroop** (Cohen et al., 1990). La red modela la tarea Stroop, en la que se debe leer una palabra o mencionar el color de las letras en que está escrita (por ejemplo **ROJO**, donde la respuesta a leer el texto es “rojo”, y la respuesta a decir el color es “azul”). Las personas responden más rápido leyendo que nombrando el color, porque la práctica acumulada de lectura hace que esa ruta neuronal sea más fuerte (ruta por defecto). Sin embargo, una señal de contexto puede amplificar rutas alternativas (utilizando las conexiones en verde) dependiendo de la instrucción (nombrar el color o leer la palabra), permitiendo a la red resolver ambas variantes de la tarea.



Adaptado de Kaufman et al. 2014

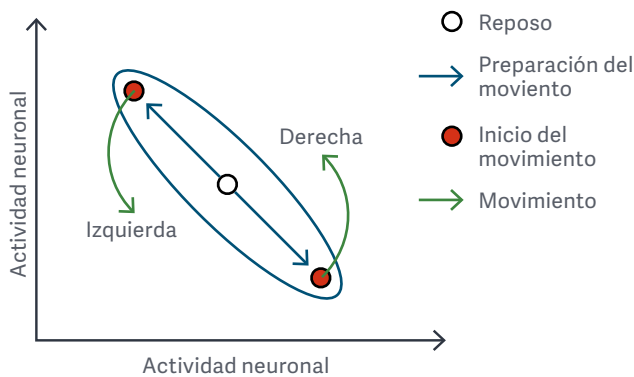


Figura 7 / Espacio de preparación motora en la corteza motora: antes de ejecutar un movimiento, la actividad neuronal evoluciona hacia una condición inicial específica dentro del subespacio nulo (zona azul), donde los cambios de actividad no producen movimiento muscular. Una vez configurada esta condición inicial (puntos rojos), el movimiento se desencadena al proyectarse la actividad hacia el subespacio de salida (fuera de la zona azul), donde sí genera movimiento. La trayectoria resultante (flechas verdes) está determinada por las condiciones iniciales establecidas durante la preparación. Este mecanismo se formaliza matemáticamente con redes recurrentes y teoría de control óptimo [39].

cidad del PFC de *aprender a aprender*, también llamada meta-aprendizaje [32, 33]. La idea central es que los pesos sinápticos de la red son modificados lentamente a través del entrenamiento con aprendizaje por refuerzo, un proceso análogo a la plasticidad sináptica que ocurre a lo largo de la vida. Sin embargo, una vez entrenada, la red no necesita modificar sus pesos para adaptarse a una tarea nueva: su dinámica recurrente, es decir, la forma en que la actividad de las neuronas evoluciona en el tiempo, implementa por sí sola un algoritmo de aprendizaje rápido. En otras palabras, la red es capaz de aprender tareas nuevas sin modificar sus pesos. La recurrencia cumple aquí el rol de memoria: le permite acumular evidencia del ambiente para inferir el contexto en el que se encuentra y resolver el problema apropiado.

Junto con mantener una representación del contexto, el PFC puede modular la actividad de otras regiones del cerebro [34]. Un modelo clásico de esto es el modelo de Stroop propuesto por Cohen et al. [35], donde señales de control amplifican o atenúan distintas rutas de procesamiento en una red. Este modelo fue diseñado para explicar la tarea Stroop, en la que una persona ve una palabra escrita en un color determinado y debe, dependiendo de la instrucción, decir la palabra escrita o el color de las letras. Por ejemplo, si aparece la palabra **ROJO** (escrita en azul), la respuesta co-

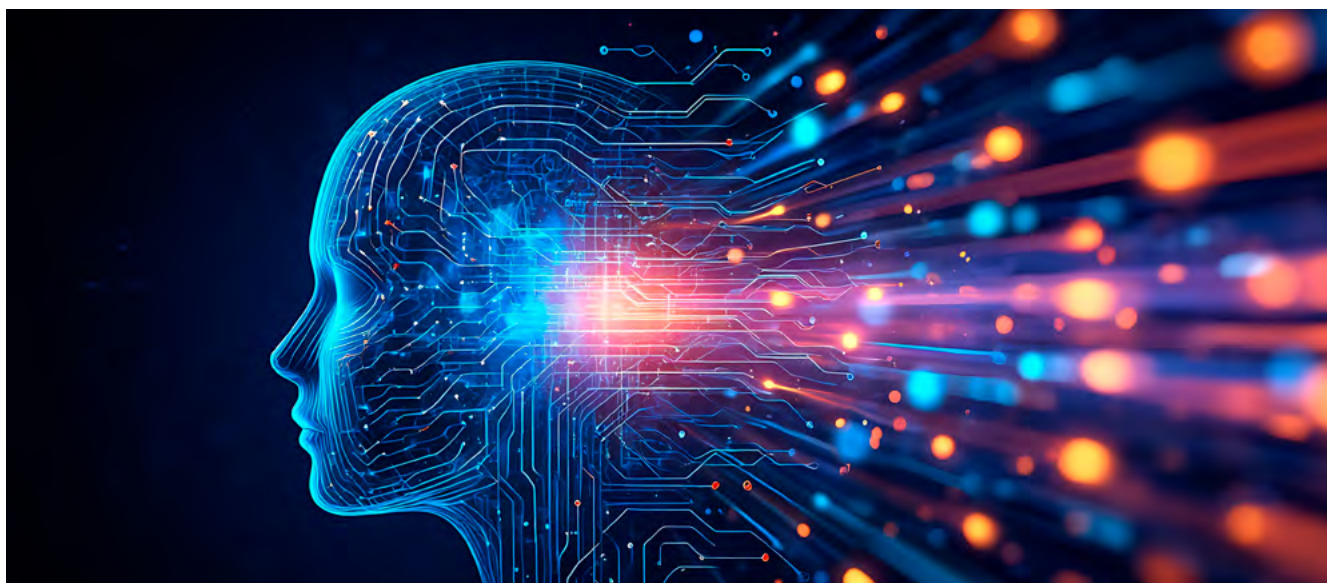
rrecta es 'azul' si la tarea es nombrar el color (ver Figura 6). Las personas pueden resolver esta tarea, pero tardan más cuando hay conflicto entre la palabra y el color. La hipótesis es que la ruta de lectura es más fuerte por práctica acumulada, y la señal de control debe amplificar activamente la ruta del color para superar esa ventaja [36].

Corteza motora

La corteza motora es la región principal responsable de la ejecución de movimientos voluntarios. No trabaja sola: está conectada con la corteza premotora y el área motora suplementaria, que planifican el movimiento antes de ejecutarlo, y con el cerebelo y los ganglios basales, que lo refinan y seleccionan [37]. Una observación clave es que la actividad neuronal en esta región comienza a evolucionar antes de que el movimiento ocurra, durante un período llamado preparación motora.

Una forma de entender este período de preparación es a través del modelo de subespacios de Churchland et al. [38, 39]. La idea central es que la corteza motora funciona como un sistema dinámico: preparar un movimiento consiste en llevar la actividad de la población neuronal a una condición inicial correcta en un espacio de alta dimensión, y el movimiento emerge como la trayectoria que el sistema recorre desde ahí. Un hallazgo elegante de este marco es la existencia de dos subespacios en la actividad neuronal: uno nulo, donde los cambios en actividad no producen movimiento muscular, y uno de salida, donde sí lo producen. Durante la preparación, la actividad evoluciona en el subespacio nulo, lo que permite al cerebro preparar el sistema dinámico antes del movimiento sin activar los músculos prematuramente. Cuando llega la señal de ejecutar, la actividad se proyecta al subespacio de salida y el movimiento ocurre. Este modelo se formaliza matemáticamente como una red recurrente cuya salida muscular es una proyección lineal de la actividad neuronal (ver Figura 7).

Un enfoque complementario es el del control óptimo [40], que trata la corteza motora no como un sistema dinámico sino como un controlador que resuelve un problema de optimización: ejecutar el movimiento deseado minimizando el esfuerzo muscular. Este marco predice bien propiedades observadas en movimientos reales, como la suavidad de las trayectorias y la forma en que la variabilidad se distribuye a lo largo del movimiento. A diferencia del modelo de subespacios, que describe la dinámica neuronal interna, el control óptimo describe el movimiento desde una perspectiva normativa: no cómo lo implementa el cerebro, sino qué problema está resolviendo. Ambos enfoques son complementarios y juntos ofrecen una descripción más completa de cómo la corteza motora planifica y ejecuta movimientos.



Acción y refuerzo

El cerebro puede entenderse, en gran medida, como un sistema de control: no sólo reacciona a estímulos, sino que genera predicciones sobre las consecuencias de sus acciones, selecciona comportamientos según metas y valores, y compara los resultados con lo que esperaba.

¿Cómo selecciona el cerebro qué acción tomar? Una forma de entender esto es a través del aprendizaje por refuerzo (RL del inglés Reinforcement Learning), un marco teórico en el que un agente aprende qué hacer a partir de señales de recompensa [41]. Imagina que estás aprendiendo a jugar ajedrez: algunos movimientos son mejores que otros a corto y largo plazo, y aprender eso requiere práctica. En principio, uno podría probar movimientos al azar y observar qué ocurre, asignando gradualmente un valor a cada jugada según sus consecuencias.

Uno de los hallazgos más importantes de la neurociencia es la conexión entre la dopamina y el error de predicción de recompensa. Si hago una jugada que creo será excelente y resulta ser un desastre, esperaba un valor alto y obtuve uno bajo: ese error de predicción está codificado por la liberación de dopamina. Niveles altos de dopamina señalan que el resultado fue mejor de lo esperado; niveles bajos, que fue peor [42, 43]. Esta señal es lo que permite al cerebro ajustar el valor de las acciones con la experiencia, y puede ser modelada matemáticamente utilizando la ecuación de Bellman de aprendizaje reforzado [44, 41].

La dopamina no sólo codifica errores de predicción: también responde a estímulos novedosos y sorprendentes, y está

Niveles altos de dopamina señalan que el resultado fue mejor de lo esperado [...] Esto explica su papel en la adicción: sustancias [...] que elevan artificialmente la dopamina generan una señal [...] que el cerebro interpreta como algo valioso de repetir.

involucrada en estados motivacionales y de atención, junto con otros neuromoduladores como la noradrenalina, la acetilcolina y la serotonina [45, 46, 47, 48]. Esto explica su papel en la adicción: sustancias o comportamientos que elevan artificialmente la dopamina generan una señal de “resultado mejor de lo esperado” que el cerebro interpreta como algo valioso de repetir, alterando gradualmente los mecanismos de regulación y el aprendizaje mismo [49].

Aprendizaje en el cerebro

Como hemos visto, las redes neuronales del cerebro son diversas y trabajan en conjunto para producir comportamiento. Esto plantea una pregunta natural: ¿todas las neuronas aprenden de la misma forma? ¿cómo debería ajustarse una conexión en la corteza visual para mejorar en una tarea como el tenis, que involucra también la corteza motora y la PFC?

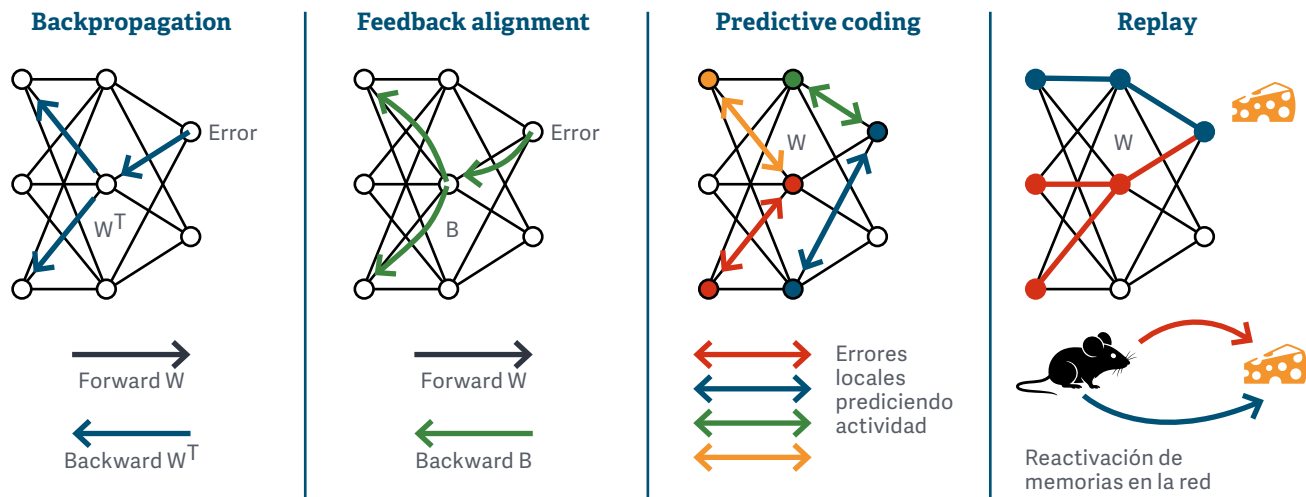


Figura 8 / Algoritmos de aprendizaje en redes neuronales artificiales y biológicas. **Backpropagation:** se calcula el error comparando la salida deseada con la entregada por la red (forward), y se retropropaga a través de los pesos sinápticos transpuestos para ajustar cada conexión (backward). Se piensa que el cerebro no puede implementar este cómputo debido a restricciones biológicas. **Feedback alignment:** similar al anterior, pero el error se retropropaga a través de pesos aleatorios fijos, distintos de los usados en el paso hacia adelante. El ajuste no es óptimo, pero el mecanismo es biológicamente plausible. **Predictive coding:** cada sinapsis se ajusta para predecir la actividad de la neurona siguiente en la red, sin necesidad de retropropagar el error explícitamente (proceso llamado configuración prospectiva). Puede alcanzar un rendimiento comparable al de backpropagation y es biológicamente plausible, aunque computacionalmente más costoso. **Replay:** no es una regla de ajuste sináptico sino un mecanismo de selección de contenido. Durante el descanso y el sueño, patrones de actividad asociados a experiencias previas se reactivan espontáneamente, consolidando el aprendizaje en las sinapsis.

El consenso actual es que el cerebro no puede implementar backpropagation, ya que requiere propagar información precisa a través de conexiones lejanas, algo biológicamente poco plausible.

Esta es una pregunta abierta, y el problema central se conoce como asignación de crédito (credit assignment) [50, 51]. Las reglas locales de plasticidad, como la hebbiana, sólo requieren información de neuronas vecinas, pero no explican bien el aprendizaje a gran escala ni funcionan bien en simulaciones.

Como mencionamos anteriormente, la *dopamina* provee una señal de retroalimentación que indica si el resultado de una acción fue mejor o peor de lo esperado. En términos más formales, esta señal corresponde al error de predicción de recompensa, y es precisamente este tipo de señal el que aparece como ingrediente central en muchas de las propuestas de reglas de plasticidad biológicamente plausibles. Sin embargo, como veremos a continuación, disponer de una señal de error global no resuelve por sí solo el pro-

blema de cómo ajustar cada una de las millones de conexiones involucradas en el aprendizaje (ver Figura 8).

Backpropagation

En inteligencia artificial, el problema de credit assignment está relativamente resuelto mediante la retropropagación del error (*backpropagation*): se calcula el error entre la respuesta de la red y la deseada (forward) y el error se propaga hacia atrás (backward) para ajustar todas las conexiones. Sin embargo, el consenso actual es que el cerebro no puede implementar *backpropagation*, ya que requiere propagar información precisa a través de conexiones lejanas, algo biológicamente poco plausible [52, 53].

Feedback alignment

Existen alternativas intermedias que intentan mantener cierta eficacia de aprendizaje global siendo al mismo tiempo biológicamente plausibles. Una de las más conocidas es el alineamiento de retroalimentación (*feedback alignment*). En *backpropagation* estándar, el error se propaga hacia atrás usando exactamente los mismos pesos que se usaron en el paso hacia adelante, lo que es biológicamente problemático porque requeriría que cada neurona “conoz-

ca” con precisión los pesos de conexiones lejanas. En *feedback alignment*, el error se propaga hacia atrás a través de un conjunto de sinapsis distinto, inicializado aleatoriamente y que no cambia durante el entrenamiento [54]. Sorprendentemente, la red aprende de todas formas, porque los pesos del paso hacia adelante se adaptan gradualmente para alinearse con estas sinapsis fijas. La ventaja biológica es clara: no se requiere transportar información precisa de la red hacia atrás, sino simplemente propagar alguna señal de error a través de conexiones independientes, algo más compatible con la anatomía neuronal conocida. El costo es que aprende más lento y con menor precisión que *backpropagation*, especialmente en tareas complejas.

Predictive coding

El *código predictivo* (*predictive coding*) parte de una idea diferente. En lugar de ajustar los pesos para minimizar directamente el error en la salida de la red, cada neurona ajusta sus conexiones para predecir la actividad de la neurona siguiente en la red. El error que se propaga no es el error global de la tarea, sino la diferencia entre lo que una neurona predijo y lo que la siguiente neurona realmente hizo [55]. A este proceso de ajuste simpático basado en código predictivo se le llama “Configuración Prospectiva”. Esto tiene dos ventajas importantes: primero, la regla de ajuste es local, ya que cada neurona sólo necesita comparar su predicción con la actividad de su vecina inmediata; segundo, a pesar de esta localidad, el ajuste global de la red emerge como consecuencia, sin necesidad de retropropagar el error explícitamente. Este tipo de algoritmo puede alcanzar un rendimiento comparable al de *backpropagation*, aunque requiere más cómputo y converge más lentamente. Su plausibilidad biológica ha generado bastante interés, ya que encaja con observaciones sobre cómo las neuronas corticales parecen codificar errores de predicción en lugar de representaciones directas del mundo. La plasticidad basada en código predictivo es parte de una idea más general [56], según la cual el cerebro está constantemente intentando predecir su próximo input sensorial y ajustando sus representaciones en función del error entre esa predicción y lo que realmente ocurre. Algunos autores van más lejos y postulan que este principio es suficiente para explicar toda la cognición y el comportamiento [57], una postura que genera debate considerable en el campo.

Replay

Más allá de la regla de plasticidad, el cerebro debe también seleccionar qué experiencias consolidar. Una hipótesis prominente es que esto ocurre mediante *replay*: durante

La memoria no se almacena en un único lugar del cerebro. Está distribuida en múltiples sistemas, implementados por distintos circuitos que operan en diferentes escalas de tiempo y cumplen funciones computacionales distintas.

el descanso, especialmente el sueño, el cerebro reactiva patrones de actividad asociados a experiencias previas, y esa reactivación, en conjunto con las reglas de plasticidad, consolida el conocimiento en las sinapsis. Por ejemplo, si en tu práctica de tenis ensayaste el saque muchas veces, durante el sueño se reactivarán neuronas asociadas a su ejecución en la corteza motora, consolidando ese aprendizaje en las sinapsis. Se piensa que el hipocampo es el principal coordinador de este proceso [58], aunque se han observado mecanismos similares en otras áreas de forma independiente [59]. Existen además teorías matemáticas basadas en aprendizaje por refuerzo que proponen criterios para explicar qué experiencias son priorizadas para replay y cuándo ocurre este proceso [60, 61].

Sistemas de memoria

Hasta ahora hemos visto cómo la experiencia modifica las sinapsis y cómo esos cambios alteran los circuitos neuronales. La memoria no se almacena en un único lugar del cerebro. Está distribuida en múltiples sistemas, implementados por distintos circuitos que operan en diferentes escalas de tiempo y cumplen funciones computacionales distintas.

El *hipocampo* permite formar asociaciones nuevas con rapidez, incluso después de una sola experiencia. Funciona como un índice temporal que vincula patrones distribuidos en la corteza. En cambio, el *neocórtex* aprende de manera gradual: sus representaciones se ajustan lentamente a partir de exposiciones repetidas.

Con el tiempo, la información inicialmente dependiente del hipocampo se integra directamente en redes corticales mediante un proceso llamado *consolidación sistémica* [62, 63]. Como vimos anteriormente, el mecanismo de replay reactiva patrones de actividad neuronal en la corteza durante el sueño y los períodos de reposo, consolidando representaciones de largo plazo. Por eso, el daño al hipocampo afec-

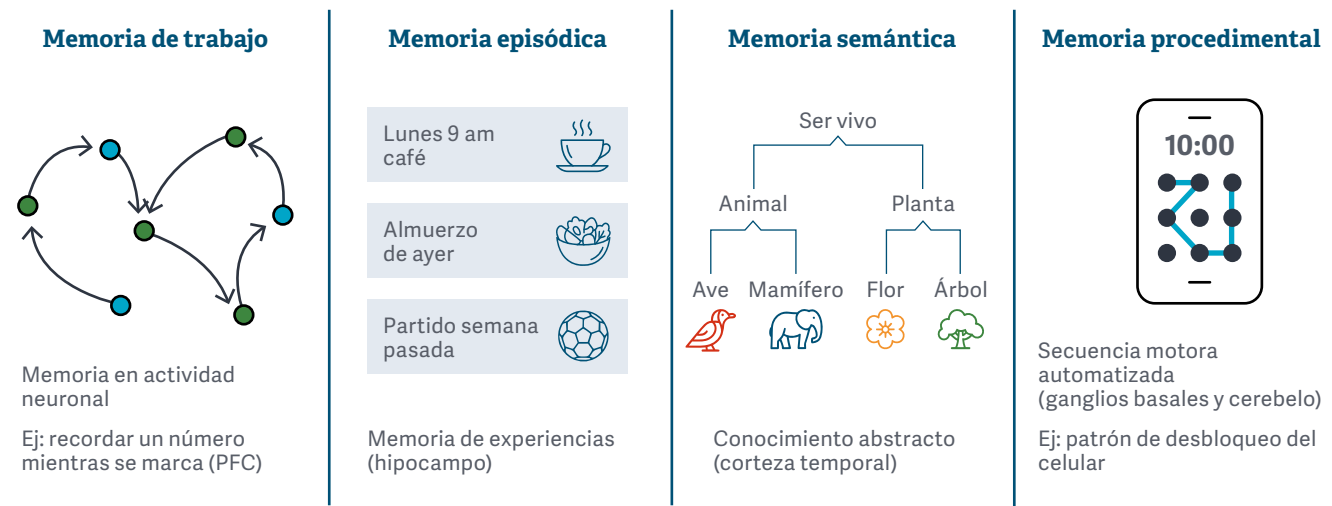


Figura 9 / Tipos de memoria y sus mecanismos. **Memoria de trabajo:** mantenida por actividad neuronal recurrente en la corteza prefrontal (PFC), sin cambios sinápticos duraderos. Ejemplo: recordar un número mientras se marca. **Memoria episódica:** registro de experiencias específicas que incluye qué ocurrió, cuándo y dónde, como el café del lunes o el partido de la semana pasada. Depende críticamente del hipocampo. **Memoria semántica:** conocimiento abstracto sobre el mundo organizado en categorías y relaciones jerárquicas, como la clasificación de seres vivos en animales y plantas. Se almacena en redes distribuidas de la corteza temporal. **Memoria procedimental:** habilidades motoras automatizadas, como el patrón de desbloqueo del teléfono o técnicas en algún deporte, que se ejecutan sin atención consciente a los pasos individuales. Involucra los ganglios basales y el cerebelo.

ta sobre todo la formación de recuerdos nuevos, mientras que muchos recuerdos antiguos permanecen intactos [64].

Sobre esta base se distinguen varios sistemas de memoria (ver Figura 9):

Memoria de trabajo

Permite mantener temporalmente información activa, por ejemplo al recordar un número telefónico mientras lo marcas. Se implementa principalmente mediante patrones sostenidos de actividad en circuitos prefrontales y parietales, más que por cambios sinápticos duraderos [65, 66]. Esta es la memoria mantenida por los modelos propuestos por Wang and Cohen sobre el PFC.

Memoria episódica

Corresponde a recuerdos de experiencias específicas, incluyendo qué ocurrió, dónde y cuándo. Depende críticamente del hipocampo, que vincula representaciones distribuidas en distintas áreas corticales para formar una memoria coherente [67, 68, 69].

Memoria semántica

Contiene conocimiento general sobre el mundo, como hechos, conceptos y relaciones. Se almacena en redes cor-

tales distribuidas, especialmente en áreas temporales y de asociación [70, 71, 72]. Muchas memorias semánticas se originan en experiencias episódicas, pero con el tiempo pierden su contexto y se transforman en conocimiento más abstracto. Se piensa que el mecanismo de replay consolida la memoria semántica a través de múltiples updates de patrones de activación [62, 63].

Memoria procedimental

Corresponde a habilidades y hábitos, como andar en bicicleta, escribir en un teclado o tocar un instrumento. Involucra principalmente a los *ganglios basales* y al *cerebelo*, que aprenden a seleccionar acciones y ajustar su ejecución mediante señales de refuerzo y corrección de error [73, 59].

Redes de Hopfield para almacenar memorias

Las redes de Hopfield [74], trabajo por el cual John Hopfield recibió el Premio Nobel de Física en 2024, son un modelo de memoria asociativa en el que las memorias se almacenan en los pesos sinápticos de una red recurrente. Su funcionamiento tiene dos etapas. Primero, el almacenamiento: cuando se quiere memorizar un patrón, los pesos entre neuronas que se activan juntas se fortalecen siguiendo una regla hebbiana. Este ajuste ocurre de una sola vez, sin entrenamiento iterativo, lo que lo hace atractivo como modelo de aprendizaje rápido a partir de una sola exposición. Segundo, la

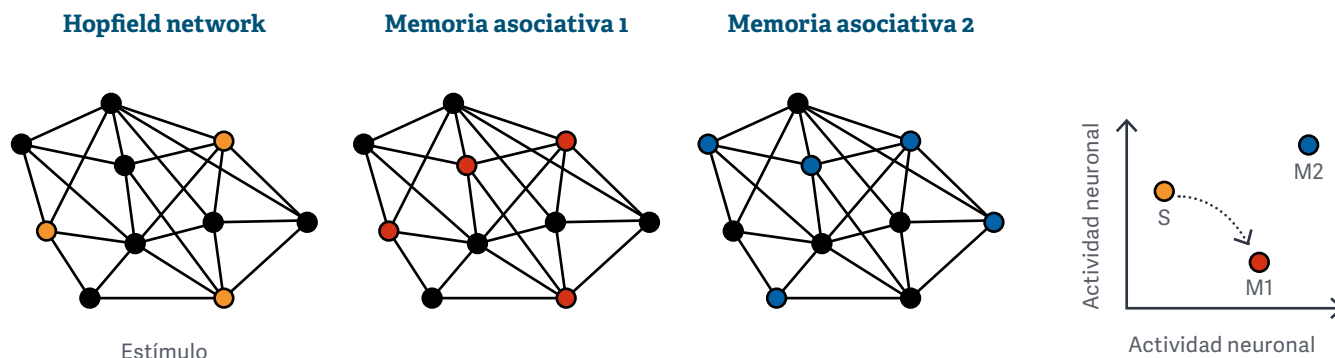


Figura 10 / Red de Hopfield y recuperación de memoria asociativa. **Izquierda:** una red de Hopfield recibe un estímulo parcial (neuronas naranjas), que activa un subconjunto de la red. La dinámica recurrente de la red evoluciona este patrón de actividad hacia la memoria almacenada más cercana: en este caso, la memoria asociativa 1 (neuronas rojas) o la memoria asociativa 2 (neuronas azules), dependiendo de cuál esté más próxima al estímulo inicial. **Derecha:** el mismo proceso representado en el espacio de actividad neuronal. El estímulo (S, naranja) corresponde a un punto en ese espacio, y la dinámica de la red lo desplaza hacia el atractor más cercano, en este caso la memoria M1 (rojo). Cada memoria almacenada corresponde a un mínimo de energía en este espacio, y la recuperación equivale a descender hacia ese mínimo.

recuperación: si se presenta a la red un estímulo parcial o ruidoso de un recuerdo almacenado, por ejemplo el rostro a medias de una persona conocida, la dinámica recurrente de la red actúa como un atractor que empuja la actividad neuronal hacia el patrón completo más cercano. En otras palabras, el sistema completa el recuerdo por sí solo (ver Figura 10).

Este modelo ha sido utilizado como descripción del área CA3 del hipocampo, que se cree funciona precisamente como una memoria auto asociativa capaz de recuperar experiencias completas a partir de pistas parciales [75, 76]. Una limitación clásica es que la capacidad de almacenamiento es modesta: más allá de cierto número de patrones, los recuerdos empiezan a interferir entre sí y la recuperación falla.

Una versión moderna de estas redes, propuesta por Ramsauer et al. [77], extiende el modelo a estados continuos y logra una capacidad de almacenamiento exponencialmente mayor. Lo notable es que esta versión moderna es matemáticamente equivalente al mecanismo de atención de los transformers, el componente central de modelos de lenguaje como GPT. Este resultado establece un puente inesperado entre un modelo de memoria biológica de los años ochenta y la arquitectura dominante en IA moderna.

Oportunidades en neurociencia computacional

Existen muchas preguntas abiertas en neurociencia que podrían beneficiarse del modelamiento matemático. Algunas las hemos discutido a lo largo del artículo: ¿cuál es la re-

gla de plasticidad del cerebro? ¿Qué nivel de abstracción es el adecuado para modelar neuronas y redes? ¿Cómo sabe el cerebro qué debe aprender y qué no? ¿Cómo la corteza prefrontal logra implementar algoritmos tan diversos? Aquí presentamos algunas preguntas adicionales que consideramos especialmente relevantes.

Planificación e imaginación

¿Cómo planifica el cerebro hacia el futuro o imagina situaciones que no han ocurrido? Los marcos teóricos más influyentes para este problema son los modelos de aprendizaje por refuerzo basados en modelos del mundo (*model-based RL* [78]) y los algoritmos de búsqueda en árboles de posibilidades, como los utilizados por AlphaGo [79, 80] para resolver el juego de Go, que venció al campeón mundial Lee Sedol por 4 a 1 en marzo de 2016, una década antes de lo que los expertos consideraban posible. Existen modelos de IA con esta capacidad, aunque los modelos de lenguaje más utilizados no la implementan de forma nativa; sin embargo existen intentos de incorporarse [81, 82].

Modularidad del cerebro

¿Por qué el cerebro está organizado en módulos especializados y no es simplemente una red grande de neuronas? Una respuesta posible proviene de la teoría de grafos: distintas funciones cognitivas podrían interferir entre sí si compartieran los mismos recursos, y la modularidad permitiría procesamiento en paralelo de distintos estímulos y tareas. Ver, escuchar, caminar y conversar simultáneamente requiere que distintos sistemas operen de forma relativa-



La inteligencia biológica y la artificial comparten principios fundamentales [...]. Quizás la inteligencia no es sólo un producto de la evolución de los seres vivos o un logro de la ingeniería, sino algo más fundamental en el universo.

mente independiente [36, 83]. Sin embargo, el origen y los límites de esta modularidad siguen siendo debatidos.

Conciencia

¿Qué es la conciencia y por qué tenemos experiencia subjetiva? Este es el llamado problema difícil de la conciencia, y sigue sin resolverse. Gran parte del debate todavía gira en torno a la definición de conciencia. Algunos proponen que la conciencia es una propiedad fundamental de cualquier sistema complejo, no exclusiva de los seres inteligentes [84]. Quizás la teoría más influyente es la del espacio de trabajo global: la conciencia emerge cuando cierta información se vuelve globalmente accesible dentro del cerebro, integrándose y quedando disponible simultáneamente para múltiples sistemas como la memoria, el razonamiento y la planificación. Bajo esta perspectiva, la mayor parte del procesamiento ocurre de forma inconsciente en subsistemas especializados, pero cuando la información entra en este espacio de trabajo compartido, se difunde ampliamente y permite una coordinación flexible entre sistemas. Esa disponibilidad global sería lo que asociamos con la experiencia consciente, y funcionalmente parece facilitar operaciones como el razonamiento abstracto, la planificación a largo plazo y el control deliberado de la conducta [85, 86].

¿Qué queremos decir con entender el cerebro?

Quizás una de las preguntas más fundamentales en neurociencia computacional es que significa "entender" cómo funciona el cerebro. Algunos incluso argumentan que el cerebro no puede entenderse a sí mismo en su totalidad, debido a que se intenta entender un instrumento (el cerebro) con el instrumento mismo, algo que suena contradictorio² [87]. Otros proponen que entender no requiere describir el cerebro funcionando con todas sus partes simultáneamen-

te, sino construir un atlas de las distintas funciones que implementa en distintas regiones, comprendiendo cada una por separado. Pero como vimos, incluso eso es difícil: cada región requiere modelos distintos, que son a su vez difíciles de comprender [88].

Para entender modelos de neurociencia computacional existen, a grandes rasgos, dos aproximaciones complementarias. La primera, quizás la más usada, consiste en proponer un modelo matemático simulable, compararlo con datos experimentales como actividad neuronal o comportamiento animal, e intervenir en su arquitectura, reglas de plasticidad u objetivo para estudiar cómo esas variaciones afectan los resultados [89, 90]. Este método es análogo a los experimentos de lesiones en el cerebro, donde se inactiva una región y se infiere su función observando los efectos en el comportamiento. Es un enfoque de arriba hacia abajo (*top-down*) que parte del sistema complejo y estudia las consecuencias de perturbarlo.

La segunda aproximación busca leyes o principios universales a través del análisis matemático, más parecido a lo que haría una física teórica [91]. Un gas está compuesto de millones de partículas con posiciones y velocidades específicas, pero la ley de los gases describe propiedades globales como la temperatura y la presión de forma exitosa sin entrar en el detalle de cada partícula. De forma análoga, muchos análisis matemáticos de redes neuronales describen propiedades globales de representación y aprendizaje sin modelar cada neurona individualmente, tomando prestadas herramientas de sistemas dinámicos, termodinámica y mecánica estadística [92, 93, 72]. La ventaja es que estas expresiones matemáticas se pueden entender de forma transparente bajo supuestos claros. La desventaja es que requieren tanta abstracción que a veces es difícil reconectar sus conclusiones con lo observado experimentalmente.

Es aquí donde existe una sinergia enorme con la inteligencia artificial: los modelos usados para describir partes del cerebro son muy parecidos a los modelos de aprendizaje de máquinas. Entender neurociencia computacional es, en gran medida, entender aprendizaje de máquinas, y viceversa. La idea que subyace a mucha de esta investigación es que la inteligencia biológica y la artificial comparten principios fundamentales, y que ambas son instancias de algo más profundo. Quizás la inteligencia no es sólo un producto de la evolución de los seres vivos o un logro de la ingeniería, sino algo más fundamental en el universo, donde el cerebro y la inteligencia artificial son dos instancias distintas de estudiarlo y entenderlo. **B**

2 <https://www.sainsburywellcome.org/web/blog/will-we-ever-understand-brain-12-leading-neuroscientists-reflect>
<https://www.thetransmitter.org/systems-neuroscience/the-limits-of-neuroscience/>

Referencias

- [1] Hodgkin, A. L., & Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117(4), 500–544. <https://doi.org/10.1113/jphysiol.1952.sp004764>.
- [2] Rall, W. (1962). Theory of Physiological Properties of Dendrites. *Annals of the New York Academy of Sciences*, 96(4), 1071–1092. <https://doi.org/10.1111/j.1749-6632.1962.tb54120.x>.
- [3] Tuckwell, H. C. (1988). *Introduction to Theoretical Neurobiology: Volume 1: Linear Cable Theory and Dendritic Structure* (Vol. 1). Cambridge University Press. <https://doi.org/10.1017/CBO9780511623271>.
- [4] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386–408. <https://doi.org/10.1037/h0042519>.
- [5] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- [6] Arcas, B. A. y, Fairhall, A. L., & Bialek, W. (2000). What Can a Single Neuron Compute? *Neural Information Processing Systems*. <https://api.semanticscholar.org/CorpusID:10445944>.
- [7] Beniaguev, D., Segev, I., & London, M. (2021). Single cortical neurons as deep artificial neural networks. *Neuron*, 109(17), 2727–2739. e3. <https://doi.org/10.1016/j.neuron.2021.07.002>.
- [8] Dittman, J. S. (2024). Taking a closer look at the synapse. *Proceedings of the National Academy of Sciences*, 121(33), e2412457121. <https://doi.org/10.1073/pnas.2412457121>.
- [9] Carvalho, T. P., & Buonomano, D. V. (2009). Differential Effects of Excitatory and Inhibitory Plasticity on Synaptically Driven Neuronal Input-Output Functions. *Neuron*, 61(5), 774–785. <https://doi.org/10.1016/j.neuron.2009.01.013>.
- [10] Kim, S., Wallace, M. L., El-Rifai, M., Knudsen, A. R., & Sabatini, B. L. (2022). Co-packaging of opposing neurotransmitters in individual synaptic vesicles in the central nervous system. *Neuron*, 110(8), 1371–1384.e7. <https://doi.org/10.1016/j.neuron.2022.01.007>.
- [11] Hebb, D. O. (1949). *The organization of behavior; a neuropsychological theory* (pp. xix, 335). Wiley.
- [12] Markram, H., Lübke, J., Frotscher, M., & Sakmann, B. (1997). Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science*, 275(5297), 213–215. <https://doi.org/10.1126/science.275.5297.213>.
- [13] Bi, G., & Poo, M. (1998). Synaptic Modifications in Cultured Hippocampal Neurons: Dependence on Spike Timing, Synaptic Strength, and Postsynaptic Cell Type. *Journal of Neuroscience*, 18(24), 10464–10472. <https://doi.org/10.1523/JNEUROSCI.18-24-10464.1998>.
- [14] Ross, M. T., Flores, D., Bertram, R., Johnson, F., & Hyson, R. L. (2017). Neuronal Intrinsic Physiology Changes During Development of a Learned Behavior. *eNeuro*, 4(5). <https://doi.org/10.1523/ENEURO.0297-17.2017>.
- [15] Disouky, A., Sanborn, M. A., Sabitha, K. R., Mostafa, M. M., Ayala, I. A., Bennett, D. A., Lu, Y., Zhou, Y., Keene, C. D., Weintraub, S., Gefen, T., Mesulam, M.-M., Geula, C., Maienschein-Cline, M., Rehman, J., & Lazarov, O. (2026). Human hippocampal neurogenesis in adulthood, ageing and Alzheimer's disease. *Nature*, 1–10. <https://doi.org/10.1038/s41586-026-10169-4>.
- [16] Lee, S., Liebana, S., Clopath, C., & Dabney, W. (2024). Lifelong Reinforcement Learning via Neuromodulation (arXiv:2408.08446). *arXiv*. <https://doi.org/10.48550/arXiv.2408.08446>.
- [17] Barton, A. J., Valdés, J. J., & Orchard, R. (2009). Neural networks with multiple general neuron models: A hybrid computational intelligence approach using Genetic Programming. *Neural Networks, Advances in Neural Networks Research: IJCNN2009*, 22(5), 614–622. <https://doi.org/10.1016/j.neunet.2009.06.043>.
- [18] Tavanaei, A., Ghodrati, M., Kheradpisheh, S. R., Masquelier, T., & Maida, A. S. (2019). Deep Learning in Spiking Neural Networks. *Neural Networks*, 111, 47–63. <https://doi.org/10.1016/j.neunet.2018.12.002>.
- [19] Murphy-Royal, C., Ching, S., & Papouin, T. (2023). A conceptual framework for astrocyte function. *Nature Neuroscience*, 26(11), 1848–1856. <https://doi.org/10.1038/s41593-023-01448-8>.
- [20] Cooper, M. L., Selles, M. C., Cammer, M., Redd, C., Gildea, H. K., Sall, J., Chiurri, K. E., Cheung, P., Wheeler, D. G., Saab, A. S., Liddelov, S. A., & Chao, M. V. (2026). Astrocytes connect specific brain regions through plastic networks. *Nature*, 1–9. <https://doi.org/10.1038/s41586-026-10426-6>.
- [21] Maldonado, K. A., & Alsayouri, K. (2026). *Physiology, Brain*. In StatPearls. StatPearls Publishing. <http://www.ncbi.nlm.nih.gov/books/NBK551718/>.
- [22] Taylor, H. P., Huynh, K. M., Thung, K.-H., Lin, G., Lyu, W., Lin, W., Ahmad, S., & Yap, P.-T. (2026). Functional hierarchy of the human neo-cortex across the lifespan. *Nature*, 1–10. <https://doi.org/10.1038/s41586-026-10219-x>.
- [23] Hawkins, J., Ahmad, S., & Cui, Y. (2017). A Theory of How Columns in the Neocortex Enable Learning the Structure of the World. *Frontiers in Neural Circuits*, 11. <https://doi.org/10.3389/fncir.2017.00081>.
- [24] Hubel, D. H., & Wiesel, T. N. (1959). Receptive fields of single neurones in the cat's striate cortex. *The Journal of Physiology*, 148(3), 574–591. <https://doi.org/10.1113/jphysiol.1959.sp006308>.



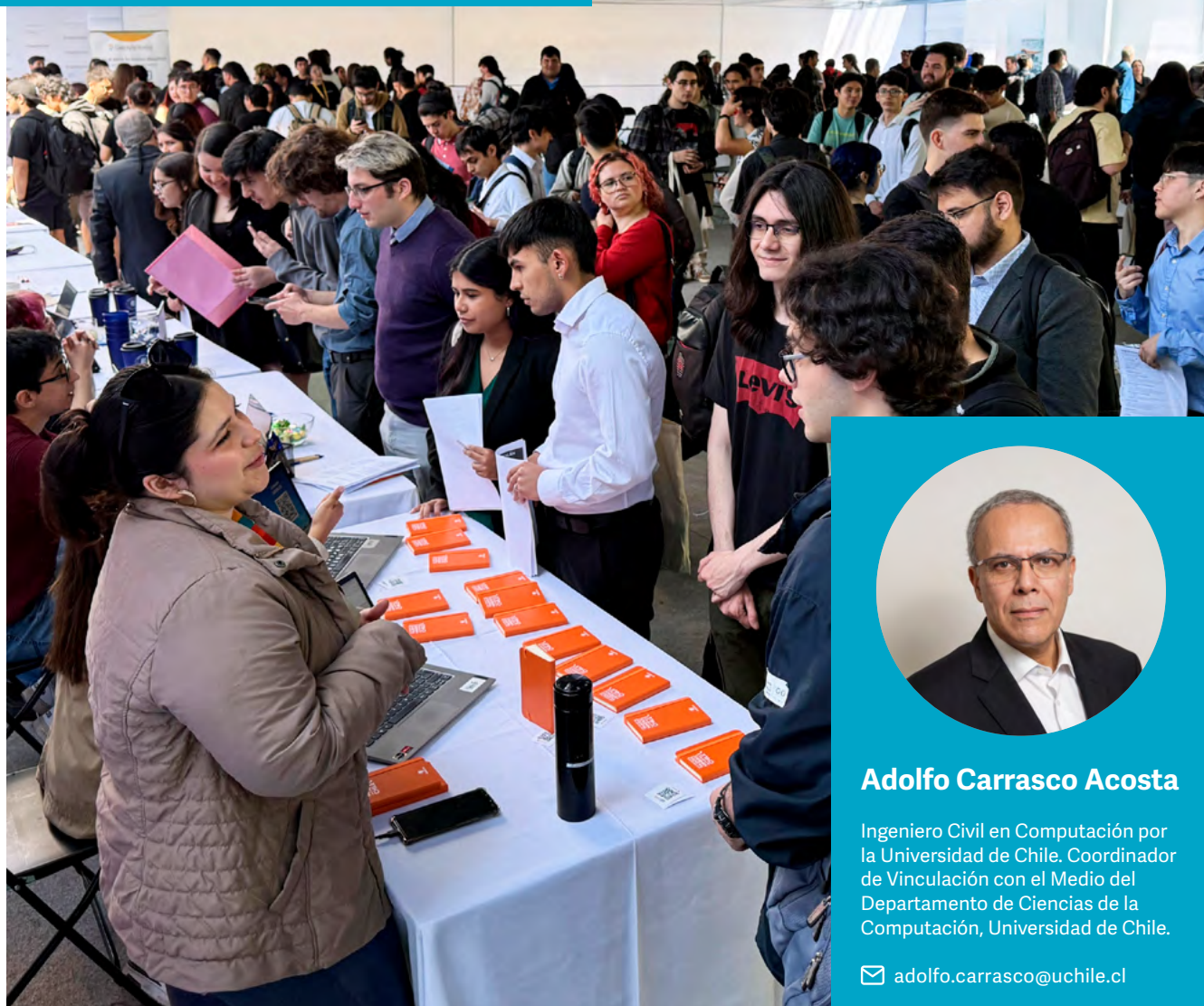
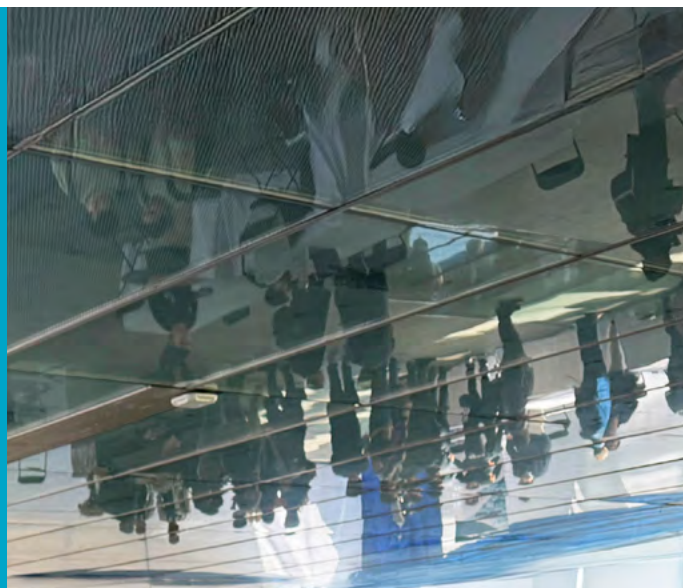
- [25] Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4), 193–202. <https://doi.org/10.1007/BF00344251>.
- [26] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4), 541–551. <https://doi.org/10.1162/neco.1989.1.4.541>.
- [27] Olshausen, B. A., & Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583), 607–609. <https://doi.org/10.1038/381607a0>.
- [28] Barlow, H. B. (2012). Possible Principles Underlying the Transformations of Sensory Messages. In W. A. Rosenblith (Ed.), *Sensory Communication* (p. 0). The MIT Press. <https://doi.org/10.7551/mitpress/9780262518420.003.0013>.
- [29] Manookin, M. B., & Rieke, F. (2023). Two Sides of the Same Coin: Efficient and Predictive Neural Coding. *Annual Review of Vision Science*, 9(Volume 9, 2023), 293–311. <https://doi.org/10.1146/annurev-vision-112122-020941>.
- [30] Wang, J. X., Kurth-Nelson, Z., Kumaran, D., Tirumala, D., Soyer, H., Leibo, J. Z., Hassabis, D., & Botvinick, M. (2018). Prefrontal cortex as a meta-reinforcement learning system. *Nature Neuroscience*, 21(6), 860–868. <https://doi.org/10.1038/s41593-018-0147-8>.
- [31] Wang, J. X. (2021). Meta-learning in natural and artificial intelligence. *Current Opinion in Behavioral Sciences, Computational Cognitive Neuroscience*, 38, 90–95. <https://doi.org/10.1016/j.cobeha.2021.01.002>.
- [32] Son, J., Lee, S., & Kim, G. (2024). When Meta-Learning Meets Online and Continual Learning: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–20. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. <https://doi.org/10.1109/TPAMI.2024.3463709>.
- [33] Vettoruzzo, A., Bouguelia, M.-R., Vanschoren, J., Rögnvaldsson, T., & Santosh, K. (2024). Advances and Challenges in Meta-Learning: A Technical Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(7), 4763–4779. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. <https://doi.org/10.1109/TPAMI.2024.3357847>.
- [34] Miller, E. K., & Cohen, J. D. (2001). An Integrative Theory of Prefrontal Cortex Function. *Annual Review of Neuroscience*, 24 (Volume 24, 2001), 167–202. <https://doi.org/10.1146/annurev.neuro.24.1.167>.
- [35] Cohen, J. D., Dunbar, K., & McClelland, J. L. (1990). On the control of automatic processes: A parallel distributed processing account of the Stroop effect. *Psychological Review*, 97(3), 332–361. <https://doi.org/10.1037/0033-295X.97.3.332>.
- [36] Musslick, S., Saxe, A., Hoskin, A., Sagiv, Y., Reichman, D., Petri, G., & Cohen, J. (2020). On the Rational Boundedness of Cognitive Control: Shared Versus Separated Representations. *OSF*. <https://doi.org/10.31234/osf.io/jkhdf>.
- [37] Rizzolatti, G., & Luppino, G. (2001). The Cortical Motor System. *Neuron*, 31(6), 889–901. [https://doi.org/10.1016/S0896-6273\(01\)00423-8](https://doi.org/10.1016/S0896-6273(01)00423-8).
- [38] Churchland, M. M., Cunningham, J. P., Kaufman, M. T., Foster, J. D., Nuyujukian, P., Ryu, S. I., & Shenoy, K. V. (2012). Neural population dynamics during reaching. *Nature*, 487(7405), 51–56. <https://doi.org/10.1038/nature11129>.
- [39] Kaufman, M. T., Churchland, M. M., Ryu, S. I., & Shenoy, K. V. (2014). Cortical activity in the null space: Permitting preparation without movement. *Nature Neuroscience*, 17(3), 440–448. <https://doi.org/10.1038/nn.3643>.
- [40] Todorov, E., & Jordan, M. I. (2002). Optimal feedback control as a theory of motor coordination. *Nature Neuroscience*, 5(11), 1226–1235. <https://doi.org/10.1038/nn963>.
- [41] Sutton, R., & Barto, A. G. (2018). *Reinforcement learning: An Introduction*. MIT Press.
- [42] Schultz, W., Dayan, P., & Montague, P. R. (1997). A Neural Substrate of Prediction and Reward. *Science*, 275(5306), 1593–1599. <https://doi.org/10.1126/science.275.5306.1593>.
- [43] Niv, Y., Duff, M. O., & Dayan, P. (2005). Dopamine, uncertainty and TD learning. *Behavioral and Brain Functions: BBF*, 1, 6. <https://doi.org/10.1186/1744-9081-1-6>.
- [44] Bellman, R. (2010). *Dynamic Programming*. Princeton University Press.
- [45] Bari, A., Xu, S., Pignatelli, M., Takeuchi, D., Feng, J., Li, Y., & Tonegawa, S. (2020). Differential attentional control mechanisms by two distinct noradrenergic coeruleo-frontal cortical pathways. *Proceedings of the National Academy of Sciences*, 117(46), 29080–29089. <https://doi.org/10.1073/pnas.2015635117>.
- [46] Cardozo Pinto, D. F., Pomrenze, M. B., Guo, M. Y., Touponse, G. C., Chen, A. P. F., Bentzley, B. S., Eshel, N., & Malenka, R. C. (2025). Opponent control of reinforcement by striatal dopamine and serotonin. *Nature*, 639(8053), 143–152. <https://doi.org/10.1038/s41586-024-08412-x>.
- [47] Chantranupong, L., Beron, C. C., Zimmer, J. A., Wen, M. J., Wang, W., & Sabatini, B. L. (2023). Dopamine and glutamate regulate striatal acetylcholine in decision-making. *Nature*, 621(7979), 577–585. <https://doi.org/10.1038/s41586-023-06492-9>.
- [48] Collins, A. G. E., & Frank, M. J. (2016). Surprise! Dopamine signals mix action, value and error. *Nature Neuroscience*, 19(1), 3–5. <https://doi.org/10.1038/nn.4207>.

- [49] Wise, R. A., & Jordan, C. J. (2021). Dopamine, behavior, and addiction. *Journal of Biomedical Science*, 28(1), 83. <https://doi.org/10.1186/s12929-021-00779-7>.
- [50] Richards, B. A., & Lillicrap, T. P. (2019). Dendritic solutions to the credit assignment problem. *Current Opinion in Neurobiology, Neurobiology of Learning and Plasticity*, 54, 28–36. <https://doi.org/10.1016/j.conb.2018.08.003>.
- [51] Ororbia, A. G. (2023). Brain-Inspired Machine Intelligence: A Survey of Neurobiologically-Plausible Credit Assignment (arXiv:2312.09257). arXiv. <https://doi.org/10.48550/arXiv.2312.09257>.
- [52] Lillicrap, T. P., Santoro, A., Marris, L., Akerman, C. J., & Hinton, G. (2020). Backpropagation and the brain. *Nature Reviews Neuroscience*, 21(6), 335–346. <https://doi.org/10.1038/s41583-020-0277-3>.
- [53] Etter, G. (2025). The brain as a blueprint: A survey of brain-inspired approaches to learning in artificial intelligence (arXiv:2511.04455). arXiv. <https://doi.org/10.48550/arXiv.2511.04455>.
- [54] Lillicrap, T. P., Cownden, D., Tweed, D. B., & Akerman, C. J. (2016). Random synaptic feedback weights support error backpropagation for deep learning. *Nature Communications*, 7(1), 13276. <https://doi.org/10.1038/ncomms13276>.
- [55] Song, Y., Millidge, B., Salvatori, T., Lukasiewicz, T., Xu, Z., & Bogacz, R. (2024). Inferring neural activity before plasticity as a foundation for learning beyond backpropagation. *Nature Neuroscience*, 27(2), 348–358. <https://doi.org/10.1038/s41593-023-01514-1>.
- [56] Rao, R. P. N., & Ballard, D. H. (1999). Predictive coding in the visual cortex: A functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2(1), 79–87. <https://doi.org/10.1038/4580>.
- [57] Friston, K., & Kiebel, S. (2009). Predictive coding under the free-energy principle. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1521), 1211–1221. <https://doi.org/10.1098/rstb.2008.0300>.
- [58] Ólafsdóttir, H. F., Bush, D., & Barry, C. (2018). The role of hippocampal replay in memory and planning. *Current Biology*, 28(1), R37–R50.
- [59] Thompson, E. J., Rollik, L., Waked, B., Mills, G., Kaur, J., Geva, B., Carrasco-Davis, R., George, T., Domine, C., Dorrell, W., & Stephenson-Jones, M. (2024). Replay of procedural experience is independent of the hippocampus (p. 2024.06.05.597547). *bioRxiv*. <https://doi.org/10.1101/2024.06.05.597547>.
- [60] Mattar, M. G., & Daw, N. D. (2018). Prioritized memory access explains planning and hippocampal replay. *Nature Neuroscience*, 21(11), 1609–1617.
- [61] Liu, Y., Mattar, M. G., Behrens, T. E. J., Daw, N. D., & Dolan, R. J. (2021). Experience replay is associated with efficient nonlocal learning. *Science*, 372(6544), eabf1357. <https://doi.org/10.1126/science.abf1357>.
- [62] McClelland, J., McNaughton, B., & O'Reilly, R. (1995). Why There are Complementary Learning Systems in the Hippocampus and Neocortex: Insights from the Successes and Failures of Connectionist Models of Learning and Memory. *Psychological Review*, 102, 419–457. <https://doi.org/10.1037/0033-295X.102.3.419>.
- [63] Sun, W., Advani, M., Spruston, N., Saxe, A., & Fitzgerald, J. E. (2023). Organizing memories for generalization in complementary learning systems. *Nature Neuroscience*, 26(8), 1438–1448. <https://doi.org/10.1038/s41593-023-01382-9>.
- [64] Squire, L. R. (2009). The Legacy of Patient H.M. for Neuroscience. *Neuron*, 61(1), 6–9. <https://doi.org/10.1016/j.neuron.2008.12.023>.
- [65] Reynolds, G. D., & Romano, A. C. (2016). The Development of Attention Systems and Working Memory in Infancy. *Frontiers in Systems Neuroscience*, 10. <https://doi.org/10.3389/fnsys.2016.00015>.
- [66] Oberauer, K., Lewandowsky, S., Awh, E., Brown, G. D. A., Conway, A., Cowan, N., Donkin, C., Farrell, S., Hitch, G. J., Hurlstone, M. J., Ma, W. J., Morey, C. C., Nee, D. E., Schweppe, J., Vergauwe, E., & Ward, G. (2018). Benchmarks for models of short-term and working memory. *Psychological Bulletin*, 144(9), 885–958. <https://doi.org/10.1037/bul0000153>.
- [67] Tulving, E. (2002). Episodic memory: From mind to brain. *Annual Review of Psychology*, 53, 1–25. <https://doi.org/10.1146/annurev.psych.53.100901.135114>.
- [68] Calderón, C. B., Loof, E. D., Ergo, K., Snoeck, A., Boehler, C. N., & Verguts, T. (2021). Signed Reward Prediction Errors in the Ventral Striatum Drive Episodic Memory. *Journal of Neuroscience*, 41(8), 1716–1726. <https://doi.org/10.1523/JNEUROSCI.1785-20.2020>.
- [69] Giallanza, T., Campbell, D., & Cohen, J. D. (2024). Toward the Emergence of Intelligent Control: Episodic Generalization and Optimization. *Open Mind*, 8, 688–722. https://doi.org/10.1162/opmi_a_00143.
- [70] Patterson, K., Nestor, P. J., & Rogers, T. T. (2007). Where do you know what you know? The representation of semantic knowledge in the human brain. *Nature Reviews Neuroscience*, 8(12), 976–987. <https://doi.org/10.1038/nrn2277>.
- [71] Chen, L., Lambon Ralph, M. A., & Rogers, T. T. (2017). A unified model of human semantic knowledge and its disorders. *Nature Human Behaviour*, 1(3), 0039. <https://doi.org/10.1038/s41562-016-0039>.
- [72] Saxe, A. M., McClelland, J. L., & Ganguli, S. (2019). A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116(23), 11537–11546. <https://doi.org/10.1073/pnas.1820226116>.



- [73] Foerde, K., & Shohamy, D. (2011). The role of the basal ganglia in learning and memory: Insight from Parkinson's disease. *Neurobiology of Learning and Memory, Memory Impairment and Disease*, 96(4), 624–636. <https://doi.org/10.1016/j.nlm.2011.08.006>.
- [74] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8), 2554–2558. <https://doi.org/10.1073/pnas.79.8.2554>.
- [75] Alonso, N., & Krichmar, J. L. (2024). A sparse quantized hopfield network for online-continual memory. *Nature Communications*, 15(1), 3722. <https://doi.org/10.1038/s41467-024-46976-4>.
- [76] Kang, L., & Toyozumi, T. (2024). Distinguishing examples while building concepts in hippocampal and artificial networks. *Nature Communications*, 15(1), 647. <https://doi.org/10.1038/s41467-024-44877-0>.
- [77] Ramsauer, H., Schäfl, B., Lehner, J., Seidl, P., Widrich, M., Adler, T., Gruber, L., Holzleitner, M., Pavlović, M., Sandve, G. K., Greiff, V., Kreil, D., Kopp, M., Klambauer, G., Brandstetter, J., & Hochreiter, S. (2021). Hopfield Networks is All You Need (arXiv:2008.02217). arXiv. <https://doi.org/10.48550/arXiv.2008.02217>.
- [78] Luo, F.-M., Xu, T., Lai, H., Chen, X.-H., Zhang, W., & Yu, Y. (2022). A Survey on Model-based Reinforcement Learning (arXiv:2206.09328). arXiv. <https://doi.org/10.48550/arXiv.2206.09328>.
- [79] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587), 484–489. <https://doi.org/10.1038/nature16961>.
- [80] McGrath, T., Kapishnikov, A., Tomašev, N., Pearce, A., Hassabis, D., Kim, B., Paquet, U., & Kramnik, V. (2022). Acquisition of Chess Knowledge in AlphaZero. *Proceedings of the National Academy of Sciences*, 119(47), e2206625119. <https://doi.org/10.1073/pnas.2206625119>.
- [81] Ding, J., Zhang, Y., Shang, Y., Feng, J., Zhang, Y., Zong, Z., Yuan, Y., Su, H., Li, N., Piao, J., Deng, Y., Sukiennik, N., Gao, C., Xu, F., & Li, Y. (2025). Understanding World or Predicting Future? A Comprehensive Survey of World Models (arXiv:2411.14499). arXiv. <https://doi.org/10.48550/arXiv.2411.14499>.
- [82] Hafner, D., Pasukonis, J., Ba, J., & Lillicrap, T. (2025). Mastering diverse control tasks through world models. *Nature*, 640(8059), 647–653. <https://doi.org/10.1038/s41586-025-08744-2>.
- [83] Béna, G., & Goodman, D. F. M. (2025). Dynamics of specialization in neural modules under resource constraints. *Nature Communications*, 16(1), 187. <https://doi.org/10.1038/s41467-024-55188-9>.
- [84] Chalmers, D. J. (2010). *The Character of Consciousness*. Oxford University Press. <https://doi.org/10.1093/acprof:oso/9780195311105.001.0001>.
- [85] Baars, B. J. (2005). Global workspace theory of consciousness: Toward a cognitive neuroscience of human experience. In S. Laureys (Ed.), *The Boundaries of Consciousness: Neurobiology and Neuropathology* (Vol. 150, pp. 45–53). Elsevier. [https://doi.org/https://doi.org/10.1016/S0079-6123\(05\)50004-9](https://doi.org/https://doi.org/10.1016/S0079-6123(05)50004-9).
- [86] Ferrante, O., Gorska-Klimowska, U., Henin, S., Hirschhorn, R., Khalaf, A., Lepauvre, A., Liu, L., Richter, D., Vidal, Y., Bonacchi, N., Brown, T., Sripad, P., Armendariz, M., Bendtz, K., Ghafari, T., Hetenyi, D., Jeschke, J., Kozma, C., Mazumder, D. R., ... Melloni, L. (2025). Adversarial testing of global neuronal workspace and integrated information theories of consciousness. *Nature*, 642(8066), 133–142. <https://doi.org/10.1038/s41586-025-08888-1>.
- [87] Jonas, E., & Kording, K. P. (2017). Could a Neuroscientist Understand a Microprocessor? *PLOS Computational Biology*, 13(1), e1005268. <https://doi.org/10.1371/journal.pcbi.1005268>.
- [88] Carandini, M. (2012). From circuits to behavior: A bridge too far? *Nature Neuroscience*, 15(4), 507–509. <https://doi.org/10.1038/nn.3043>.
- [89] Lillicrap, T. P., & Kording, K. P. (2019). What does it mean to understand a neural network? arXiv:1907.06374 [Cs, q-Bio, Stat]. <http://arxiv.org/abs/1907.06374>.
- [90] Richards, B. A., Lillicrap, T. P., Beaudoin, P., Bengio, Y., Bogacz, R., Christensen, A., Clopath, C., Costa, R. P., Berker, A. de, Ganguli, S., Gillon, C. J., Hafner, D., Kepecs, A., Kriegeskorte, N., Latham, P., Lindsay, G. W., Miller, K. D., Naud, R., Pack, C. C., ... Kording, K. P. (2019). A deep learning framework for neuroscience. *Nature Neuroscience*, 22(11), 1761–1770. <https://doi.org/10.1038/s41593-019-0520-2>.
- [91] Saxe, A., Nelli, S., & Summerfield, C. (2021). If deep learning is the answer, what is the question? *Nature Reviews Neuroscience*, 22(1), 55–67. <https://doi.org/10.1038/s41583-020-00395-8>.
- [92] Goldt, S., Advani, M., Saxe, A. M., Krzakala, F., & Zdeborová, L. (2019). Dynamics of stochastic gradient descent for two-layer neural networks in the teacher-student setup. *Advances in Neural Information Processing Systems*, 32. https://proceedings.neurips.cc/paper_files/paper/2019/hash/cab070d53bd0d200746fb852a922064a-Abstract.html.
- [93] Mei, S., Misiakiewicz, T., & Montanari, A. (2019). Mean-field theory of two-layers neural networks: Dimension-free bounds and kernel limit (arXiv:1902.06015). arXiv. <https://doi.org/10.48550/arXiv.1902.06015>.

Vinculación con el Medio en el DCC



Adolfo Carrasco Acosta

Ingeniero Civil en Computación por la Universidad de Chile. Coordinador de Vinculación con el Medio del Departamento de Ciencias de la Computación, Universidad de Chile.

✉ adolfo.carrasco@uchile.cl



Resumen / A partir de los tres pilares que rigen a la Universidad de Chile, el DCC ha formado un área de Vinculación con el Medio, la cual promueve actividades que unen a empresas y organizaciones externas con los más de 800 estudiantes actuales del Departamento. Esto se logra con iniciativas tales como la Feria Laboral 2026 que se realizará en septiembre y la formación de la red DCC Alumni.

¿Por qué es necesaria la vinculación en el DCC?

Desde 1842, cuando la Universidad de Chile se fundó, siempre ha estado en el centro de la actividad chilena y personajes tan relevantes como Andrés Bello o Valentín Letelier, han colaborado no sólo en lo académico, sino también en lo social. De esta forma, actualmente la Universidad despliega su quehacer en los ámbitos docente, investigación y vinculación, tomando esto último un rol creciente puesto que es una actividad bidireccional donde no sólo la Universidad entrega saberes, sino también aprende de la experiencia de su entorno.

En el DCC estamos trabajando en esa misma línea y, de hecho, parte de la formación de pregrado en Ingeniería Civil en Computación, requiere que los alumnos/as tengan un contacto real con el mundo “fuera de Beauchef” en sus prácticas profesionales, en el curso Proyecto de Software y en su memoria o proyecto de título. Todas estas interacciones requieren no sólo de las ganas de los alumnos/as, sino también de las oportunidades que las empresas, instituciones y organizaciones les puedan brindar, tanto como espacios de práctica profesional, aportando con proyectos y/o siendo lugares donde se obtenga el primer trabajo.

Nuestra realidad y objetivos

Teniendo presente que el DCC alberga ya a más de 800 estudiantes de pregrado, a partir del año 2025 se estableció una oficina que apoye estas actividades y sea el nexo entre el DCC y su entorno. La Coordinación de Vinculación con el Medio tiene la misión de establecer puentes con el entorno y apoyar al DCC y, en particular, a las coordinaciones de Docencia, Prácticas Profesionales, Prácticas de Título y al ramo Proyecto de Software, facilitando que más alumnos/as tengan acceso a mejores lugares y oportunidades para lograr un crecimiento en su ruta hacia convertirse en ingenieros e ingenieras.

Cuando nos vinculamos, todos ganamos

Pero esta vinculación no es sólo para que los y las estudiantes tengan lugares de práctica. Las empresas y orga-

En la “Feria Laboral 50 Años DCC” [...] empresas como Codelco o Banco de Chile reclutaron a más de 100 postulantes cada una.

nizaciones ganan muchísimo con estos procesos de vinculación, porque se les da la posibilidad de tener acceso a conocimientos de frontera, miradas frescas, innovación y, no menos relevante, la cultura que los alumnos/as del DCC llevan impregnada en sus actividades de cada día.

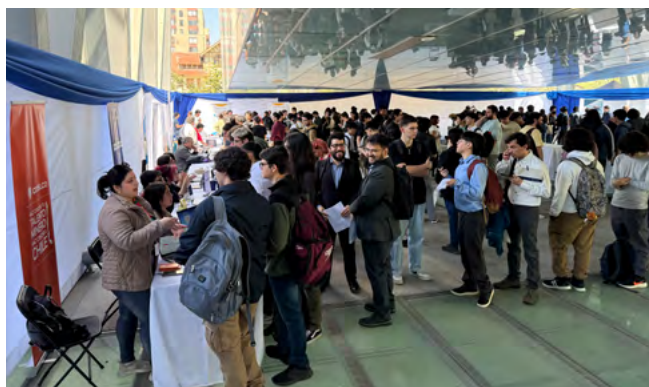
El valor que como DCC estamos entregando al entorno es, por lejos, una muy buena inversión para aquellas organizaciones que están trabajando con nosotros.

2025, el año de la formalización

Durante 2025 la coordinación de Vinculación con el Medio tuvo una importante actividad, que fue el organizar de manera centralizada y con apoyo de estudiantes, docentes y funcionarios/as la Feria Laboral que, históricamente, era una actividad que el Centro de Alumnos del DCC (CaDCC) armaba con recursos limitados e invirtiendo muchas horas-persona propias.

De esta manera, con el marco del aniversario número cincuenta del DCC, se creó la “Feria Laboral 50 Años DCC”, donde toda la comunidad DCCiana aportó con ideas y trabajo, logrando que se tuviera durante todo un día la presencia de 18 empresas de todos los sectores productivos, interactuando con más de 300 estudiantes con el objetivo de hacer *match* entre empresas y estudiantes que requirieran realizar su Práctica Profesional 1.

El resultado se tuvo a la vista, con empresas como Codelco o Banco de Chile, que reclutaron a más de 100 postulantes cada una, o empresas de nicho que buscaban capacidades técnicas y humanas muy específicas en sus futuros practicantes. Todas ellas lograron sus objetivos de encontrar a las mejores personas y el DCC pudo apoyar a que sus estudiantes inte-



Feria Laboral 50 Años DCC, realizada en octubre de 2025.



Lanzamiento de la red Alumni DCC, iniciativa orientada a fortalecer los vínculos entre el Departamento y sus egresado/as.

ractuaran comunicacionalmente con ejecutivos de estas organizaciones, ganándose un espacio de diferenciación que, en definitiva, hizo que obtuvieran sus prácticas profesionales.

Estamos proyectando un crecimiento sustentable

Durante este año 2026, los objetivos de la Coordinación de Vinculación con el Medio del DCC obviamente crecen y son el realizar la Feria Laboral DCC 2026 con 30 empresas, esta vez brindando oportunidades para que los alumnos/as puedan encontrar lugares para Práctica Profesional 1, Práctica Profesional 2, Proyecto de Título e, incluso, lugares para su primer trabajo una vez que hayan egresado.

A la vez, se está trabajando en levantar la red de exalumnos/as del DCC (DCC Alumni), que marca un punto de apego de los antiguos/as DCCianos/as, para que puedan convivir, recordar, robustecer su amistad con personas con las que compartieron en el pasado y, aún más importante, ser una luz para los actuales estudiantes, mostrándoles cómo han combina-

Estamos realizando encuentros de alumnos/as y exalumnos/as [...] donde apoyamos a que las nuevas ideas se unan con la experiencia.

do los conocimientos que la Universidad les enseñó, con las técnicas que tuvieron que aprender en el mercado, para desarrollar así su propia experiencia y camino de vida.

Durante este 2026 ya hemos iniciado este camino de unión entre actuales y antiguos/as DCCianos, con las "Crónicas DCCianas", que son conservatorios con exalumnos/as quienes entregan su experiencia y visión a las actuales generaciones, charlas técnicas donde empresas y organizaciones muestran cómo están consumiendo y aumentando el conocimiento tecnológico y, en lo social, estamos realizando encuentros de alumnos/as y exalumnos/as como una forma de tener un lugar común de conversación y esparcimiento, donde apoyamos a que las nuevas ideas se unan con la experiencia, para que todos/as logren crecer un poco más en su visión del mundo y su misión como ingenieros/as.

La conclusión es que somos un nexo de apoyo para el DCC

Como conclusión, podemos decir que estamos en un camino cambiante y dinámico, donde la tecnología forma parte ya no relevante sino crucial de nuestras vidas y que toda la comunidad DCCiana, estudiantes, docentes, funcionarios y alumni, tenemos como tarea el cumplir con los tres pilares de la Universidad de Chile, que son la docencia, la investigación y la vinculación. Para ello, desde la Coordinación de Vinculación con el Medio en el DCC estamos trabajando para que estos espacios sean mayores y accesibles para todos nosotros/as. **B**

¿Eres una empresa interesada en vincularte con el DCC? ↓

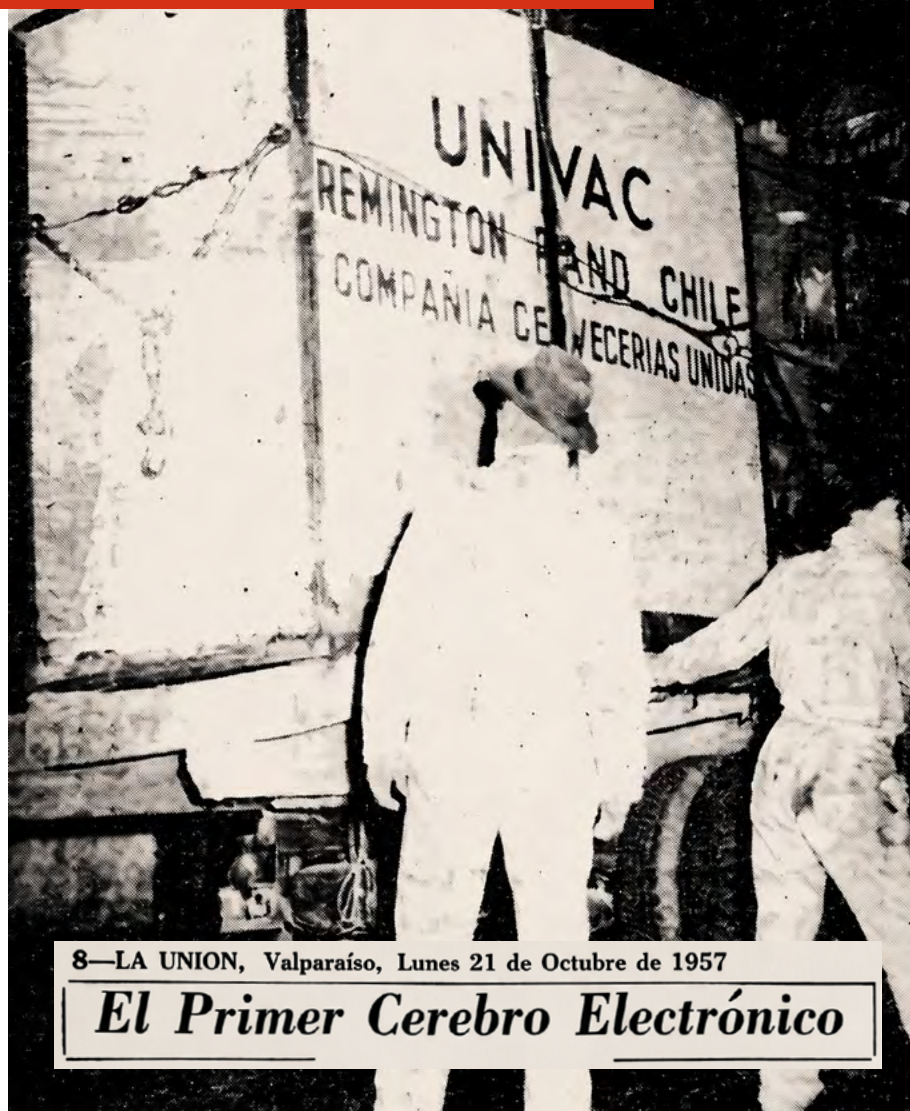


¿Eres un exalumno/a interesado en la red DCC Alumni? ↓



“El Primer Cerebro Electrónico”,

Valparaíso, Chile, 1957



Juan Álvarez Rubio

Master of Mathematics (Computer Science) por la Universidad de Waterloo, Canadá. Ingeniero de Ejecución en Procesamiento de la Información por la Universidad de Chile. Académico del Departamento de Ciencias de la Computación de la Universidad de Chile. Junto a su labor como docente, trabaja en reconstruir la historia de la computación en Chile.

✉ jalvarez@dcc.uchile.cl

Resumen / El primer computador digital llegó a Chile en 1957. Calificado por la prensa como un “cerebro electrónico”, se trató de un computador UNIVAC (UNiversal Automatic Computer) que fue adquirido por la Compañía de Cervecerías Unidas y se instaló en la ciudad de Valparaíso. El UNIVAC-60 fue un computador fabricado para leer, procesar y perforar información en tarjetas (“punched-card electronic computer”) y orientado principalmente a aplicaciones del área de contabilidad.

Introducción

Durante muchos años se afirmó que el primer computador digital había llegado a Chile a comienzos de la década del sesenta. De hecho, los catastros de la época no registraron ningún computador anterior. El inusitado hallazgo en un documento permitió rastrear, investigar y comprobar que el “primer cerebro electrónico”, como lo llamó la prensa de la época, llegó al país en 1957 y pasó casi desapercibido debido al contexto mundial de la “guerra fría” y la “carrera espacial” y por los acontecimientos que conmovieron al país. Este artículo describe el contexto, la adquisición, la llegada, la instalación, las características y el uso del computador UNIVAC-60 de la Compañía de Cervecerías Unidas en la ciudad de Valparaíso.

El contexto mundial y nacional

En 1957 se vivían los años de la “guerra fría”, particularmente reflejada en la “carrera espacial”. De hecho, en octubre de 1957, en Chile y en todo el mundo, se observaron las órbitas del satélite soviético Sputnik-1 (ver Figura 1), y, en noviembre, las del Sputnik-2 que fue tripulado por la perrita Laika. Meses después fueron lanzados los primeros satélites estadounidenses: Explorer-1 y Vanguard-1 [1]. En ambos casos, las noticias reportaban la utilización de “cerebros electrónicos” [2, 3].

En 1957 Chile tenía 7,5 millones de habitantes y el presidente de la República de Chile era, por segunda vez, Carlos Ibáñez del Campo. En el mes de enero, el país fue impactado por la muerte de Gabriela Mistral, que había obtenido el Premio Nobel de Literatura en 1945. En el mes de abril se produjo un violento estallido social, gatillado por el alza de los pasajes en el transporte público, que produjo muchas víctimas. Adicionalmente, el país fue afectado por una epidemia de influenza. En el área política, y después de las elecciones parlamentarias de marzo, comenzó la campaña para las elecciones presidenciales de 1958 con Alessandri, Frei y Allende como los principales candidatos.

En el área del procesamiento de datos, en los años veinte las instituciones estatales de ferrocarriles, aduanas y estadísticas habían sido las primeras en utilizar máquinas Hollerith para procesar información registrada en tarje-



Figura 1 / Póster conmemorativo Sputnik-1.

tas perforadas. Particularmente conocido fue su uso para producir los resultados del censo de 1930. Posteriormente, otros organismos fiscales y empresas privadas continuaron incorporando estas máquinas que, hasta el año 1957, fueron ocasionalmente también llamadas “cerebros electrónicos” [4]. Anecdóticamente, la prensa calificó también de “cerebro mecánico” a un computador analógico [5].

La adquisición del primer computador

La Compañía de Cervecerías Unidas (CCU) se fundó en 1902 por la fusión de dos empresas cerveceras: una en Valparaíso y Limache, y la otra en Santiago (Providencia) [6]. Desde sus inicios adoptó los más modernos métodos y técnicas de

producción y a fines de los años cincuenta contaba con más de 7 mil accionistas, 524 empleados y 2.271 obreros. Considerando los volúmenes de información y el crecimiento sostenido, la CCU utilizó máquinas para tarjetas perforadas hasta que decidió adquirir un computador digital.

La CCU compró el computador UNIVAC a la Fábrica Sperry Rand Co., representada en Chile por Remington Rand, en 60 mil dólares [7] con un año de garantía y con servicio de mantención con un pago anual del 5% del precio. Como parte de la venta, la corporación Remington Rand incluyó la capacitación de una persona durante 6 meses. El Sr. Roberto Bosque (ver Figura 2), uno de los técnicos en contabilidad mecanizada de la CCU, realizó un curso de 4 meses más una práctica de 2 meses en el Instituto de Electrónica de la Remington Rand [8]. Fue el único alumno latinoamericano en un curso de 50 empleados de las grandes industrias estadounidenses.

A su regreso, en el vapor Maipo en abril de 1957 [9], Roberto Bosque declaró que el computador era el tercero en Sudamérica, junto con otros dos en firmas petroleras en Venezuela. Destacó “la capacidad de trabajo y el tiempo récord en que lo realiza con un 100% de seguridad” y que “la velocidad se mide en microsegundos”. Por otra parte, señaló que “el personal que se ahorrará en los cálculos se podrá emplear en otras secciones” evitando los despidos y la contratación de nuevos empleados [10].

La llegada e instalación del computador

El computador UNIVAC de la CCU llegó en agosto de 1957 embalado en 14 cajas en una nave de la Sudamericana de Vapores. En primer lugar, tuvo que sortear problemas arancelarios. En efecto, no había ninguna partida específica en el Arancel Aduanero y superaba largamente el peso para ser considerada una máquina de calcular. Por lo tanto, la



Figura 2 / Roberto Bosque, abril 1957.

Junta General de Aduanas tuvo que crear una partida especial para esta máquina [8].

El segundo obstáculo fue la instalación en las oficinas de la CCU ubicadas en el sexto piso en la calle Prat 725 en el centro de Valparaíso. El tamaño y el peso de 1.750 kilos obligó a un estudio previo de la resistencia del piso, romper y ensanchar un ventanal y contratar una grúa. El izamiento fue realizado entre las 2 y las 6 de la madrugada del domingo 20 de octubre de 1957, obligando a la suspensión del tránsito [11]. La delicada y espectacular maniobra para levantar las dos cajas más voluminosas, que requirió la acrobática labor de obreros chilenos, fue informada en los diarios de Valparaíso, pero pasó casi inadvertida en la prensa santiaguina, y en ninguna de las noticias se utilizó la palabra computador (ver Tabla 1).

Día	Diario	Ciudad	Titular
19	El Mercurio	Santiago	Una máquina de contabilidad obliga a suspender tránsito
20	El Ilustrado	Santiago	Máquina contable hizo suspender el tránsito ayer
20	El Mercurio	Valparaíso	Un cerebro electrónico de 1.750 kilos será izado hoy al 6° piso de un edificio
21	El Mercurio	Valparaíso	Difícil maniobra para poder instalar cerebro electrónico
21	La Unión	Valparaíso	El Primer Cerebro Electrónico
22	La Nación	Santiago	Cerebro electrónico llegó a Valparaíso

Tabla 1 / Titulares de diarios de octubre de 1957.

El computador UNIVAC-60 representa el “eslabón perdido” en la cadena histórica de la computación en el país.

La fotografía de la instalación de las unidades principales fue el único registro visual del computador que iniciaría una nueva etapa computacional en el país (ver Figura 3). El equipamiento se instaló en una sala de 9 metros de largo por 3 de ancho [10]. El Jefe del Departamento Electrónico de la Remington Rand, Martín Álvarez, señaló que el computador UNIVAC “puede realizar las 4 operaciones aritméticas en milésimas de segundos y 79.000 sumas en un minuto y posee una memoria que le permite almacenar datos para acelerar todo tipo de operación de cálculo y reproducción” [8].

El computador

Los computadores UNIVAC (UNiversal Automatic Computer) corresponden a una línea de computadores digitales que comenzaron como productos de la Eckert-Mauchy Corporation. John Presper Eckert y John William Mauchly habían diseñado y construido el ENIAC (Electronic Numerical Integrator and Calculator) en la Universidad de Pennsylvania en 1945. En 1949, UNIVAC se convirtió en una división de la empresa Remington Rand, y en 1955 se fusionó con la empresa Sperry-Gyroscope dando origen a la Sperry Rand Corporation [12].

En particular, el UNIVAC modelo 60, fue lanzado en 1951 y la Remington Rand lo consideraba un “Computador Electrónico para Tarjetas Perforadas” (“punched card electronic computer”) comercializado especialmente como un máquina para el área de contabilidad. Por tratarse de un computador de primera generación utilizó la tecnología de tubos [13].

El UNIVAC-60 estaba compuesto de dos unidades principales: la unidad de “Lectura y Perforación de Tarjetas” (“Card Sensing Punching Unit”) y la unidad de “Computación Electrónica” (“Electronic Computing Unit”) (ver Figura 4). Las tarjetas que se utilizaban medían 7.335 por 3.25 pulgadas con 45 columnas y 12 filas (540 posibilidades de perforaciones circulares) y costaban 1/8 de centavo de dólar. Considerando las perforaciones para las letras y los dígitos, en cada columna se podían registrar 2 caracteres (ver Figura 5) a diferencia de las tarjetas IBM de 80 columnas (una por caracter) con perforaciones rectangulares [14]. Las perforaciones de los dígitos entre 0 y 9 reflejaban la forma en que se representaban internamente en 5 “quinary bits”: 00000, 10000, 10001, 01000, 01001, 00100, 00101, 00010, 00011, 00001 (los impares con un bit en 1 y los pares con 2 bits en 1).



Figura 3 / Instalación del UNIVAC en Valparaíso, octubre 1957.

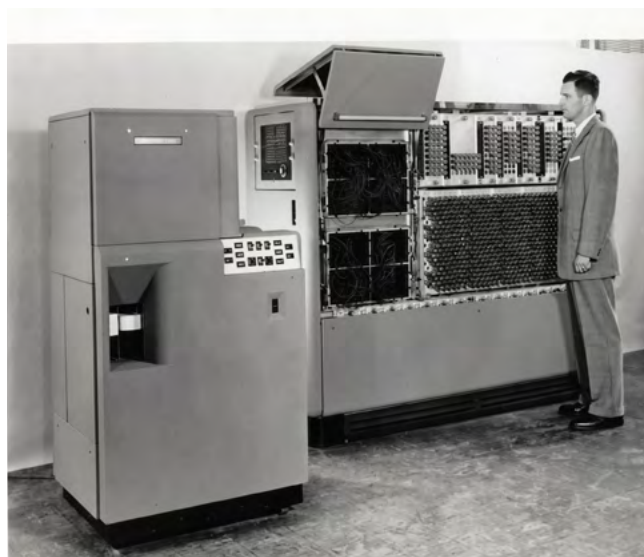


Figura 4 / “Unidad de Tarjetas Perforadas” y “Unidad de Computación” del UNIVAC-60.

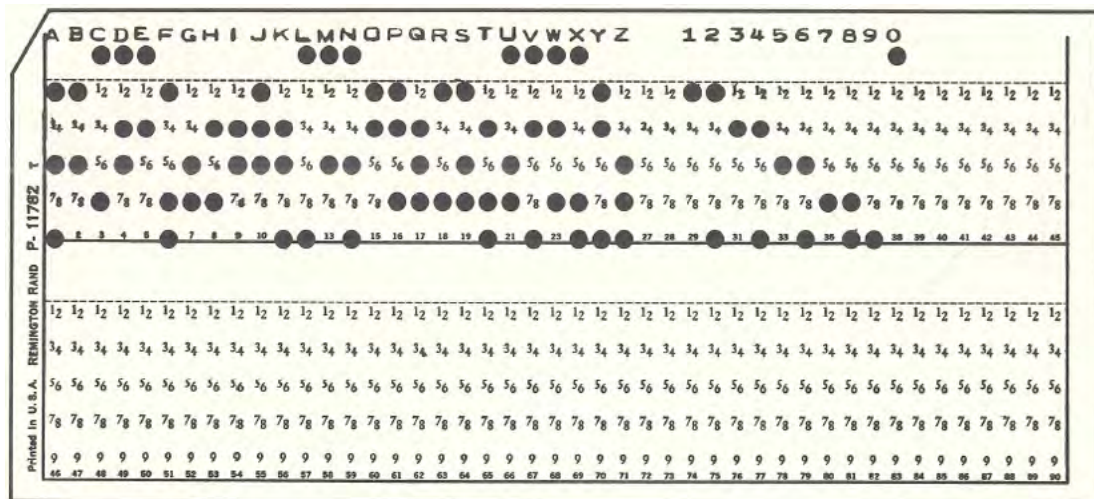


Figura 5 / Tarjeta Remington de 45 columnas.

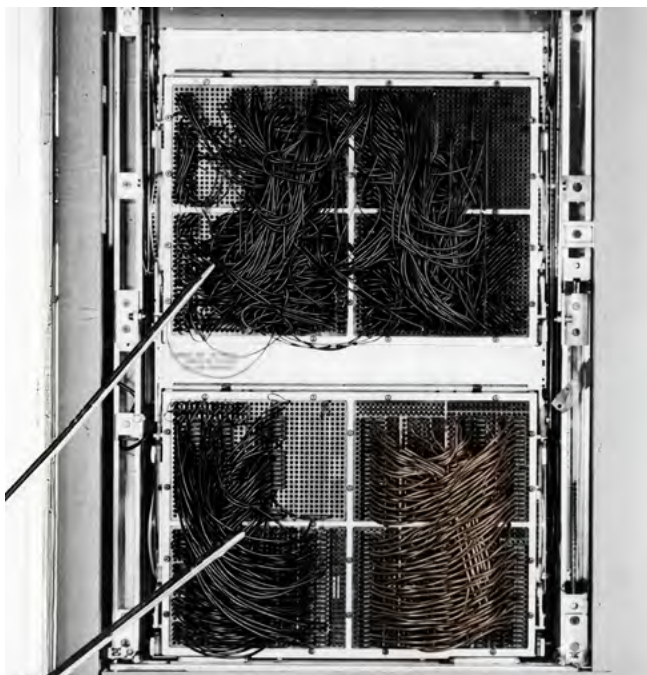
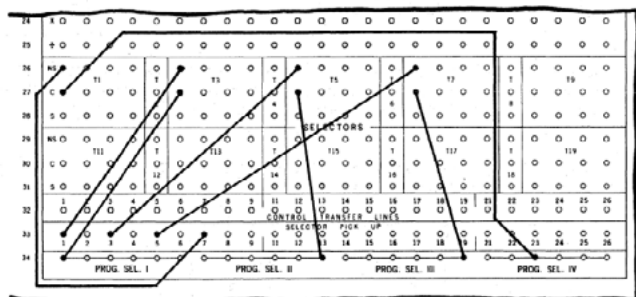


Figura 6 / Unidad de Tarjetas Perforadas" y "Unidad de Computación" del UNIVAC-60.

La Unidad de Computación ejecutaba las instrucciones "cableadas" en los paneles de Programa ("Constant & Programa Panel") y de Entrada/Salida ("Input & Output Panel"). Las "instrucciones" se especificaban conectando, a través de cables, algunos de los selectores contenidos en los paneles o placas (ver Figura 6). Los programas se cableaban y mantenían en distintos paneles para ser montados y ejecutados las veces que fuera necesario.

Los programas podían contener un máximo de 40 "steps" (instrucciones). Cada "step" consistía de 6 componentes: la operación aritmética (o alguna otra), el primer y segundo operando, el lugar para el resultado y las ubicaciones de los siguientes "steps" en caso de resultado positivo o negativo. Los operandos y el resultado podían tener de 1 a 10 dígitos. La capacidad de los almacenamientos de entrada (lectura), de salida (para perforación) e intermedio (para resultados parciales) era de 60 columnas, a diferencia de 120 columnas del UNIVAC-120 [13].

En resumen, el UNIVAC-60 representa una evolución de las máquinas construidas para realizar funciones específicas con información registrada usando perforaciones en tarjetas. La integración y ampliación de las funciones, y la posibilidad de controlarlas a través de programas cableados en los paneles de conexiones, representan la transición a los computadores de la siguiente generación programables por software.

La continuación

La venta del computador UNIVAC-60 a la CCU fue aprovechada por Remington Rand para difundir sus "departamentos de máquinas de Contabilidad, UNIVAC, Electrónico



Figura 7 / Equipamiento UNIVAC para la PAA (1967).

**[El computador UNIVAC]
superaba largamente el peso
para ser considerada una
máquina de calcular. Por lo
tanto, la Junta General de
Aduanas tuvo que crear una
partida especial.**

y Tabulación" [15]. Paralelamente, IBM [16] y NCR [17] anunciaban nuevos "cerebros electrónicos para ayudar a empresas comerciales". Posteriormente, el Instituto Chileno de Administración y Racionalización de Empresas (ICARE) en su encuentro de 1958 incluyó una presentación sobre el "nuevo sistema de computador UNIVAC electrónico que en breve deberá alcanzar un gran desarrollo de aplicación en

nuestro medio con el consiguiente beneficio para las industrias y empresas que lo adopten" [18].

Por otra parte, en 1958 las empresas mineras norteamericanas de cobre (Chile Exploration, Andes Copper y Santiago Mining) publicaron en la prensa la inserción "contribución del cobre al progreso electrónico" indicando que "la rapidez y precisión con que debe trabajar Univac han hecho imprescindible emplear en su construcción el cobre y sus aleaciones, porque es el más práctico y ventajoso conductor de electricidad que se conoce en el mundo" [19].

Pasaron algunos años para que ingresaran los siguientes computadores digitales, esta vez de segunda generación (transistorizados). En diciembre de 1961 llegó un computador IBM-1401 al Servicio de Aduanas para aplicaciones estadísticas y administrativas. En los años siguientes ingresaron otros computadores IBM-1401 a varias instituciones estatales: Impuestos Internos, Ferrocarriles, Registro Civil, Cajas de Previsión, Fuerzas Armadas.

Paralelamente, en las universidades ingresaron computadores transistorizados orientados a aplicaciones científicas y de ingeniería. En junio de 1962 la Universidad de Chile recibió el computador alemán ER-56 de la Standard Elektrik Lorenz. Seguidamente, llegaron computadores IBM-1620 a las universidades: Católica (1963), Santa María (1964) y de Concepción (1965). Por otra parte, en 1964 la Universidad Técnica del Estado (hoy USACH) recibió en donación un computador DATATRON de primera generación (tubos). Posteriormente, la Universidad de Chile adquirió equipos UNIVAC para el sistema de soporte de la Prueba de Aptitud Académica (PAA), que reemplazó al Bachillerato como prueba de selección de alumnos a las universidades desde 1966 (ver Figura 7).

A partir de 1966 ingresaron computadores de "tercera generación" (con tecnología de "estado sólido" que antecedió a los circuitos integrados) y de "propósito general", es decir, para aplicaciones "comerciales" y científicas. El computador emblemático de este período fue el IBM-360. Diversos modelos del IBM-360 se instalaron en ENAP (1966), Universidad de Chile (1966), Empresa Nacional de Computación (1968), Banco del Estado (1969) y algunas otras instituciones y empresas.

Informe de la Universidad Técnica Federico Santa María

La existencia del computador de la CCU estaba registrada en los archivos de la Universidad Técnica Federico Santa

María (UTFSM) en Valparaíso. En efecto, en el “Informe de la Comisión sobre adquisición de una máquina computadora digital (tipo IBM-1620)” para la UTFSM se menciona al computador UNIVAC-60 [20]. El informe emitido en 1960 adjunta un documento previo del Sr. Sergio Leiva titulado “Computadores Electrónicos” [21]. El documento de 17 páginas distingue entre “computadores análogos y computadores digitales”. Al respecto, en primer lugar, se revisa el “Analog Computer Mod. 3000” que dispone el Instituto Fisiológico de la Universidad de Chile desde marzo de 1959. A continuación, se describen las características de un computador analógico construido en el Laboratorio de Electrónica de la UTFSM.

La sección más importante del documento se refiere al computador digital UNIVAC-60. Se consigna que “la máquina está a cargo de un técnico y tres operarios” y se dedica “al cálculo estadístico, dividendos, jornales, costos, depreciación de activos, etc.”. Además, se afirma que “el rendimiento se considera óptimo, habiendo significado un notable aumento en la capacidad de trabajo y entre los planes futuros de la CCU está incluido el cambio de 4 máquinas mecánicas: una Clasificadora, una Detectora de tarjetas, una Intercaladora y una Tabuladora, cuyas velocidades no se ajustan bien con el computador Univac”.

Por otra parte, el documento afirma que “en el aspecto matemático, el computador Univac 60 puede efectuar la evaluación de soluciones reales de ecuaciones algebraicas de cualquier grado, funciones trascendentales, ecuaciones diferenciales, etc., basándose prácticamente en una técnica de aproximación del binomio de Newton y de las series de Fourier y de Taylor”. De hecho, se hicieron pruebas del cálculo de raíces, de funciones trigonométricas (y sus inversas) y de logaritmos. Para graficar, se per-

El UNIVAC-60 representa [...] la transición a los computadores de la siguiente generación programables por software.

foraron los valores en tarjetas y se imprimieron utilizando la máquina tabuladora.

Teniendo en cuenta la evaluación de los computadores analógicos y digitales, en septiembre de 1960 la Comisión concluyó que “en este momento no sería aconsejable la adquisición de una máquina computadora digital como la IBM 1620” (ofrecida en US\$29.800). En caso de tener que resolver un problema matemático de envergadura “sería más conveniente pedir a la CCU que nos presten su UNIVAC (pagando desde luego el arriendo correspondiente), y a partir del año 1962 recurrir al equipo ER-56 recién adquirido por la Universidad de Chile” [20].

Conclusiones

El computador UNIVAC-60 que adquirió la CCU representó elementos de continuidad y cambio respecto de la incorporación de la tecnología computacional en el país (ver Tabla 2). Continuidad, por tratarse de la evolución de la maquinaria para procesar tarjetas perforadas que hasta entonces se utilizaba para el procesamiento de datos. Cambio, por representar una revolución que significó un avance cualitativo en la rapidez y seguridad en la manipulación de la información con el potencial de abordar nuevas aplicaciones en las áreas de gestión y de ciencia.

Tecnología	Máquinas	Aplicaciones	Control	Medio
Máquinas especiales para tarjetas	lectora/perforadora clasificadora/intercaladora tabuladora	estadística contabilidad	por hardware	panel cableado
Computador digital para tarjetas	UNIVAC-60 Compañía Cervecerías Unidas 1957	contabilidad gestión ciencia	programa por hardware	instrucciones en panel cableado
Computador digital	IBM-1401 (Aduana, 1961) ER-56 (U.Chile, 1962) IBM-1620 (UTFSM, 1964)	gestión ciencia ciencia	programa por software	instrucciones en lenguajes de programación

Tabla 2 / Evolución de la tecnología de procesamiento de datos en Chile

En otras palabras, el computador UNIVAC-60 representa el “eslabón perdido” en la cadena histórica de la computación en el país. Programado físicamente por hardware, representa la transición entre las máquinas para realizar procesos específicos con tarjetas perforadas que se utilizaron desde la década del veinte, y los computadores digitales transistorizados (de segunda generación) programados por software que llegaron a Chile en la década del sesenta.

En síntesis, hace 70 años la Compañía de Cervecerías Unidas adquirió un costoso computador digital de primera generación para optimizar el procesamiento automático de su información contable. La perspectiva histórica permite concluir que la llegada del UNIVAC-60 significó un aporte significativo a la computación en Chile incorporando el primer “cerebro electrónico” en un contexto mundial y nacional de conflicto pero de indudable progreso tecnológico. **B**

Referencias

- [1] Martos, Alberto. “Breve Historia de la carrera espacial”. Ed. Nowtilus. 2009.
- [2] “Cerebro electrónico dirige al Sputnik”. La Nación. 6 noviembre 1957.
- [3] “Un cerebro mecánico verá y oirá lo que el hombre no sabe del espacio”. El Mercurio de Valparaíso. 4 diciembre 1957.
- [4] “‘Cerebros electrónicos’ contribuyen a modernizar la tesorería provincial”. El Mercurio de Valparaíso. 28 abril 1957.
- [5] “Un cerebro mecánico está en construcción”. El Mercurio de Valparaíso. 12 septiembre 1957.
- [6] “Compañía de Cervecerías Unidas: una tradición dinámica”. El Mercurio de Valparaíso. 28 julio 1957.
- [7] “Un cerebro electrónico de 1.750 kilos será izado hoy al 6° piso de un edificio”. El Mercurio de Valparaíso. 20 de octubre de 1957.
- [8] “Llegó el primer cerebro electrónico; su instalación será prueba para acróbatas”. El Mercurio de Valparaíso. 1 septiembre 1957.
- [9] “Chile adquirió su primer ‘cerebro electrónico’”. La Estrella de Valparaíso. 29 abril 1957.
- [10] “Chile adquirió su primer Cerebro Electrónico; la CCU lo instalará”. El Mercurio de Valparaíso. 29 abril 1957.
- [11] “Difícil maniobra para poder instalar cerebro electrónico”. El Mercurio de Valparaíso. 21 octubre 1957.
- [12] Coello, Carlos. “Breve historia de la computación y sus pioneros”. EFE. 2003.
- [13] Remington Rand. “UNIVAC 60 and 120 Punched-Card Electronic Computers”. https://bitsavers.org/pdf/univac/univac_60_120/SPTM4276_UNIVAC_60_120_Operating_Instructions.pdf
- [14] Brooks, Frederick; Iverson, Kenneth. “Automatic Data Processing”. John Wiley. 1963.
- [15] Aviso “Remington Rand Chile Ltda”. El Mercurio de Valparaíso. 22 agosto 1957.
- [16] “IBM instaló en Caracas su primer centro electrónico en Latinoamérica”. La Nación. 10 febrero 1957.
- [17] “Nuevos ‘cerebros’ electrónicos para ayudar empresas comerciales”. La Unión. 24 agosto 1957.
- [18] “La próxima semana habrá gran actividad en ICARE”. El Mercurio de Valparaíso. 12 julio 1958.
- [19] Aviso “Contribución del Cobre al Progreso Electrónico”. La Unión. 7 febrero 1958.
- [20] Frucht, Roberto (coordinador); Von Brand, Max; Feick, Guillermo; Stutzman, Leroy; Tierney, John. “Informe de la Comisión sobre adquisición de una máquina computadora digital (tipo IBM. 1620)”. UTFSM. 16 septiembre 1960.
- [21] Leiva, Sergio. “Computación Electrónica: Informe”. UTFSM. Septiembre 1960.

Agradecimientos

En primer lugar, a María Cecilia Cornejo, encargada del archivo de la UTFSM, que me permitió acceder al documento interno que mencionaba al UNIVAC-60 con el que se inició esta investigación. A Betty Bosque, que me proporcionó información de su padre Carlos Roberto Bosque Hull (1919–1992). A Constanza Bravo encargada del archivo del diario La Nación en la Universidad Diego Portales. Al personal de la Biblioteca Nacional que me facilitó el acceso a la prensa de Santiago y Valparaíso. Finalmente, al DCC y a Federico Olmedo (editor), y a todo el equipo de revista Bits, por apoyar el proyecto de Historia de la Computación en Chile.

Un legado de docencia,
investigación e innovación:

**Profesor José Alberto
Pino Urtubia fue
distinguido como
Profesor Emérito de la
Universidad de Chile**



Resumen / El profesor José Alberto Pino Urtubia, cofundador del DCC, recibió el título de Profesor Emérito de la Universidad de Chile tras más de 50 años de trayectoria. En este artículo, recorremos su legado y presentamos su discurso de investidura, en el que repasa, con voz propia, algunos de los hitos y personas que definieron su carrera académica.

La Casa de Bello distinguió al académico José Alberto Pino Urtubia, del Departamento de Ciencias de la Computación (DCC), con el título de Profesor Emérito, en reconocimiento a su destacada trayectoria y a su contribución fundamental al desarrollo de la computación como disciplina científica y académica en el país.

La emotiva ceremonia se realizó el miércoles 18 de marzo en el auditorio Enrique d'Etigny de la Facultad de Ciencias Físicas y Matemáticas (FCFM), y reunió a la Rectora Rosa Devés Alessandri, al decano Francisco Martínez Concha, autoridades universitarias, académicas y académicos, estudiantes, colegas, amistades y familiares, quienes fueron parte de un homenaje marcado por el reconocimiento y la gratitud.

Con más de 50 años de vínculo con la Universidad, el profesor Pino es una figura clave en la historia de la computación en Chile. Como uno de los fundadores del DCC en 1975, contribuyó a consolidar esta disciplina dentro de la ingeniería, en un contexto en que aún buscaba su lugar en el ámbito académico. Su trabajo fue fundamental en la creación e impulso de programas que hoy son pilares de la Facultad, como el Magíster en Ciencias mención Computación, la carrera de Ingeniería Civil en Computación y el Doctorado en Ciencias, mención Computación. A ello se suma su participación en el desarrollo del Magíster en Tecnologías de la Información, más de 200 publicaciones científicas y la creación de BIRDS, el primer software nacional comercializado internacionalmente.

Su labor de organización científica y académica ha sido igualmente relevante. Fue fundador y presidente de la Sociedad Chilena de Ciencias de la Computación, presidente por dos períodos del Centro Latinoamericano de Estudios en Informática (CLEI). Asimismo, cofundó y dirigió la revista Informática, primera publicación profesional del área en Chile. En el ámbito universitario, dirigió el DCC entre 1978 y 1983 y participó activamente en la Comisión de Evaluación Académica de la Facultad, la cual también presidió.

La distinción como Profesor Emérito es una de las más altas que otorga la Universidad de Chile. No sólo reconoce una trayectoria excepcional, sino también un legado que ha



El profesor José Alberto Pino, en primer plano, durante su etapa escolar en cuarto básico del Liceo de Hombres de San Felipe.



Descanso durante una conferencia, con Ricardo Baeza-Yates.

marcado el desarrollo de una disciplina y la formación de generaciones, dejando una huella profunda en la Universidad de Chile y en el país.

Una vida ligada a la Universidad de Chile

En sus palabras, la Rectora Rosa Devés destacó que la vida del profesor Pino se funde con la historia de la Universidad de Chile. "Los logros académicos también tienen espíritu, y algunas personas, por sus condiciones humanas extraordinarias, son capaces de otorgarles esa impronta con mayor luminosidad. Es ese componente de humanidad el que hace que las publicaciones derivadas de la investigación permanezcan siempre, de algún modo, ligadas a la memoria de las y los estudiantes que se formaron mientras esas ideas tomaban cuerpo; al esfuerzo colectivo desplegado para obtener los recursos que las hicieron posible y a los vínculos internacionales que abrieron espacios de intercambio y

La distinción como Profesor Emérito [...] no sólo reconoce una trayectoria excepcional, sino también un legado que ha marcado el desarrollo de una disciplina y la formación de generaciones.



Premiación en Casa Central de la Universidad de Chile, con Ramón Frederick y el entonces rector, Víctor Pérez (noviembre 2009).

diálogo fecundo. La historia de una universidad está sostenida precisamente por trayectorias como la del profesor José Pino, quienes con oficio, con trabajo, con visión y encarnando los valores esenciales, van dando forma a instituciones duraderas capaces de trascender su tiempo”, dijo.

Por su parte, el decano Francisco Martínez lo reconoció como “el ejemplo de un académico que dedicó su vida a la Facultad y que en su trayectoria ayudó a construir con un grupo de visionarios, corajudos, un nuevo departamento”.

En la presentación del homenajeado, el director del DCC, Aidan Hogan, relevó su liderazgo en momentos complejos, así como su convicción permanente en el proyecto del Departamento. “La visión que tenía hace 50 años es hoy una realidad: un DCC moderno, una comunidad vibrante y de gran impacto para la sociedad”, señaló, destacando además que este reconocimiento lo convierte en el primer Profesor Emérito del Departamento.

En un discurso cercano y reflexivo, el profesor Pino recordó los inicios del Departamento y los desafíos que enfrentaron para consolidarlo. Evocó una época en que la computación

era vista con desconfianza y en la que iniciativas como la creación de la carrera de Ingeniería Civil en Computación debieron abrirse paso con dificultad.

A través de anécdotas personales, también repasó su propia historia, desde sus años como estudiante en un liceo en San Felipe hasta su ingreso a la Universidad de Chile hace seis décadas. En ese recorrido, destacó el valor de quienes lo orientaron en su camino y reflexionó sobre el rol formativo de la academia.

A continuación reproducimos el discurso del profesor José Alberto Pino Urtubia dictado durante la ceremonia de investidura.

Discurso profesor José Alberto Pino Urtubia

Profesora Rosa Devés Alessandri, Rectora de la Universidad de Chile, Profesor Francisco Martínez Concha, decano de la Facultad de Ciencias Físicas y Matemáticas, autoridades, académicos, estudiantes, funcionarios, familia, amigos todos: buenas tardes.

Muchas gracias a la Universidad de Chile, mi universidad, por esta distinción que acepto con humildad, no por mis propios méritos, sino en nombre de muchas personas. En primer lugar, a ese pequeño grupo de jóvenes audaces que hace medio siglo creamos el Departamento de Ciencias de la Computación, en particular, Patricio Poblete, Alfredo Piquer y Fernando Silva. En seguida, a los académicos que pronto se fueron integrando y mantuvieron el entusiasmo inicial, entre ellos Jorge Olivos, Juan Álvarez, Ernesto Azorín, Rafael Hernández, José Benguría. A los académicos que llegaron después e hicieron brillar nuestro querido Departamento, en particular, María Cecilia Rivara, Ricardo Baeza-Yates, Gonzalo Navarro, Claudio Gutiérrez, Nancy Hitschfeld, Aidan Hogan, Éric Tanter, Benjamín Bustos, José Piquer, Cecilia Bastarrica, Felipe Bravo, Alejandro Hevia, Jocelyn Simmons, y muchos otros. También a esos estudiantes que en un comienzo confiaron que su futuro profesional podría centrarse en la entonces joven disciplina de la Computación. Además, a los funcionarios del Departamento, que aportaron con su destacado trabajo para que la institución pudiera progresar. No se preocupen, no voy a dar el nombre de todos en la foto... bueno, sólo de unos 300...

Hoy la Computación es una disciplina respetada. Pero hace 40 o 50 años atrás la Computación era mirada quizás con una mezcla de desconfianza, indiferencia, e incluso en algunos casos, con subestimación. Recuerdo que, en 1984, nuestra propuesta de crear la carrera de Ingeniería Civil en Computación fue aprobada en el Consejo de Docencia de la

Facultad por un margen muy estrecho de votos. En la actualidad, suena increíble que haya habido tanta resistencia. El Departamento no nació de un plan de las altas autoridades de la época, ni siquiera por la iniciativa de algún profesor senior iluminado, sino por las ideas y empuje de un grupo de muchachos inexpertos, que teníamos mucho por aprender. Nadie nos regaló nada. Cada centímetro cuadrado de espacio físico, cada peso de presupuesto, cada programa docente que desarrollamos, fueron logrados con mucho esfuerzo. La historia del Departamento es una de trabajo sostenido y grupal. Es por ello que estoy tan contento de representar a ese Departamento, que hace poco celebró sus 50 años. No es que no hayamos tenido discrepancias internas. Somos humanos. Pero supimos superarlas para no desviarnos de hacer investigación, docencia y extensión universitarias y buscar la excelencia. Por lo tanto, este premio de hoy es en realidad, para el Departamento de Ciencias de la Computación, y para todos quienes lo integraron durante su historia.

Quiero mencionar a quienes, en mi caso, me ayudaron y apoyaron en mis tareas académicas, las cuales empezaron en septiembre de 1969, cuando oficialmente fui contratado por la Universidad como ayudante. En primer lugar, agradezco a mi familia, que me ha apoyado e inspirado siempre: en particular, a mis queridos padres, Pedro José y Silvia, quienes me dieron mucho afecto y con mucho esfuerzo financiaron mis estudios, a mi querida esposa Adriana; a mis hijos Francisco, Rosario y Alberto; a mis ocho nietos, todos los cuales me han llenado de alegrías; a mi hermana Patricia, y a mis cuñados Jaime y Eli. A Carlos Pérez García, mi compañero en ir descubriendo la Computación cuando éramos estudiantes. A mi amigo Víctor Pérez Vera, quien me acogió cuando fui a mi primer postgrado en Estados Unidos y con quien escribimos libros y artículos. A mis colegas y amigos Nelson Baloian y Sergio Ochoa, con quienes hemos trabajado juntos por tantos años. A mis buenos amigos e investigadores del exterior, Pedro Antunes, Marcos Borges, Wolfram Luther y Ernst Leiss. A Luis Guerrero, Gustavo Zurita, Sergio Peñafiel, Jonathan Frez, Flavia Santoro, David Fuller, todos coautores científicos frecuentes. A mis numerosos colegas y amigos de otros departamentos. A mis ex tesis de doctorado, por contagiarme su entusiasmo juvenil por la investigación: Valeria Herskovic, Orlando Erazo, César Collazos, Olivier Motelet, Andrés Neyem y Jens Hardings. A mis ex tesis de Magíster, en particular al primero, Mario Jofré. A mis 88 ex memoristas de Ingeniería Civil en Computación, que no puedo nombrar todos, sólo a manera simbólica el primero de ellos, que también fue el primer titulado de esta carrera, Ronald Corovic. A Margarita Serei, inolvidable jefa administrativa y amiga, a Francia Slimming, mi fiel secretaria por más de 30 años. En fin, siempre quedará corto en agradecer a tantas personas que han hecho posible esta vida académica.



Autoridades universitarias, académicos y académicas, estudiantes, colegas, amistades y familiares acompañaron la ceremonia de investidura como Profesor Emérito de José Alberto Pino Urtubia.



En un emotivo discurso, el profesor José Alberto Pino Urtubia recordó los inicios del Departamento y los desafíos de impulsar la computación y la carrera de Ingeniería Civil en Computación en sus primeros años.

Ahora me gustaría destacar a tres personas, muy distintas entre sí, que de alguna manera, me ayudaron a definir mi carrera académica en algún momento de mi vida.

La primera persona es Don Manuel Labrind, quien fue rector del Liceo de Hombres de San Felipe, institución en la que estudié. El liceo tenía una buena tradición, en él había estudiado el Presidente Pedro Aguirre Cerda, quien tenía como lema "Gobernar es educar". Sin embargo, el liceo tenía escasos recursos materiales y humanos. De hecho, era usual que los estudiantes no tuviéramos clases de alguna asignatura durante todo un año. Don Manuel, además de rector, fue mi profesor de Historia. Me prestó libros de literatura durante varios años. Cuando estaba en el penúltimo año de mi liceo, pregunté a mis padres qué debería estudiar



Profesor José Alberto Pino Urtubia junto a la rectora de la Universidad de Chile, Rosa Devés, durante la ceremonia de investidura como Profesor Emérito.

después. Ellos no habían ido a la universidad, así es que simplemente me respondieron que lo que yo quisiera, que ellos me apoyaban de todo corazón. Para saber qué estudiar, comencé a averiguar. Recuerden que en 1964 no había Internet, ni computadores personales, ni teléfonos celulares. La información pública era escasa. Decidí entonces, escribir a una universidad pequeña pidiendo información sobre las carreras que dictaban. A vuelta de correo postal, recibí un folleto que describía las carreras que la universidad ofrecía. Mi interés iba por el área científica/matemática y allí vi que había carreras de varias especialidades de ingeniería civil, así como carreras de técnico. La descripción de las ingenierías decía que éstas eran de altísimo nivel, de máxima exigencia y que sólo muy pocos postulantes eran aceptados. Las carreras de técnico, en cambio, eran descritas para personas más normales, me pareció a mí. Más aún, como yo tenía vacíos de formación en matemáticas y física, quizás era buena idea elegir entre las carreras técnicas. Así es que elegí la de Técnico en Electricidad. Poco tiempo después, el rector Labrind me prestó un libro, y aproveché de preguntarme si había decidido qué hacer después del liceo. Le conté, entonces, que había decidido estudiar técnico en electricidad en una universidad pequeña. El rector me respondió entonces con una frase enfática que al comienzo me desconcertó y que hasta el día de hoy aún recuerdo.

“No, señor, si a usted le gusta esa área, usted va a estudiar Ingeniería y en la Universidad de Chile”. Como me quedé callado, me preguntó si le había escuchado bien. Cuando le respondí que sí, me dijo que esa era la carrera para mí. Bueno, esa es una de las razones por las cuales hoy estoy aquí. En caso contrario, tal vez habría hecho la instalación eléctrica en la casa de más de alguno de ustedes. Quizás alguien diga que don Manuel fue autoritario y que debió haberme explicado más las razones para su convicción. Puede ser, pero estoy muy contento de que, con el conocimiento y experiencia tuyas, me haya ayudado en un momento en que yo mismo tenía muy pocos elementos para tomar una decisión importante. Es por eso que lo he elegido como la primera persona que influyó en mi carrera.

La Universidad de Chile me aceptó como alumno y el primer año fue muy duro, pero pasé todos mis cursos. Al comienzo de mi segundo año conocí a otra persona de interés. Ese año de 1967, por primera vez en la Escuela de Ingeniería se daría un curso semestral de Computación. Al asistir a la primera clase de ese curso, el profesor estaba muy nervioso y de todo lo que dijo no logré discernir de qué hablaba. Usaba muchas palabras en inglés, que no tenían explicación. Así es que salí de clase desorientado, porque francamente no había entendido nada. Caminando por el patio, me encontré con el zócalo del edificio de Química, que vi tenía un letrero “Centro de Computación”. Bueno, pensé, quizás aquí pueda encontrar algo útil. Entré a un pasillo poco iluminado y nadie a la vista. De pronto, aparece por una puerta, un joven, con cara amistosa, que me dice: “Caballero, ¿en qué lo podemos servir?”. Le expliqué mi problema, y sonriendo, me respondió que lo siguiera. Entramos a una sala grande, semi a oscuras. Adentro había lo que me pareció un monstruo electrónico, con muchos estantes con circuitos y cables. Algunos estantes tenían pequeñas lucecitas que me parecieron los ojos del monstruo que parpadeaban. El joven se acercó a lo que llamó “la consola”, con más lucecitas. “¿Qué te gustaría que calculáramos?”, dijo, “algo largo, como una serie de Taylor”. Le dije una serie con 500 términos y se sentó entonces frente a un teletipo y escribió allí en lenguaje de máquina, lo que denominó un “programa”. Tomó la cinta de papel perforado producida por el teletipo, y la puso en otro estante, que dijo, era una “lectora de cinta de papel perforado”. El monstruo, en una especie de suspiro, se tragó la cinta. El joven se sentó frente a la consola y me dijo “veamos ahora cuánto demora el cálculo”. Apretó un botón, parpadearon lucecitas y al instante, se escuchó la perforación de cinta de papel. El estudiante recogió la cinta, fue al teletipo, y allí se imprimió el resultado. Con aire de triunfo, me miró y preguntó: “¿qué tal, te presento al computador ER-56”. Yo no lo podía creer, sin embargo, allí estaba ese monstruo esperando personas que le pidieran cálculos y quizás más. Quedé fascinado. Me dedicaría a instruir ese

monstruo, a trabajar con él. Un poco atolondrado, le pregunté a mi anfitrión como se llamaba él. "Dittmar Krall", me respondió, "ahora estoy cursando cuarto año de Ingeniería Eléctrica". En seguida, le pedí que me prestara algún libro para introducirme en el tema. Me pasó un manual de lenguaje de máquina del ER-56. No era precisamente un libro didáctico, pero poco me importó. Pronto pasé a programar ese computador y también el IBM 360/40 que acababa de llegar a la Facultad. Mi carrera profesional estaba decidida; seguí la especialidad de Ingeniería Matemática, porque no había una de computación, pero mi disciplina era esta última. ¿Qué había hecho Dittmar por mí? Muchísimo. Hoy es trivial el cálculo que tanto me fascinó en marzo de 1967, pero Dittmar hizo lo que no pudo el profesor del curso: en unos minutos mostrarme un universo nuevo, motivarme tanto que marcó mi futuro profesional y académico.

Varios años después fui, recién casado, a estudiar postgrado en Ciencias de la Computación en la Universidad de Michigan, Estados Unidos. Uno de mis profesores jóvenes fue Arnie Rosenthal, quien también fue mi primer profesor guía allí. Un día, Arnie me invitó a conversar a su oficina. "Sabes?", me dijo, "tengo un problema que creo tú puedes resolver". Me explicó entonces, que había diversos algoritmos publicados para resolver problemas específicos de centralidad en ciertos grafos. Más aún, estos algoritmos eran ineficientes. Su apuesta era que los problemas podrían resolverse con un sólo algoritmo general y eficiente. La dificultad, por supuesto, era que había que inventar ese algoritmo, si es que existía. Ese resultado sería para publicarlo en un journal de primer nivel, me dijo. Las próximas semanas me dediqué a pensar cómo podría ser ese algoritmo. Dibujaba grafos y discurría cómo era posible calcular el centro. Ni siquiera el centro era concreto, ya que había varias definiciones de centralidad. Pasaba horas concentrado, aún en sueños trataba de inventar el algoritmo. "Parece que estuvieras en otro planeta", decía mi esposa, con toda razón... pero en realidad estaba en la luna. Hasta que de repente, sucedió el milagro. Se me ocurrió un algoritmo. Lo programé y probé, lo analicé en su eficiencia, en sus casos límites, etc. Todo funcionaba, y era eficiente, tal como Arnie había intuido. Lleno de alegría fui a verlo. En el camino, pensaba que me iba a felicitar, que tendríamos una pequeña fiesta. Pero no. Cuando me escuchó, respondió sin inmutarse "buen trabajo; ahora tienes que escribir el borrador del artículo correspondiente", sin siquiera mirar el algoritmo. Así es que eso hice. Ahora, Arnie leyó, se centró en el algoritmo y me comentó: "el algoritmo está demasiado largo; tienes que acortarlo". Uds. comprenderán lo difícil que es cambiar un algoritmo que uno mismo ha inventado soñando con él. Pero lo hice. Lleno de esperanza, se lo llevé. Ahora escribió: "los recorridos del grafo son muy raros, usa recorridos estándares". ¡Otra iteración más, uff! De hecho,

Hoy la Computación es una disciplina respetada. Pero hace 40 o 50 años atrás la Computación era mirada quizás con una mezcla de desconfianza, indiferencia.

fueron muchas iteraciones adicionales. A esas alturas, confieso que había perdido toda esperanza de que ese artículo alguna vez pudiera enviarse a un journal. De pronto me decía "empeoró el algoritmo; volvamos a la versión anterior". Finalmente, después de mucho tiempo, Arnie me informó que le parecía bien. Mandamos el artículo al journal y fue aceptado para publicación. Ahí finalmente celebramos. Pasó tiempo hasta que el artículo se publicó. Mi primer artículo de journal. Yo pensé en algún momento ¿por qué Arnie me hizo trabajar, y volver a trabajar, cuando él mismo podría haber hecho las mejoras al algoritmo y seguramente, con su experiencia, habríamos acortado mucho los tiempos? En ese momento, no tenía respuesta, pero después lo entendí perfectamente. Lo que Arnie me enseñó es a hacer investigación científica, lo que significa explorar todo, ser riguroso y exigente, a meterse por calles sin salida para después volver y buscar por otras vías, a sudar mucho. Al investigar, uno está en terreno desconocido, por lo tanto, la tarea no es fácil. Toda esta educación, Arnie la hizo sin aspavientos, sin grandes titulares del tipo "voy a enseñarte a investigar". Me enseñó algo que no se aprende leyendo, sino experimentándolo y sufriendolo. Después no seguí trabajando en algoritmos, sino en otras áreas, pero el aprendizaje no se perdió, pude seguir aplicándolo. Arnie había cambiado para siempre mi manera de enfrentar los desafíos científicos.

Señoras y señores, hace 60 años que ingresé a estudiar Ingeniería en la Universidad de Chile, siguiendo la sabia recomendación de mi rector de liceo. 60 años son pocos para una institución más que centenaria como esta universidad, fundada en 1842. Pero son suficientes para que un departamento en una disciplina como la Computación nazca, se desarrolle y florezca. También 60 años son suficientes para aprender que el mejor regalo que un ser humano puede recibir es un buen consejo. Especialmente ahora, en que estamos inundados de contenido, nosotros los profesores, no deberíamos centrarnos sólo en proveer aún más información, sino también en dar un consejo acertado y oportuno, iluminar un camino, dar una guía esclarecedora y motivadora a nuestros estudiantes y discípulos. Este regalo, si lo van a obtener alguna vez, será sólo por acción nuestra. Invito, entonces, a todos, a ser también Labrinds o Kralls o Rosenthals para quienes lo merecen tanto: nuestros estudiantes. Muchas gracias. **B**

Estudiantes DCC



En esta sección de la Revista estudiantes recientemente *graduadxs* del Departamento de Ciencias de la Computación (Universidad de Chile) nos cuentan, junto a sus *profesorxs* guías, sobre sus trabajos de memoria y/o tesis.

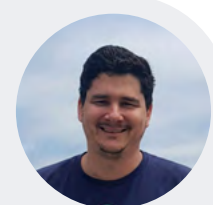
Video-Text Translation for Bridging Vision and Language

Estudiante

Jesus Perez-Martin

Profesores guías

Benjamín Bustos y Jorge Pérez



Soy Jesus Perez-Martin, Ph.D. del Departamento de Ciencias de la Computación de la Universidad de Chile. Mi formación previa es de Licenciado en Ciencias de la Computación por la Universidad de La Habana. Estuve un par de años trabajando entre la industria y la docencia universitaria, y mi interés por el aprendizaje automático aplicado me llevaron a investigar la intersección entre visión y lenguaje. Hoy, además de mi trabajo doctoral, me desempeño como Staff Data Scientist en Walmart, donde desarrollo y escaló soluciones de IA orientadas a contenido y experiencia de e-commerce.

Hay una idea simple detrás de mi doctorado: *si un sistema puede “entender” un video y contarlos con palabras —y, a la vez, transformar texto en contenido visual— entonces podemos construir puentes mucho más sólidos entre visión y lenguaje*. Eso habilita desde accesibilidad (descripciones automáticas) hasta nuevas formas de búsqueda y generación de contenido en contextos reales como el e-commerce.

Mi investigación se centra en tres líneas conectadas entre sí. Primero, *video→texto* (*video captioning*): cómo generar descripciones que no sólo sean correctas, sino también naturales. En muchos modelos, el texto termina siendo genérico (“alguien hace algo”) porque es difícil capturar qué está pasando y expresarlo bien. En mi trabajo, el foco fue mejorar la forma en que el modelo combina la evidencia visual con representaciones más “lingüísticas” (semántica y sintaxis) para producir descripciones más informativas y mejor estructuradas.

Segundo, *descripción densa de video* (*dense video captioning*): en vez de una sola descripción para todo el video, se busca detectar eventos (qué ocurre y cuándo ocurre) y describirlos con precisión. Aquí el desafío es doble: localizar temporalmente los segmentos relevantes y evitar redundancias, manteniendo coherencia. Parte importante de mi tesis propone hacerlo de forma más integrada, con una visión de “un solo paso”, pensando además en escenarios cercanos al tiempo real.

Y tercero, *texto→video*, con un énfasis aplicado: cómo pasar de una frase o consulta corta a un video que sea coherente con esa intención. En este punto conecto directamente

con una línea que me interesa mucho por su impacto práctico: *traducción automática neuronal* (NMT) centrada en contenido de sitios web de e-commerce. En e-commerce, el lenguaje tiene particularidades: hay marcas, atributos, restricciones de estilo, formatos, y una necesidad permanente de consistencia. En mi caso, exploré un *pipeline* donde la traducción especializada (por ejemplo, español→inglés) ayuda a condicionar mejor la generación, especialmente cuando el modelo generativo está entrenado con datos mayoritariamente en inglés.

Como suele ocurrir en investigación, no todo fue “proponer un modelo y listo”. Uno de los retos más fuertes fue evaluar de manera justa: en generación (tanto de texto como de video), las métricas automáticas no siempre reflejan lo que una persona considera “bueno”. Además, trabajar con video significa lidiar con una señal rica, larga y ruidosa: hay que aprender a seleccionar lo relevante y a representar la temporalidad de forma efectiva. También fue un desafío permanente “traducir” los avances técnicos a una historia clara: explicar la contribución con simpleza sin perder precisión, algo que en la práctica se vuelve una habilidad tan importante como la parte algorítmica.

En lo personal y profesional, mi trayectoria hoy está muy conectada con la industria. Actualmente trabajo como Staff Data Scientist en Walmart, en temas ligados a experiencia de e-commerce y modelos de lenguaje/traducción aplicados a contenido. En ese sentido, mis planes tras el doctorado son seguir profundizando esta intersección entre investigación y producto, empujando soluciones que puedan pasar de un paper a un sistema real: modelos multimodales que entiendan mejor el comportamiento del usuario, que generen contenido útil y consistente, y que mejoren la experiencia en plataformas a gran escala.

Mirando hacia atrás, lo más valioso del doctorado fue aprender un “oficio”: formular problemas, diseñar soluciones, validarlas con rigor y comunicarlas de forma convincente. Ese aprendizaje queda para siempre, y es el puente que me permite moverme con naturalidad entre la frontera científica y el impacto en aplicaciones reales.

Gradual Retrieval, Ranked Enumeration, and Lazy Evaluation of Graph Database Joins

Estudiante Asunción Gómez
Profesor guía Gonzalo Navarro



En bases de datos de grafos, el *join* es de las operaciones más costosas y, al mismo tiempo, más frecuentes. Los sistemas tradicionales usan algoritmos de *join* de a pares, lo cual se ha demostrado que es subóptimo. Nuevos enfoques consisten en hacer el *join* de múltiples tablas a la vez, lo cual puede reducir el tiempo y espacio, y ha permitido diseñar algoritmos *worst-case optimal*, es decir, óptimos en el peor caso. El uso de estructuras de datos compactas ha permitido ir más allá: no sólo crear algoritmos *worst-case optimal*, sino también reducir considerablemente el espacio requerido, sin afectar la eficiencia en las operaciones. Arroyuelo et al. (2018) diseñaron un nuevo algoritmo de *join* basado en el uso de *qdags*, una versión compacta de los árboles *quadtree*. En su algoritmo buscan optimizar el tiempo de cómputo total del *join* así como el espacio en memoria. Si bien este algoritmo es competitivo e incluso más eficiente que otros algoritmos usados en los actuales sistemas de bases de datos de grafos para calcular todo el *join*, en muchos casos nos basta con encontrar unos pocos resultados.

El objetivo de esta tesis es diseñar e implementar estrategias para disminuir el tiempo de obtención de algunas tuplas del *join*. La primera parte de este trabajo, "Gradual retrieval", busca entregar resultados lo antes posible. La segunda parte, "Ranked enumeration" busca obtener aquellos resultados más importantes ordenados de acuerdo a un peso o prioridad. La tercera parte, "Lazy qdags", tiene como objetivo implementar la versión perezosa de los *qdags*, la cual permite evaluar toda el álgebra relacional (no sólo el *join*) sobre bases de datos y, además, puede disminuir el tiempo de cómputo de los primeros resultados en algunos casos. El código del proyecto se encuentra público en www.github.com/asugomez/qdags.

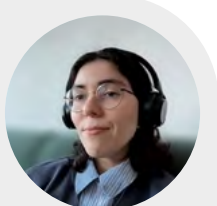
Este trabajo fue presentado en el 2do Congreso de Estudiantes de Postgrado de la Universidad de Chile (su póster quedó en tercer lugar). Un resultado en particular fue luego extendido en conjunto con otros investigadores de A Coruña, España, y publicado en el artículo:

Antonio Fariña, Adrián Gómez-Brandón, Asunción Gómez-Colomer y Gonzalo Navarro. *Cache-Friendly Compressed Boolean Matrices*. Proc. SPIRE'25, págs. 95–108.

Asunción fue una estudiante muy entusiasta, participando en todas las oportunidades que se le presentaron donde compartir su investigación, incluyendo workshops del IMFD, del DCC, y otros. Hoy en día realiza su doctorado en Francia.

An RDF Database Compact Representation for Time- and Space-Efficient Regular Path Queries

Estudiante Josefa Robert
Profesor guía Gonzalo Navarro



La tesis contribuye al problema de manejar y consultar bases de datos tipo Resource Description Framework (RDF) de manera eficiente, con un enfoque en las Two-way Regular Path Queries (2RPQs). La primera contribución de este trabajo es la implementación de una representación compacta para grafos introducida por Navarro, junto con un conjunto de métodos para extraer información valiosa del grafo. Desarrollamos, sobre esta representación, un algoritmo para resolver 2RPQs, que utiliza autómatas de Glushkov para manejar las expresiones regulares y que simula el autómata

en el grafo para identificar los caminos válidos. Para llevar a cabo esto, se utilizan en gran medida los métodos implementados anteriormente para el grafo. Evaluaciones empíricas muestran mejoras en la velocidad de procesamiento de las consultas y en el uso del espacio en comparación con soluciones existentes, particularmente para 2RPQs que involucran una única variable. En general, esta tesis presenta un enfoque competitivo para resolver 2RPQs y establece un rumbo para futuras investigaciones en el área.

Los resultados de este trabajo fueron publicados en una conferencia:

Gonzalo Navarro y Josefa Robert. *Compressed Graph Representations for Evaluating Regular Path Queries*. Proc. SPIRE'24, págs. 218–232.

Josefa fue una alumna muy independiente, que demostró gran madurez académica. En particular, se encargó ella de buena parte de la escritura del artículo.

MobilityHexVis: Analyzing Socioeconomic Urban Mobility Patterns through Small Multiples Visualization**Estudiante** Alan Acevedo**Profesores guías** Eduardo Graells-Garrido y Horacio Samaniego

El estudio “MobilityHexVis: Analyzing Socioeconomic Urban Mobility Patterns through Small Multiples Visualization”, desarrollado por Alan Acevedo en su tesis de magíster en el DCC junto a Eduardo Graells-Garrido (DCC) y Horacio Samaniego (Universidad Austral de Chile), aborda la dificultad de analizar fenómenos socioespaciales con datos masivos de telefonía móvil. Entender el desplazamiento de grupos según su nivel socioeconómico es necesario para la plani-

ficación urbana, pero el volumen de datos complica su análisis. MobilityHexVis es una plataforma interactiva basada en *small multiples* y glifos hexagonales. Esta herramienta permite comparar patrones de movilidad por nivel socioeconómico y distancia. Cada hexágono usa una división bicolor para mostrar proporciones de origen y destino, e integra el índice de Gini para medir la mixtura social urbana.

La relevancia de esta investigación está en la integración de visualización de datos y ciencia urbana, con una publicación en la revista “Environment and Planning B”. El sistema, aplicado en Santiago, identificó centros de empleo y brechas en la movilidad según el estrato social. El proyecto destaca por su enfoque multidisciplinario, donde la computación aporta evidencia para el diseño de ciudades inclusivas. Esta labor, vinculada a un proyecto Fondecyt Regular, valida la calidad del postgrado en el DCC. El trabajo conjunto entre estudiantes y académicos genera tecnología con impacto social y científico, al proponer herramientas computacionales para resolver problemas urbanos complejos.

Desarrollo de lineamientos para el deployment personalizado del producto de software TIMA**Estudiante** Cristóbal Carmona**Profesor guía** Sergio Ochoa

El objetivo de esta tesis fue generar lineamientos específicos para mejorar el proceso de *deployment* personalizado del software TIMA (TIMining Aware), un gemelo digital 3D de apoyo a faenas mineras. Cada faena minera es particular y, debido a esto, cada proceso de *deployment* de TIMA se vuelve particularmente complejo. La complejidad surge a partir de la cantidad y diversidad de actividades que el gemelo digital debe apoyar, y de la variedad de datos que debe manejar en cada faena. Esta heterogeneidad genera alta in-

certidumbre e ineficiencia, y también una alta dependencia de personal calificado.

Los lineamientos generados en el marco de esta tesis estuvieron enfocados en abordar esta problemática. Particularmente, se definieron guías específicas para abordar tres aspectos clave: 1) automatización de subprocesos “cuello de botella”, 2) testing automatizado de funcionalidades, y 3) uso de buenas prácticas para la definición de conectores y procesos.

Los resultados de aplicar estos lineamientos fueron exitosos, facilitando el proceso de *deployment* personalizado de TIMA según las características de cada faena minera. También redujo significativamente la dependencia de personal calificado para llevar a cabo este proceso. Aunque este trabajo logró los objetivos planteados, la diversidad de faenas mineras es tan amplia que hace que se requieran más pruebas en producción para robustecer las capacidades de este gemelo digital.

Postulaciones abiertas



Magíster en **TECNOLOGÍAS DE LA INFORMACIÓN**

Lidera la gestión, desarrollo e innovación de soluciones TI

Áreas de especialización:

- Ciencia e Ingeniería de Datos
- Gestión de Proyectos Informáticos
- Ingeniería de Software
- Inteligencia Artificial
- Protección de Datos Personales

 Duración:
2 años

 Modalidad:
Online sincrónico

COORDINACIÓN ACADÉMICA



Sergio Ochoa

Doctor en Ciencias de la Computación.
Profesor Asociado del DCC, U. de Chile.

**Conoce el
programa**



CONTACTO

 ec@dcc.uchile.cl

 +56 9 8434 8251

 dcc.uchile.cl/mti

Departamento de
Ciencias de la Computación

EDUCACIÓN

CONTINUA DCC

Potencia tu carrera en tecnología

 Clases online sincrónicas

 Enfoque práctico

 Certificación
Universidad de Chile

Más
información



CONTACTO

 ec@dcc.uchile.cl

 +56 9 8434 8251

 dcc.uchile.cl/ec

   /educacioncontinuada

DIPLOMAS

Actualiza y potencia tu perfil profesional en áreas clave de la computación.

- Ciencia e Ingeniería de Datos
- Gestión de Proyectos Informáticos
- Ingeniería de Software
- Inteligencia Artificial
- Protección de Datos Personales
- Python Aplicado a la Ciencia de Datos
- Tecnologías de la Información

MAGÍSTER EN TECNOLOGÍAS DE LA INFORMACIÓN

Especialízate en el desarrollo, adopción y gestión de tecnologías de la información.

PROGRAMAS CORPORATIVOS

Capacita a tus equipos con programas diseñados a la medida.

Metodología:

- Identificamos desafíos y brechas de competencias.
- Creamos un programa a medida de tu organización.
- Desarrollamos la capacitación con expertos del DCC.
- Medimos resultados e impulsamos la mejora continua.



UNIVERSIDAD
DE CHILE



dcc
CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE CHILE



B