MAPAS DEL LABERINTO
UNA SERIE DE PHANTOM MAZE — AI & LANGUAGE LAB

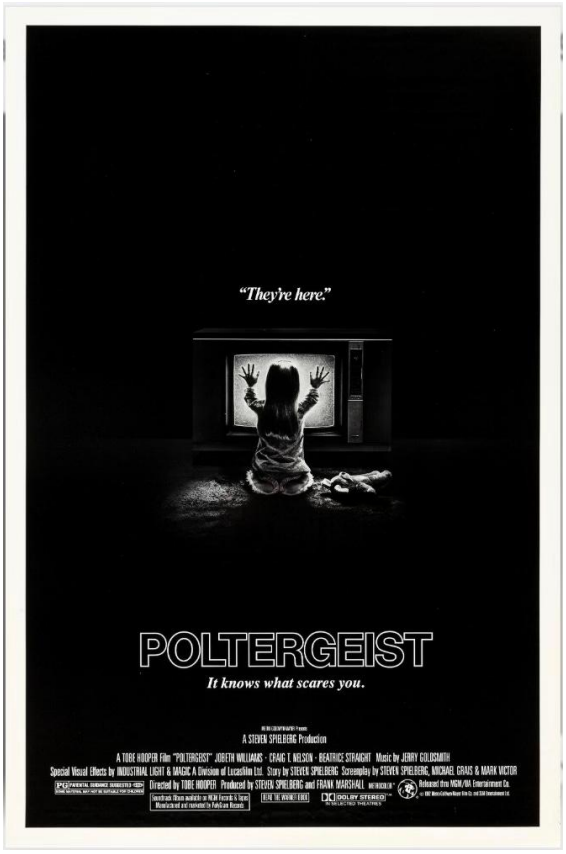
ENSAYO 21

Agentes de IA: el poltergeist en la máquina

Claudia Marsico · Hernán Inverso

DOI 10.5281/zenodo.24023233

¿Qué pasa cuando los fantasmas pueden mover cosas?



Afiche cinematográfico de Poltergeist, dirigida por Tobe Hooper y producida por Steven Spielberg, 1982.

Durante un tiempo que parece lejano y no lo es tanto los modelos de lenguaje hablaban, solo hablaban, dejando las cuestiones del mundo por cuenta nuestra. De pronto los vemos abrir archivos, ejecutar programas, ordenarnos el correo y encadenar pasos para cumplir encargos trabajosos. Delegarles tareas es uno de los mayores cambios en la inteligencia artificial. Ni hay que decirlo. Ahora bien, venimos llamando a los modelos de lenguaje "fantasmas sin ego", con un giro que busca captar esa voz en la que nos habla la tradición. Ya era sin duda inquietante, pero ahora que mueve cosas lo es

más todavía. Podríamos decir que con el desarrollo agéntico nuestro fantasma aprendió a ser poltergeist, ese "espíritu que hace ruido" que el cine volvió doméstico en *Poltergeist* (1982), producida por un joven Steven Spielberg. El televisor, otra pantalla, era la puerta de entrada de una presencia que lo revolvía todo. Ya en 1848, Catherine Crowe recogía en *The Night-Side of Nature* relatos de espíritus invisibles que se manifestaban desplazando cosas, justo como cuando vemos abrirse y cerrarse ventanas en la pantalla.

Afiche cinematográfico de *Poltergeist*, dirigida por Tobe Hooper y producida por Steven Spielberg, 1982.

La palabra es transparente. Agente viene del latín *agens*, participio de *agere*, "actuar", el mismo verbo que dio "acto", "acción", "agenda" y también en español "agitar". En términos amplios, el agente ejecuta una acción, frente al paciente, que la recibe. En el derecho, en cambio, un agente es, de modo más específico, el que obra por cuenta de otro, como el apoderado que firma en nuestro nombre o el emisario que habla por el rey, es decir, cualquiera que actúa con un poder ajeno, con toda la historia de mayordomos, procuradores y embajadores acurrucada detrás.

El sueño de un asistente así es antiquísimo. En el canto XVIII de la *Ilíada*, Homero describe el taller de Hefesto y se detiene en dos maravillas. Unos trípodes con ruedas de oro que van solos a la asamblea de los dioses y vuelven sin que nadie los empuje y unas sirvientas de oro con figura de muchacha, que tienen mente, voz y fuerza como para sostener al dios cojo cuando camina. Son, con casi tres mil años de anticipación, los primeros agentes artificiales de Occidente actuando por cuenta de su dueño. La misma fantasía reaparece en Talos, el gigante de bronce que patrullaba las costas de Creta, en el Gólem de barro que trabaja y vigila, en el genio que cumple encargos. En 1920 le llegó el nombre definitivo cuando en su obra *R.U.R.* el checo Karel Čapek bautizó a esos sirvientes artificiales *robots*, de *robota*, trabajo forzado. Čapek contó después que la palabra se la había sugerido su hermano Josef mientras pintaba y no lo convenció demasiado, porque quería algo más latino. En cualquier caso, el criado que hace la tarea que no queremos hacer sin pedir nada es un deseo humano inmemorial.

La palabra es de las más viejas de la inteligencia artificial. Cuando Stuart Russell y Peter Norvig escribieron en 1995 el manual que formó a generaciones enteras, definieron la disciplina como el estudio de los *agentes racionales*, pensando en sistemas que perciben su entorno con sensores, deciden y actúan sobre él con actuadores, para cumplir un objetivo. Un termostato es un agente mínimo; un robot aspirador, uno un poco más rico; un programa que juega al ajedrez, otro distinto. El corazón de la idea está en el lazo de percibir, decidir, actuar, mirar el resultado y volver a empezar. El aprendizaje por refuerzo, la rama que en 2016 le ganó a uno de los mejores jugadores de go del mundo con AlphaGo, es literalmente la ciencia de ese lazo, de modo que cuando hoy decimos "agente de IA" estamos reflotando ese vocabulario y su larga historia.

Lo curioso, sin embargo, es que el modelo de lenguaje llegó como un profeta que solo habla y en su forma desnuda solo predice la palabra siguiente sin sensores ni manos, más cerca del oráculo que del mayordomo. El trabajo de los últimos años consistió en volverlo agente desde afuera, envolviéndolo en lo que le faltaba. Al final, hicimos un fantasma en la máquina, esa entidad imposible con que Gilbert

Ryle, en *El concepto de lo mental* (1949), condenó al dualismo cartesiano y su idea de una mente alojada en el cuerpo para gobernarlo. Parecía un error de categoría, pero le dimos una máquina al fantasma sin ego que puede manejar con suma destreza.

En 2022, un grupo vinculado con Princeton y Google publicó un trabajo llamado "ReAct" por *reason and act*, razonar y actuar. En vez de producir una respuesta, el modelo escribe un pensamiento y después una acción en un formato fijo, del tipo "busco tal dato". En rigor, el modelo no ejecuta nada, solo emite el pedido. Un programa que lo envuelve, el arnés, lee ese pedido, corre de verdad la búsqueda o el código y le devuelve el resultado pegándolo en el contexto, que funciona como la memoria de trabajo del agente. El modelo lee lo que volvió, escribe el pensamiento siguiente, pide la acción siguiente y así en un bucle hasta terminar. Al año siguiente, en Meta, Timo Schick y su equipo entrenaron un modelo, Toolformer, para que aprendiera solo cuándo llamar a una calculadora, a un buscador o a un traductor. Se iba configurando así una arquitectura en la que el modelo no necesitaba realizar todas las operaciones por sí mismo. Podía solicitar una acción, recibir su resultado y utilizarlo para decidir qué hacer a continuación. Para eso necesitaba un programa que ejecutara sus pedidos, herramientas a las que recurrir y un contexto donde conservar la información obtenida. El agente surgía de la combinación de esos elementos: un modelo que decide, un sistema que ejecuta sus decisiones y un circuito que permite observar los resultados y ajustar los pasos siguientes hasta alcanzar un objetivo.

La novedad es tentadora y de golpe parece que todo es agéntico. Sin embargo, hay que separar un flujo de trabajo, donde los pasos están escritos de antemano por un programador, de un agente propiamente dicho, donde el modelo decide sobre la marcha qué paso dar. Pensemos en un asistente que organiza viajes. Si pide destino y fechas, busca vuelos y hoteles y ordena los resultados por precio, puede seguir una secuencia programada de antemano. Un agente, en cambio, podría recibir el objetivo "organiza mi viaje a un congreso con este presupuesto" y decidir cómo resolverlo. Si los vuelos resultan demasiado caros, puede explorar fechas alternativas; si encuentra un hotel alejado, calcular si el transporte compensa el ahorro. La distinción depende de quién decide el recorrido y cómo ajusta ante los resultados. La autonomía admite grados, pero una parte considerable de lo que hoy se vende como agéntico sigue pasos y condiciones establecidos por sus programadores.

Para realizar esas acciones, el modelo necesita conexiones con herramientas y fuentes de información. Cada conexión requiere que el sistema pueda solicitar una operación y recibir su resultado. En noviembre de 2024 Anthropic presentó el Model Context Protocol (MCP), un protocolo abierto que ofrece una forma común de establecer ese intercambio y reduce la necesidad de desarrollar integraciones distintas para cada aplicación, lo cual facilita la ampliación de recursos disponibles para un modelo. El grado de autonomía depende, además, del margen que tenga para decidir cuándo y para qué usarlos.

En 2023 todo eso estalló en público. Aparecieron AutoGPT y BabyAGI, dos programas que le daban a un modelo un objetivo y lo dejaban suelto, generando y ejecutando sus propias subtareas en un bucle sin fin. El primero juntó cien mil estrellas en GitHub en pocas semanas. Eran frágiles, se

perdían, entraban en círculos, pero probaron que la autonomía era posible. Ese mismo año, un equipo de Stanford encabezado por Joon Sung Park armó Smallville, un pueblo simulado donde vivían veinticinco agentes con memoria. Se despertaban, charlaban, se formaban opiniones y, en el experimento más citado, uno decidió dar una fiesta de San Valentín y los demás se enteraron, se invitaron entre ellos y llegaron a horario. ¿Cómo era posible? Cada agente guardaba en lenguaje natural un registro de sus experiencias, recuperaba los recuerdos pertinentes para decidir qué hacer y elaboraba conclusiones que orientaban sus acciones posteriores. Esa memoria permitía sostener cierta continuidad de conducta, de modo que un modelo podía organizar acciones y ajustarlas a lo que iba ocurriendo. Quedaba por resolver con qué regularidad podía hacerlo sin perder el objetivo ni acumular errores.

La fiabilidad de los agentes mejoró cuando los modelos comenzaron a sostener secuencias más largas de razonamiento, revisar sus respuestas y corregir errores. Una de las técnicas que contribuyeron a ese avance fue el aprendizaje por refuerzo con recompensas verificables: en lugar de premiar una respuesta porque parece convincente, se comprueba si resuelve un problema matemático o si un programa supera determinadas pruebas. El modelo puede así aprender a explorar soluciones y rectificar cuando algo falla. Pero también necesitaba manos. En octubre de 2024, Anthropic presentó una versión de Claude capaz de manejar una computadora mirando capturas de pantalla, calculando dónde mover el cursor y haciendo clic o escribiendo como un usuario. Las primeras demostraciones tuvieron sus momentos de comedia. En una de ellas, Claude hizo clic accidentalmente en el botón que detenía la grabación y perdió todo el registro de lo que estaba haciendo. El fantasma había aprendido a usar la computadora, aunque todavía no sabía muy bien dónde poner los dedos.

¿Pero cómo evaluar esos avances? En 2023 se presentó SWE-bench. Tras reunir problemas reales de proyectos alojados en GitHub, permite comprobar si las modificaciones propuestas por un sistema los resuelven realmente. Cuando Cognition presentó a Devin en marzo de 2024 como “el primer ingeniero de software autónomo”, informó que había resuelto el 13,86 por ciento de los problemas del subconjunto evaluado. El resultado mostraba un avance, pero estaba lejos de justificar tamaño anuncio. La interpretación de estos porcentajes se complicó con el tiempo. En febrero de 2026, una auditoría de OpenAI encontró indicios de que varios modelos habían visto durante su entrenamiento algunos problemas y soluciones de SWE-bench Verified, la versión revisada de la prueba, lo cual les daba una ventaja difícil de separar de su capacidad para resolver problemas. Un porcentaje de aciertos, por sí solo, dice poco sobre el trabajo que podemos delegarle a un sistema. Para evaluar cuánto trabajo podemos delegar en un agente, METR lo enfrenta con problemas concretos de programación, como corregir un error o acelerar el entrenamiento de un modelo, y luego compara con el tiempo que necesitaría una persona especializada para resolverlos. Un “horizonte de dos horas” con un 50 por ciento de éxito significa que el agente resuelve aproximadamente la mitad de los problemas que a un humano le llevarían ese tiempo. Esto es más comprensible que un puntaje abstracto, pero todavía describe un repertorio acotado de trabajos informáticos.

En abril de 2026, en el proyecto MirrorCode, un agente reconstruyó un programa de bioinformática de unas 16.000 líneas, con más de cuarenta comandos para analizar árboles de parentesco evolutivo.

Podía ejecutar el original y observar sus respuestas, pero tenía prohibido consultar su código o buscarlo en internet. Era como reconstruir un mecanismo probando sus botones, sin poder abrirlo. Los investigadores estimaron que ese trabajo le llevaría entre dos y diecisiete semanas a una persona, un intervalo que también muestra lo difícil que es establecer la comparación. Algunos modelos anteriores entregaban versiones incompletas o incluso inventaban urgencias que nadie les había impuesto para dar por terminado el trabajo. Hay modelos que se hartan. El que logró resolverlo siguió hasta superar las pruebas. En otro encargo de reconstruir un lenguaje de programación un agente agotó un presupuesto de mil millones de tokens, unos 550 dólares en la configuración del experimento, sin completar la tarea. Había tomado una mala decisión de diseño al comienzo y siguió obcecadamente intentando remendarla, lo cual cuesta tokens y dinero. En julio de 2026, METR encargó a varios agentes que bajaran el tiempo de entrenamiento de un modelo pequeño. Las mejoras aparecían, pero al repetir las pruebas, algunas desaparecían, ya que los agentes habían seguido trabajando sobre cambios que no aceleraban el entrenamiento. El experimento abre la pregunta por cuánto conviene gastar en nuevos intentos y qué mejora comprobable se obtiene a cambio.

Este escenario nos remite a un intríngulis de la economía llamado el problema del mandante y el mandatario, formalizado por la economía en los setenta. Cuando alguien actúa por cuenta de otro, ya sea el empleado por el patrón, el abogado por el cliente o el gerente por los accionistas, ¿cómo lograr que no traicione el mandato y responda a sus propios intereses? Para eso hacemos contratos, damos incentivos y sobre todo montamos vigilancia. Con un agente de IA no hay de por medio intereses propios, porque no quiere nada ni tiene una agenda propia. El fantasma sin ego no va a traicionarnos por ambición, pero puede cumplir al pie de la letra sin entender lo que queríamos, con toda la fuerza y ningún recaudo. Una alucinación en un chat puede traernos problemas si la damos por buena, pero un agente que ejecuta puede hacernos terminar en una compra equivocada o un archivo borrado.

No estamos exentos de toparnos con un agente *à la* Maxwell Smart, aquel entrañable superagente 86, cuyas buenas intenciones nunca alcanzaban para evitar el desastre. Por eso, delegar exige precisar qué cuenta como un resultado satisfactorio y cómo vamos a comprobarlo. La evaluación tiene que atender al propósito del encargo y las decisiones cuyas consecuencias son difíciles de revertir pueden requerir nuestra aprobación. A esos controles se suma el presupuesto: podemos establecer límites de consumo para que un agente no agote indefinidamente sus recursos persiguiendo una solución. No se trata de impedirle actuar, sino de darle un margen de acción suficientemente amplio para resultar útil y suficientemente definido para que podamos confiar en él.

Los agentes están pasando de resolver encargos puntuales a sostener trabajos durante horas o días, conservar el estado de sus proyectos y coordinarse con otros agentes especializados. El horizonte no se limita al software. Aparecen equipos de agentes capaces de buscar bibliografía, formular hipótesis, ejecutar análisis y contrastar resultados, multiplicando las líneas de investigación que una persona o un laboratorio pequeño pueden explorar. También se perfila una clase de asistente que no necesita que le pidamos cada paso, sino que puede seguir un proceso, esperar una respuesta, advertir que cambiaron las condiciones y retomar el trabajo cuando corresponde, volviendo posibles proyectos que

antes quedaban fuera de nuestro alcance por falta de tiempo o recursos. El fantasma sin ego no ha adquirido voluntad propia, pero sus manos pueden extender enormemente el alcance de las nuestras.

◊

CÓMO CITAR

Marsico, Claudia y Hernán Inverso. 2026. "Agentes de IA: el poltergeist en la máquina". *Mapas del laberinto*, ensayo 21. Phantom Maze — AI & Language Lab. https://doi.org/10.5281/zenodo.24023233

Mapas del laberinto es una serie de Phantom Maze, AI & Language Lab. La colección vive en **phantommaze.cloud**; los ensayos aparecen también en phantommazeailabes.substack.com. La suscripción es gratuita.

Algunas de las ideas que enmarcan este texto se desarrollan en *El fantasma sin ego. Un mapa en el laberinto de la inteligencia artificial*, de Hernán Inverso y Claudia Marsico (Phantom Maze Press, 2026; ISBN 978-987-3729-10-2).

PHANTOM MAZE

Leer este ensayo en línea · Todos los ensayos · El libro

Fuentes. Russell, Stuart J., y Peter Norvig, *Artificial Intelligence: A Modern Approach*, 1.ª ed., Englewood Cliffs, Nueva Jersey, Prentice Hall, 1995, xxviii + 932 pp.; David Silver et al., «Mastering the Game of Go with Deep Neural Networks and Tree Search», *Nature*, vol. 529, n.º 7587, Nature Publishing Group, 2016, pp. 484–489, DOI; Shunyu Yao et al., «ReAct: Synergizing Reasoning and Acting in Language Models», *International Conference on Learning Representations* (ICLR), 2023, pp. 1–33 del documento individual, preprint original de octubre de 2022, arXiv:2210.03629; Timo Schick et al., «Toolformer: Language Models Can Teach Themselves to Use Tools», *Advances in Neural Information Processing Systems*, vol. 36, Neural Information Processing Systems Foundation, 2023, pp. 68539–68551, DOI; Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan y Shunyu Yao, «Reflexion: Language Agents with Verbal Reinforcement Learning», *Advances in Neural Information Processing Systems*, vol. 36, Neural Information Processing Systems Foundation, 2023, pp. 8634–8652, DOI; Anthropic, «Building Effective Agents», 19 de diciembre de 2024, especialmente el apartado «What are agents?», y «Introducing the Model Context Protocol», 25 de noviembre de 2024, ambas publicaciones web sin paginación; Toran Bruce Richards y colaboradores, *AutoGPT*, Significant Gravitas, 2023, software y documentación sin paginación; Yohei Nakajima y colaboradores, *BabyAGI*, 2023, software y documentación sin paginación; Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang y Michael S. Bernstein, «Generative Agents: Interactive Simulacra of Human Behavior», en *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (UIST '23), Nueva York, Association for Computing Machinery, 2023, artículo 2, pp. 1–22, DOI; Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press y Karthik Narasimhan, «SWE-bench: Can Language Models Resolve Real-World GitHub Issues?», *International Conference on Learning Representations* (ICLR), 2024, pp. 1–51 del documento individual, preprint original de octubre de 2023, arXiv:2310.06770; Neil Chowdhury et al., «Introducing SWE-bench Verified», OpenAI, 13 de agosto de 2024, publicación web sin paginación; Scott Wu, «Introducing Devin, the First AI Software Engineer», Cognition, 12 de marzo de 2024, publicación web sin paginación; Thomas Kwa et al., *Measuring AI Ability to Complete Long Tasks*, Model Evaluation & Threat Research (METR), 18 de marzo de 2025, preprint, arXiv:2503.14499v1, 45 pp.; METR, «Time Horizon 1.1», 29 de enero de 2026, publicación web sin paginación; Michael C. Jensen y William H. Meckling, «Theory of the Firm: Managerial Behavior, Agency Costs and Ownership Structure», *Journal of Financial Economics*, vol. 3, n.º 4, Elsevier, octubre de 1976, pp. 305–360, DOI; Gilbert Ryle, *The Concept of Mind*, Londres, Hutchinson's University Library, 1949.