

No revisado por pares ni publicado en revista. Borrador v5 (2026). El contenido puede cambiar. © José Luis D'Andrade · IdentIA / Self-Improving Society. **Todos los derechos reservados.** Cita breve con atribución permitida. **Queda prohibida** la reproducción, traducción, adaptación o reutilización, total o parcial (incluido uso comercial), sin **autorización previa por escrito** del titular.

Las fórmulas del paper de sistemas, explicadas

Self-Improving Society — Paper v5.0 (sistemas). Cada fórmula tal como aparece en el paper, seguida de una explicación para lectores no técnicos y, donde el nombre lo merece, de dónde sale y por qué se llama así.

§3.1 · El objeto de identidad y la medición

$$\Psi(p) = \langle \text{SOUL}(p), M(p), L(p), K(p) \rangle$$
$$m_t(p) = \langle R_t, \theta_t \rangle$$

Estas dos fórmulas responden a una pregunta que en una institución se hace un auditor: **si cambiamos el modelo de inteligencia artificial que usa un agente, ¿sigue siendo el mismo agente?**

La primera dice qué **es** un agente. Su identidad, que se escribe Ψ , está hecha de cuatro cosas que se conservan en el tiempo: su carta fundacional (SOUL: para qué existe y cómo debe comportarse), su memoria (M: lo que ha aprendido), su linaje (L: de dónde viene y qué versiones ha tenido) y sus habilidades (K: lo que sabe hacer y de dónde salió cada habilidad). Es lo mismo que hace que un empleado siga siendo la misma persona aunque cambie de ordenador: su historial, su formación y su puesto no dependen de la máquina.

La segunda dice qué es una **medición** del agente en un momento dado: con qué motor está funcionando ahora (R_t) y qué nivel de habilidad le estamos midiendo hoy (θ_t). Eso es el ordenador y la evaluación de desempeño de este trimestre: cambian, y no por eso cambia la persona.

La consecuencia práctica es la que importa a quien firma: **cambiar de proveedor de modelo no cambia la identidad del agente**, y por tanto no rompe la trazabilidad de lo que hizo antes. Y al revés: si alguien altera la carta fundacional, la memoria o el linaje, ya no es el mismo agente, y el sistema lo trata como tal.

De dónde sale. De ningún sitio que citar: estas dos fórmulas son aportación propia del paper (Definiciones 1 y 2 de §3.1). Los trabajos previos más cercanos miden propiedades de identidad de agentes de modelo de lenguaje o listan la identidad como requisito; ninguno da la identidad como objeto persistente separado de su realización, que es lo que hacen estas dos fórmulas. **Por qué Ψ (psi).** Es la letra griega que la física usa para el estado de un sistema. El paper la toma prestada con ese único sentido, «el estado que persiste», y advierte expresamente que no se apoya en ninguna otra lectura de la física: es notación, no analogía.

§5.1 · Variación y selección

$$\text{RSI} = \underbrace{\text{Variation}}_{\text{exists (runtime)}} \circ \underbrace{\text{Selection}}_{\text{this ladder}}$$

Esta fórmula dice de qué está hecha la **automejora** (RSI, por *recursive self-improvement*): de dos operaciones distintas, encadenadas.

La primera es la **variación**: producir cambios. Los agentes ya lo hacen por sí solos: escriben nuevas habilidades, corrigen las que tienen, resumen lo aprendido. Es como una plantilla que cada semana propone maneras nuevas de hacer el trabajo. Eso ya existe y no hubo que construirlo; lo trae el motor de agentes.

La segunda es la **selección**: decidir cuáles de esos cambios son realmente mejores y merecen quedarse, y cuáles se descartan. Eso es lo que falta en casi todos los sistemas, y es lo que construye este proyecto: la «escalera» de la que hablan las secciones siguientes.

El símbolo del medio, el círculo, significa «una después de la otra»: primero se varía, luego se selecciona, y el resultado vuelve a entrar en el ciclo. La frase que acompaña a la fórmula en el paper es la clave: **variación sin selección es deriva** (el sistema cambia, pero no se sabe si a mejor), y **selección sin variación no tiene nada que elegir**.

Para quien decide, la lectura es esta: un sistema que «aprende solo» sin un mecanismo explícito de selección no está mejorando, está cambiando. La diferencia entre las dos cosas es quién juzga el cambio, y con qué evidencia. Todo lo que viene después de esta fórmula es la respuesta a esa pregunta.

De dónde sale. De Darwin. El paper lo dice sin rodeos: la mejora nunca la declara el que cambió; la selecciona una comparación medida contra alternativas, en un entorno. Variación y selección son los dos motores de *El origen de las especies* (1859), y la fórmula no hace más que escribirlos uno junto al otro.

§5.4 · F0 — Reputación Beta–Bernoulli

$$\pi \sim \text{Beta}(\alpha, \beta), \quad \alpha = s + 1, \quad \beta = (n - s) + 1, \quad \mathbb{E}[\pi] = \frac{\alpha}{\alpha + \beta}$$
$$\hat{p}_{\text{LB}} = \frac{\hat{p} + \frac{z^2}{2n} - z \sqrt{\frac{\hat{p}(1 - \hat{p})}{n} + \frac{z^2}{4n^2}}}{1 + \frac{z^2}{n}}, \quad \hat{p} = \frac{s}{n}$$

Estas dos fórmulas resuelven un problema que cualquier jefe de departamento reconoce: **¿cómo comparas a alguien que lleva una semana con alguien que lleva un año?**

Un agente nuevo ha hecho una tarea y le salió bien: 1 de 1, el 100 %. Otro lleva cien tareas y acertó noventa: el 90 %. Si ordenas por porcentaje, el novato queda primero. Nadie con criterio haría eso, porque un solo acierto no dice casi nada. El problema es que un ordenador, si no se le indica lo contrario, sí lo hace.

La primera fórmula corrige el error de raíz: en lugar de guardar un porcentaje, guarda **una distribución**, es decir, el porcentaje y cuánta confianza se tiene en él. Con s aciertos en n intentos, la fórmula Beta produce una curva: estrecha y afilada para el veterano (sabemos bien cuánto vale), ancha y plana para el novato (podría valer casi cualquier cosa). La esperanza, $E[\pi]$, es el valor central de esa curva.

La segunda fórmula, la de Wilson, es la regla de decisión: **no se ordena por el valor central, sino por el peor valor plausible**. Para el veterano, ese peor valor es alto, porque hay mucha evidencia; para el novato, es bajo, porque hay poca. Así el 90/100 queda por encima del 1/1, que es lo que dictaría el sentido común.

Lo que esto compra para la institución es prudencia automática: **nadie sube de rango por un golpe de suerte**, y un agente recién creado tiene que demostrar su valor antes de que el sistema le confíe trabajo importante. Es el primer peldaño de la escalera y el único que ya funciona sin más datos que un recuento de aciertos.

Por qué se llama «reputación Beta–Bernoulli», y de dónde sale. El nombre junta dos apellidos de la estadística clásica. **Bernoulli** (Jacob Bernoulli, siglo XVII) da nombre al experimento más simple que existe: algo que sale bien o mal, sin término medio, como lanzar una moneda o resolver una tarea. Cada tarea que hace un agente es un «ensayo de Bernoulli»: acierto o fallo. **Beta** es la distribución que los estadísticos usan desde hace dos siglos para responder a la pregunta natural que sigue: *después de ver s aciertos en n intentos, ¿cuál es la probabilidad de acierto de esta moneda, y con cuánta seguridad lo sé?* Las dos encajan de forma exacta: si empiezas sin saber nada (una Beta plana) y vas sumando aciertos y fallos, la curva se va afilando alrededor de la tasa real, y las fórmulas $\alpha = s + 1$ y $\beta = (n - s) + 1$ son literalmente «los aciertos más uno, los fallos más uno». Es el mismo mecanismo que usan los sistemas de valoración de vendedores en los mercados en línea, y por la misma razón: un vendedor con una sola reseña de cinco estrellas no puede quedar por encima de uno con mil reseñas y media de 4,8. Llamarlo **reputación** en el paper es deliberado: lo que se está calculando no es una nota, es cuánto merece confianza un agente a la vista de su historial, con la incertidumbre incluida en el propio número. La **cota de Wilson** lleva el nombre de Edwin Wilson, que la publicó en 1927; es la corrección estándar para no fiarse de porcentajes calculados sobre pocos casos.

§5.5 · F1 — Bradley–Terry y TrueSkill: comparar, no puntuar

$$P(i \succ j) = \frac{\gamma_i}{\gamma_i + \gamma_j} = \sigma(\beta_i - \beta_j)$$

$$\gamma_i \leftarrow \frac{\sum_{j \neq i} w_{ij}}{\sum_{j \neq i} \frac{n_{ij}}{\gamma_i + \gamma_j}}$$

$$\mu_i \leftarrow \mu_i + \frac{\sigma_i^2}{c} v\left(\frac{\mu_i - \mu_j}{c}\right), \quad \sigma_i^2 \leftarrow \sigma_i^2 \left[1 - \frac{\sigma_i^2}{c^2} w\left(\frac{\mu_i - \mu_j}{c}\right)\right]$$

El peldaño anterior mide a cada agente por separado: cuántas tareas acertó. Este peldaño hace algo distinto y más exigente: **los compara entre sí**. La razón es que «acertó» depende de la tarea: un agente que acierta tareas fáciles no es mejor que uno que acierta tareas difíciles. Comparar dos agentes sobre la misma tarea elimina ese sesgo, igual que en un torneo no importa cuántos puntos hiciste en total sino a quién ganaste.

La primera fórmula (Bradley–Terry) dice cómo se convierte una lista de enfrentamientos en un ranking. Cada agente tiene una «fuerza» y_i , y la probabilidad de que i venza a j es la fuerza de i dividida entre la suma de las dos. Es exactamente el modelo que usa la clasificación pública de modelos de lenguaje Chatbot Arena, y el mismo principio que el Elo del ajedrez: ganar a un fuerte vale más que ganar a un débil.

La segunda fórmula es el procedimiento para calcular esas fuerzas a partir de los resultados. Se llama iteración MM y tiene una virtud práctica que el paper subraya: se puede escribir en unas pocas líneas sin ninguna librería externa. Eso importa a quien audita: la clasificación se puede recalcular a mano y verificar, no viene de una caja negra.

La tercera (TrueSkill) añade lo que a Bradley–Terry le falta: la **incertidumbre**. Cada agente tiene una media μ (cuánto vale) y una desviación σ (cuánto seguro se está de ello), y cada resultado mueve las dos: la media hacia arriba o abajo, y la incertidumbre siempre hacia abajo, porque cada partida enseña algo.

Lo que gana la institución es un ranking que resiste la trampa más habitual: **elegir tareas fáciles para parecer bueno**. Aquí solo cuenta a quién venciste.

Por qué se llaman así, y de dónde salen. Bradley–Terry lleva los apellidos de dos estadísticos, Ralph Bradley y Milton Terry, que lo publicaron en 1952 en la revista *Biometrika* para un problema muy concreto: cómo ordenar productos cuando lo único que se tiene son comparaciones por pares, del tipo «el catador prefirió A antes que B». Su idea fue asignar a cada cosa un número de fuerza y decir que la probabilidad de que A gane a B es la fuerza de A dividida entre la suma de las dos. Setenta años después sigue siendo el modelo estándar para convertir enfrentamientos en un ranking: es lo que hay detrás del Elo del ajedrez, de la clasificación Chatbot Arena y de buena parte de los sistemas que aprenden de preferencias humanas. La segunda fórmula, la iteración «MM», es el método que David Hunter propuso en 2004 para calcular esas fuerzas de forma sencilla y estable; el proyecto lo eligió porque cabe en unas pocas líneas de código y se puede recalcular a mano.

TrueSkill no es un apellido sino un nombre comercial: lo crearon en 2006 Ralf Herbrich, Tom Minka y Thore Graepel en Microsoft Research para el emparejamiento de jugadores en Xbox Live. El problema que resolvía era práctico: Elo funciona bien con muchas partidas, pero un jugador nuevo necesita docenas para que su puntuación sea fiable, y mientras tanto lo empareja mal. TrueSkill añadió a la puntuación una **medida de incertidumbre** que empieza grande y se reduce con cada partida, y extendió el modelo a empates y a equipos. El nombre («habilidad verdadera») alude a eso: la puntuación visible es una estimación de la habilidad real, y el sistema sabe cuánto le falta para acertarla. El paper lo usa por la misma razón que Xbox: **no fiarse de un recién llegado hasta que la incertidumbre haya bajado**.

§5.6 · F2 — Teoría de Respuesta al Ítem: la compuerta de fiabilidad

$$P(\text{profile } p \text{ solves task } k) = \sigma(a_k(\theta_p - b_k))$$

Los dos peldaños anteriores tienen una grieta común: miran al agente, no a la tarea. Este peldaño la cierra. La fórmula dice que la probabilidad de que un agente resuelva una tarea depende de **dos cosas a la vez**: la habilidad del agente (θ) y la dificultad de la tarea (b), y de un tercer número (a) que mide cuánto distingue esa

tarea entre buenos y malos. Si la tarea es tan fácil que la acierta cualquiera, no dice nada del agente; si es tan difícil que nadie la resuelve, tampoco.

La consecuencia es la que da nombre a la sección: es la **compuerta de fiabilidad**. Con este modelo, un agente **no puede subir de rango eligiendo tareas fáciles**, porque la dificultad de cada tarea se estima al mismo tiempo que la habilidad de cada agente, y acertar lo fácil vale poco. Y el sistema puede decir además si sus propias estimaciones son estables: si la habilidad de un agente cambia mucho de una estimación a la siguiente, la señal no es fiable, y **el sistema se niega a actuar sobre ella**. Versiones anteriores de esta nota añadían que eso es exactamente lo que ocurrió en la única ejecución en producción que el paper reporta —la compuerta no convergió y todo lo que está por encima quedó bloqueado—. **Esa lectura queda retirada** (§7.3 del paper de sistemas): la prueba de convergencia medía el cambio por iteración antes del paso que centra las dificultades de los ítems, así que la bandera no podía reportar éxito con ningún dato. La regla sigue en el código; ninguna ejecución en vivo se ha observado aplicándola.

Por qué se llama así, y de dónde sale. La Teoría de Respuesta al Ítem nació en los exámenes estandarizados. Su problema original era el mismo que el nuestro: dos estudiantes que responden preguntas distintas no se pueden comparar por número de aciertos, porque las preguntas no valen lo mismo. La solución fue modelar cada pregunta («ítem») con su propia dificultad y estimar la habilidad de cada persona a la vez que la dificultad de cada pregunta. La formuló Frederic Lord en Estados Unidos a partir de los años cincuenta, en paralelo con el danés Georg Rasch (1960); el libro de Lord de 1980, *Applications of Item Response Theory to Practical Testing Problems*, es el que cita el paper. El modelo concreto de la fórmula se llama «de dos parámetros» (2PL): dificultad y discriminación por ítem. Es la matemática que hay detrás de pruebas como el SAT, el GRE o PISA, y el paper la usa para lo mismo: **medir a los agentes como se mide a los examinados, descontando la pregunta**.

§5.7 · La dependencia que cierra la escalera

$$\text{enable}(F5) \iff \text{Done}(F2) \wedge \text{Done}(F3) \wedge \text{Done}(F4)$$

Esta no es una fórmula de cálculo sino una **regla**, escrita en lógica para que no admita interpretación. Dice: el sistema solo puede activar la automejora recursiva (F5, el último peldaño, donde el sistema se edita a sí mismo) si, y solo si, se cumplen tres condiciones a la vez: que la compuerta de fiabilidad haya convergido (F2), que actuar sobre la medición sea estable y reversible (F3) y que la diversidad de agentes esté protegida (F4).

El paper insiste en un detalle que cambia todo: «hecho» (*Done*) no significa «el código existe», sino «**la condición se ha medido en vivo y se cumple**». Versiones anteriores decían que por eso hoy F5 está bloqueado: no por una decisión de política, sino porque la medición de F2 dio *converged = False*. **Eso queda retirado** (§7.3 del paper de sistemas): la bandera no podía valer `True` bajo el criterio entonces en uso, y tras el arreglo un ajuste en un entorno de desarrollo da *converged = True*. Hoy F5 y F3 están apagados **por decisión**. Lo que cualquiera puede seguir comprobando leyendo el dato es la regla misma, no una aplicación en vivo de ella.

De dónde sale. El paper lo emparenta con el segundo teorema de incompletitud de Gödel (1931): un sistema suficientemente expresivo no puede demostrar su propia consistencia desde dentro. No afirma que el teorema se aplique literalmente; adopta su moraleja de ingeniería: las garantías que autorizan a un sistema a modificarse deben estar **fuera** del bucle que certifican. Un sistema que califica su propia mejora con la misma señal que está optimizando no tiene suelo donde apoyarse.

Apéndice D.1 · El análogo humano y el término Ω

$$\hat{\Psi}_{\mathcal{M}}(p^H) = \langle S_{\mathcal{M}}(p^H), M_{\mathcal{M}}(p^H), L_{\mathcal{M}}(p^H), K_{\mathcal{M}}(p^H) \rangle, \quad p^H \not\equiv \hat{\Psi}_{\mathcal{M}}(p^H)$$

Esta fórmula es una advertencia escrita en símbolos. Toma la misma tupla de identidad de §3.1 y la aplica a una **persona** (p con superíndice H, de *humano*), vista a través de un modelo M cualquiera: lo que ese modelo capta de su carácter, su memoria, su historia y sus habilidades. Y a continuación afirma, con el signo «no equivale», lo que cualquiera sabe pero los sistemas olvidan: **lo que un modelo captura de una persona nunca es la persona**. Siempre queda un residuo que el modelo no ve, y ese residuo se llama Ω .

La razón de escribirlo es de diseño, no de filosofía. Si Ω formara parte de la identidad Ψ , nadie podría verificar si una identidad se ha conservado, porque una de sus piezas sería, por definición, inobservable. Por eso el paper mantiene Ω **fuera** del objeto de identidad: la identidad está hecha solo de lo que se puede guardar, inspeccionar y hacer cumplir. Lo que no se puede, no se promete.

Por qué Ω (omega). Es la última letra del alfabeto griego, y en muchas disciplinas se usa para «el resto», «el límite» o «lo que queda fuera». El paper la elige con ese sentido: es el término de incompletitud del propio formalismo, la parte que el modelo admite no modelar.

Apéndice D.2 · El mapa de comportamiento

$$b_t(p) = f(\Psi(p), m_t(p)) + \varepsilon_t, \quad \varepsilon_t \sim \Omega(p)$$

La última fórmula une las anteriores. Dice de qué depende **la conducta observada** de un agente en un momento dado (b_t): de su identidad (Ψ , lo que persiste), de la medición del momento (m_t : el motor que lo ejecuta y la habilidad que se le estima), y de un ruido (ε) que no se modela y que procede de Ω .

Dicho de otra forma: dos ejecuciones del mismo agente, con la misma identidad y el mismo motor, pueden comportarse de forma distinta, y la fórmula lo reconoce en vez de ocultarlo. Lo importante es **dónde** vive ese ruido: en el comportamiento, no en la identidad. La identidad sigue siendo verificable; lo que se acota es el efecto del ruido, y eso es trabajo de la gobernanza, no de la ontología.

Para quien decide, esto responde a una objeción habitual: «si el agente no se comporta siempre igual, ¿cómo pueden decir que es el mismo?». La respuesta del paper es que **ser el mismo y comportarse igual son cosas distintas**, igual que en las personas, y que el sistema solo promete la primera, que es la que se puede auditar.