mdt.

Ex Datis, Claritas

La IA que no sabe escribir: qué es Jev y cómo funciona

Jev no genera texto: toma decisiones y devuelve su probabilidad. Qué es, cómo funciona y dónde encaja en marketing y analítica de clientes.

Manuel Delgado Tenorio · 23 de septiembre de 2026

Publicado originalmente en www.manueldelgado.com/blog/jev-ia-que-no-escribe



Llevamos casi cuatro años asociando la inteligencia artificial con un chat: le escribes algo y te contesta. Tanto que, para mucha gente, "IA" y "*chatbot*" son ya sinónimos.

El pasado 15 de septiembre, TypeSafe AI, una empresa de San Francisco fundada en 2024 por Diogo Almeida (que pasó cuatro años en OpenAI trabajando en las técnicas de entrenamiento de ChatGPT), salió a la luz con 40 millones de dólares de financiación y un modelo, Jev, que no genera ni una frase. En sus primeras 24 horas, casi el 13% de los equipos de pago de Vercel ya lo estaba usando a través de su AI Gateway, más del doble que cualquier lanzamiento anterior (incluida la familia GPT-5.6). En una semana ha pasado de ser un desconocido a ser el tema de conversación de quienes construyen software.

Lo curioso es que, a diferencia de casi todos los lanzamientos de este año, Jev no promete ser más listo que nadie. **Lo que trae es otra forma de poner a trabajar la inteligencia de un modelo de lenguaje: en vez de pedirle que escriba, le pides que decida.** Y eso, para quienes trabajamos con datos de clientes, tiene mucho más recorrido de lo que parece.

Del clasificador de toda la vida al clasificador que generaliza

Cualquier negocio con clientes toma miles de pequeñas decisiones cada día. ¿Este cliente se va a ir? ¿Este *lead* merece una llamada? ¿De qué se queja esta reseña? ¿Qué oferta le enseño a esta persona? **Durante años, la forma de automatizar esas decisiones ha sido el clasificador**: un modelo entrenado con ejemplos que recibe unos datos y devuelve una probabilidad. Los modelos de propensión a la baja o de *scoring* de *leads* son exactamente eso, y a lo largo de los años he construido unos cuantos.

Los clasificadores tienen virtudes que se suelen infravalorar: son rápidos, baratos y devuelven un número con el que se puede operar (si la probabilidad de baja supera el 70%, llama el equipo de retención). Pero tienen dos límites serios. El primero, que cada uno sabe hacer una sola cosa: **el modelo que predice bajas no sabe nada de reseñas, y para cada decisión nueva hace falta recopilar datos etiquetados y entrenar otro modelo.** El segundo, que se llevan mal con el texto libre. Un modelo de propensión maneja muy bien la antigüedad, el gasto o el número de incidencias, pero entiende mucho peor el correo enfadado que el cliente acaba de mandar.

Los grandes modelos de lenguaje resolvieron justo eso. ChatGPT, Claude o Gemini pueden leer ese correo y decirte si el cliente se quiere ir, por qué y con qué urgencia, sin entrenamiento específico. El problema es que lo hacen escribiendo, palabra a palabra, y eso tiene un coste. **Usar un LLM para tomar millones de decisiones pequeñas es lento y caro**, y la seguridad que dice tener en su respuesta no es una probabilidad en la que se pueda confiar para fijar umbrales.

Jev se coloca en medio: **entiende el lenguaje como un LLM, pero responde como un clasificador**, con una decisión y su probabilidad. TypeSafe llama a esta nueva categoría "*System One models*", aunque a mí me parece más clara la etiqueta que proponen Maggie Appleton y Simon Willison: modelos de decisión.

Conviene, eso sí, poner a Jev en su sitio. Si ordenas los modelos de IA por lo que sale de ellos, y no por lo que entra, el panorama es mucho más amplio y más asentado de lo que sugiere tanto chat: modelos de voz que oyen y hablan sin pasar

por el texto, como Gemini 3.8 Live; modelos de difusión que generan imagen y vídeo, como Veo o Sora; modelos de visión, lenguaje y acción que convierten lo que ve una cámara en órdenes para los motores de un robot, como Gemini Robotics, π0 o Helix; modelos del mundo que predicen qué va a pasar en un entorno físico, como Genie 3 o Cosmos; AlphaFold, que predice la estructura de las proteínas y le valió a sus creadores el Nobel de Química de 2024; el AIFS del Centro Europeo de Previsiones Meteorológicas a Plazo Medio, en producción desde febrero de 2025 y que este año ha dejado de ejecutar los modelos externos de Google, Microsoft, Huawei y Nvidia que corría en paralelo; o los modelos fundacionales de series temporales, como TimesFM o Chronos. **En todos ellos, lo que sale no es una frase, sino un número, una imagen o una acción.** Jev es el último en llegar a esa familia, y el que más la acerca al día a día del marketing.

Cómo funciona: un estado, unas preguntas y unas probabilidades

A Jev no se le escribe un *prompt* ni se charla con él. Según su documentación, cada llamada tiene dos piezas. La primera es el **estado** (`state`), que es el material sobre el que hay que decidir: puede ser un texto suelto, una lista de mensajes o, lo que recomiendan para la mayoría de los casos, un objeto con campos con nombre, lo que permite mezclar el texto de un cliente con sus datos. La segunda es un conjunto de **preguntas cerradas**, cada una con su nombre, su tipo, sus instrucciones y sus criterios.

Hay tres tipos de pregunta:

- **Noul** (sí o no). Devuelve la probabilidad de que la respuesta sea sí, un número entre 0 y 1. Ojo con leerlo mal: un noul de 0,9 no significa que algo sea "muy" cierto, sino que hay un 90% de probabilidades de que lo sea.
- **Choice** (elección). Elige entre un conjunto de hasta 255 opciones, cada una descrita con un criterio, y devuelve la probabilidad de cada opción.
- **Score** (puntuación). Sitúa algo en una escala de entre 2 y 10 niveles que tú defines, y devuelve la probabilidad de cada nivel y una puntuación que es la media ponderada de todos ellos.

Las respuestas de elección y de puntuación incluyen además un campo de confianza, que resume lo concentradas que están las probabilidades: si casi todo el peso cae en una opción, la confianza es alta; si se reparte entre varias, es baja. **Esa probabilidad no es un adorno: es lo que permite decidir qué hace el sistema solo y qué pasa a una persona.** TypeSafe entrena el modelo con una técnica propia, que llama RLCD (aprendizaje por refuerzo para decisiones calibradas), precisamente para que esas probabilidades sean honestas.

Dos detalles técnicos más explican su velocidad y su precio. Jev evalúa todas las preguntas de una llamada en paralelo, así que hacerle diez preguntas sobre el mismo estado cuesta prácticamente lo mismo, en tiempo, que hacerle una. Y como no escribe, no hay salida que cobrar: **la entrada cuesta 0,042 dólares por millón de *tokens* y la salida es gratis**, según la ficha del modelo, que también fija un contexto de 64.000 *tokens* por llamada y solo admite texto (nada de imágenes, audio o vídeo, de momento).

Un caso completo: el cliente que se quiere ir

Veámoslo con un ejemplo. Una operadora de fibra y móvil recibe este mensaje de un cliente por el formulario de contacto: "*Es la tercera vez este mes que me quedo sin internet y nadie me da una solución. He visto que otra compañía ofrece la misma velocidad por diez euros menos. Si esta semana no está arreglado, pido la portabilidad*".

En una sola llamada, el agente de IA que procesa los formularios de contacto le pasa a Jev el mensaje y algunos datos del cliente, y le hace tres preguntas, una de cada tipo: ¿quiere irse?, ¿por qué?, ¿con qué firmeza?

```
{
  "model": "jev-latest",
  "state": {
    "mensaje": "Es la tercera vez este mes que me quedo sin internet y nadie me da
una solución. He visto que otra compañía ofrece la misma velocidad por diez euros
menos. Si esta semana no está arreglado, pido la portabilidad.",
    "cliente": {
      "antiguedad_meses": 38,
      "tarifa": "Fibra 1 Gb + 2 líneas móviles",
      "incidencias_90_dias": 3,
      "permanencia": "sin permanencia"
    }
  },
  "questions": {
    "riesgo_baja": {
      "type": "noul",
      "instructions": "¿Expresa el cliente intención de darse de baja o de irse a
otra compañía?",
      "criteria": {
        "true": "Menciona la baja, la portabilidad o el cambio de compañía, aunque
sea de forma condicional.",
        "false": "Se queja o pide ayuda, pero no plantea irse."
      }
    },
    "motivo_principal": {
      "type": "choice",
      "instructions": "¿Cuál es el motivo principal del descontento del cliente?",
      "criteria": {
        "averias": "Cortes, fallos técnicos o calidad de la conexión.",
        "atencion": "Trato recibido o falta de respuesta del servicio de atención
al cliente.",
        "precio": "Precio de la tarifa o comparación con ofertas de
otras compañías.",
        "necesidades": "Cambio en sus circunstancias o en lo que necesita
del servicio.",
        "otro": "Cualquier otro motivo."
      }
    },
    "firmeza": {
      "type": "score",
      "instructions": "¿Con qué firmeza expresa el cliente su intención de irse?",
      "criteria": [
        "No menciona irse.",
        "Lo insinúa de pasada.",
        "Lo plantea como una posibilidad.",
        "Lo plantea como un ultimátum con condiciones.",
        "Ya ha decidido irse o ha iniciado el trámite."
```

```
      ]
    }
  }
}
```

Y esta sería la respuesta, en unas décimas de segundo (es una respuesta simulada, a efectos ilustrativos, pero no es muy distinta de lo que devolvería Jev en la realidad):

```
{
  "model": "jev-1.13.0",
  "answers": {
    "riesgo_baja": {
      "type": "noul",
      "noul": 0.94
    },
    "motivo_principal": {
      "type": "choice",
      "choice": "averias",
      "confidence": 0.48,
      "probabilities": {
        "averias": 0.58,
        "atencion": 0.22,
        "precio": 0.17,
        "necesidades": 0.01,
        "otro": 0.02
      }
    },
    "firmeza": {
      "type": "score",
      "score": 2.93,
      "confidence": 0.64,
      "legend": {
        "0": "No menciona irse.",
        "1": "Lo insinúa de pasada.",
        "2": "Lo plantea como una posibilidad.",
        "3": "Lo plantea como un ultimátum con condiciones.",
        "4": "Ya ha decidido irse o ha iniciado el trámite."
      },
      "probabilities": {
        "0": 0.0,
        "1": 0.02,
        "2": 0.15,
        "3": 0.71,
        "4": 0.12
      }
    }
  },
  "usage": {
    "input_tokens": 512,
    "output_tokens": 61
  }
}
```

Leamos el resultado como lo leería un equipo de retención. El riesgo de baja es muy alto (un 94%) y la firmeza está en 2,93, prácticamente en el nivel del ultimátum (la escala va de 0 a 4 y la puntuación es la media de los niveles ponderada por su probabilidad). Hasta aquí, nada que no dijera cualquier persona que leyera el

mdt.

mensaje. Lo interesante es el motivo. Jev se inclina por las averías, pero con una confianza de solo 0,48, porque reparte buena parte de la probabilidad entre la atención al cliente (22%) y el precio (17%). **El reflejo habitual ante un cliente que amenaza con irse es ofrecerle un descuento, y aquí sería probablemente la respuesta equivocada**: lo que este cliente pide es que le arreglen la conexión y que alguien le haga caso.

Con esos tres números, el sistema puede actuar sin que nadie lea el mensaje: si el riesgo de baja supera, por ejemplo, 0,8 y la firmeza pasa de 2,5, el caso va hoy mismo al equipo de retención, con una visita técnica prioritaria como primera propuesta. Y cuando la confianza en el motivo es baja, como aquí, una persona lee el mensaje antes de decidir la oferta. **Es lo mismo que llevamos años haciendo con los modelos de propensión (fijar umbrales sobre una probabilidad), pero ahora aplicado a texto libre y sin entrenar nada.** La propia documentación recomienda, con buen criterio, que el umbral dependa del riesgo de cada acción: no hace falta la misma seguridad para etiquetar un mensaje que para ofrecer un descuento de 120 euros.

Rápido y barato, pero ¿cuánto?

Las cifras de TypeSafe son espectaculares. En su anuncio habla de latencias de entre 70 y 500 milisegundos, frente a los entre 3 y 329 segundos de los modelos de frontera, y en sus propias evaluaciones con flujos de trabajo reales llega a hablar de un modelo 193,6 veces más rápido y 444,6 veces más barato, aunque reconoce que es el extremo alto. Las pruebas independientes las reducen mucho, no obstante. La más completa que he visto hasta ahora, publicada por AY Automate con 791 decisiones etiquetadas, deja esta foto:

	Lo que dice TypeSafe	Lo que mide la prueba independiente
Velocidad	Entre 70 y 500 ms por llamada; hasta 193,6 veces más rápido	Mediana de 0,33 s; entre 2 y 3,6 veces más rápido que los modelos comparados
Coste	Hasta 444,6 veces más barato	Entre 4,7 y 40,6 veces más barato por cada mil decisiones
Precisión	La misma inteligencia que un modelo de frontera	78,8% frente al 84,0% de GPT-5.6 Terra al clasificar entre 77 categorías; a la par de los modelos pequeños

La conclusión es clara: **Jev es mucho más rápido y mucho más barato, pero no es más listo, y en las tareas difíciles eso se nota.** En la prueba de clasificar mensajes de clientes de un banco entre 77 categorías (el conjunto de datos público Banking77), acierta lo mismo que modelos pequeños como GPT-5.4 nano o Gemini 3.5 Flash-Lite, y cinco puntos menos que el modelo grande de OpenAI. Eso sí, a un coste de alrededor de un céntimo y medio de dólar por cada mil decisiones.

La velocidad, además, no es solo cuestión de comodidad. Ably Labs lo mostró con una demostración muy gráfica: puso a varios modelos a jugar al Pong. En 12 segundos, Jev tomó 47 decisiones sobre hacia dónde mover la pala; Gemini, Claude y GPT, dos o tres cada uno. Los modelos de lenguaje acertaban casi siempre el movimiento, pero cuando terminaban de contestar la pelota ya había pasado. **Hay decisiones que, si no se toman a tiempo, da igual lo bien que se tomen**, y en marketing hay muchas: qué recomendación mostrar mientras se carga una página, qué oferta lanzar mientras el cliente sigue en la aplicación o a quién enrutar una llamada entrante en tu *contact centre*.

El grande dirige, el pequeño decide

Nada de esto significa que Jev vaya a sustituir a ChatGPT o a Claude. No sabe escribir, no sabe razonar en voz alta y no sabe explicarte nada. **Su sitio natural es ser un paso dentro de un proceso más amplio**, muchas veces dirigido por un modelo convencional: un agente que atiende a un cliente o que procesa un expediente y que, cada vez que necesita una decisión rápida (¿esto es urgente?, ¿este documento es el que busco?, ¿esta respuesta que acabo de redactar se sale de las normas de la empresa?), se la pregunta a Jev y sigue adelante.

La prueba de AY Automate cuantifica la versión más sencilla de este patrón, la cascada: Jev resuelve las decisiones en las que su confianza es de al menos 0,8 y pasa las demás a GPT-5.6 Terra. Solo hubo que escalar entre el 19% y el 23% de los casos, y **la cascada acertó lo mismo que el modelo grande en solitario, por poco más de la cuarta parte del coste y en la mitad de tiempo.** Es el tipo de resultado que convierte una buena idea en una decisión de arquitectura, y TypeSafe lo recoge como patrón en su documentación.

Willison apunta otro uso muy práctico: reordenar resultados de búsqueda. Un algoritmo barato y clásico selecciona los cien documentos candidatos, y Jev puntúa cada uno según su relevancia para la consulta. Cambia "documentos" por "productos" y "consulta" por "lo que este cliente ha estado mirando", y tienes un sistema de recomendación.

De Kahneman a Jevons

Los nombres que ha elegido TypeSafe dicen mucho de su apuesta, y a mí me tocan de cerca por mi investigación sobre economía conductual. "*System One*" viene de Daniel Kahneman y su *Pensar rápido, pensar despacio*: el sistema 1 es el pensamiento rápido e intuitivo, el que te hace frenar cuando un balón cruza la calle; el sistema 2, el lento y deliberado, el que usas para hacer la declaración de la renta. **Los modelos de lenguaje que razonan serían el sistema 2 de la IA; Jev quiere ser el sistema 1.**

Y "Jev" viene de William Stanley Jevons, el economista inglés que en el siglo XIX observó que, cuando las máquinas de vapor se hicieron más eficientes, el consumo de carbón no bajó, sino que se disparó. La apuesta de TypeSafe es que con la inteligencia pasará lo mismo: **si una decisión cuesta una fracción de céntimo, se tomarán miles de millones de decisiones que hoy nadie se plantea tomar.** Es el mismo razonamiento que hay detrás de su pregunta de presentación: "los modelos llevan años siendo sobrehumanos en el chat, así que ¿dónde está toda la automatización?".

Las pegas

Las hay, y son serias. La primera es la opacidad. Un LLM ya es una caja negra, pero al menos puedes pedirle que justifique su respuesta (aunque su justificación no siempre sea fiable). **Con Jev, metas el texto que metas, lo único que sale es un número**, y si marca a un cliente como de alto riesgo no hay forma de saber qué frase lo delató. Willison lo ilustró con un experimento sencillo: le preguntó a Jev si cada ciudad de la bahía de San Francisco era una "buena ciudad", y el modelo puso a Cupertino, la de Apple, la primera y a East Palo Alto la última, sin más explicación.

Su conclusión, que comparto: **que nadie lo use para ordenar candidatos a un puesto de trabajo.**

En analítica de clientes, esta advertencia es especialmente pertinente. En el ejemplo anterior he dejado fuera, a propósito, lo que paga el cliente. **Si metes su valor en el estado, el modelo puede acabar decidiendo a quién se retiene y a quién no sin que nadie sepa qué peso ha tenido ese dato**, y un sistema que decide en silencio qué cliente merece atención es exactamente el tipo de sistema que hay que vigilar. La buena noticia es que, precisamente porque Jev es tan barato, someterlo a cientos de pruebas controladas para destapar esos sesgos cuesta unos céntimos. No hay excusa para no hacerlo.

La segunda pega es la calibración. Todo el valor de Jev descansa en que sus probabilidades sean honestas, y, que yo sepa, esa afirmación todavía no la ha contrastado nadie ajeno a TypeSafe de forma sistemática. La tercera, más prosaica, es el idioma: la propia ficha del modelo reconoce que está optimizado para el inglés y que en otros idiomas acierta menos. **Si vas a usarlo con mensajes de clientes en español, pruébalo antes con tus propios datos.**

Y la cuarta es la crítica que más se ha repetido entre programadores: que, en el fondo, Jev "no es más que un clasificador" con un entrenamiento nuevo. Es verdad, pero no creo que eso sea un problema sino, al contrario, su mayor virtud. Jev es, además, una primera versión, de una empresa pequeña, que en su primera semana ya está a pocos puntos del mejor modelo del mundo en una tarea concreta. El primer ChatGPT tampoco era gran cosa comparado con lo que usamos hoy.

Dónde encaja en marketing y analítica de clientes

Casi en cualquier punto en el que hoy haya una persona leyendo textos para clasificarlos, o una decisión que no se toma porque sería demasiado caro tomarla caso a caso. Algunos ejemplos:

- **Clasificar opiniones y reseñas** por tema, gravedad y sentimiento, en el momento en que llegan (choice y score).
- **Cualificar *leads*** a partir del texto del formulario, la conversación con el comercial o la web de la empresa que pregunta (score).
- **Detectar oportunidades de venta cruzada** en las conversaciones de atención al cliente: ¿menciona este cliente una necesidad que cubre otro producto? (noul).
- **Decidir la siguiente mejor acción** para cada cliente entre un catálogo de campañas, ofertas o mensajes (choice).
- **Codificar las respuestas abiertas de las encuestas**, esa tarea que casi todo el mundo deja para el final y que tantas veces se queda sin hacer (choice).
- **Revisar que los mensajes de una campaña** cumplen las normas de marca y las legales antes de enviarse (noul).

Ninguna de estas tareas es nueva. Lo nuevo es que **ya no hace falta un proyecto de ciencia de datos para cada una**: basta con describir bien la pregunta y los criterios, y medir con cuidado los resultados.

Software que consume inteligencia

Almeida lo resumió en una frase que me parece lo más interesante de todo este lanzamiento: "las personas no pueden ser las únicas consumidoras de inteligencia".

Durante estos años hemos diseñado la IA para hablar con personas. **Jev es una de las primeras piezas diseñadas para hablar con software**, y el software necesita otra cosa: respuestas rápidas, baratas, en un formato fijo y con una medida honesta de su incertidumbre.

Y no va a ser la última. En menos de una semana ya había versiones abiertas inspiradas en Jev, como Kev u Open-Jev, e incluso una prueba estándar, JevBench, para comparar "modelos tipo Jev". **Cuando algo tiene un éxito tan repentino, lo que viene detrás es una oleada de modelos rápidos, baratos y cada vez más específicos**: para detectar fraude, para clasificar reclamaciones, para decidir qué anuncio enseñar a cada persona.

Así que la pregunta que yo me haría, si dirigiera un equipo de marketing o de analítica, no es si usar un *chatbot* o no. Es otra: **¿qué decisiones sobre mis clientes no estoy tomando hoy porque era demasiado caro (o lento) tomarlas una a una?** Si Jevons tiene razón, esa lista va a ser mucho más larga de lo que pensamos.



Sobre el autor

Manuel Delgado Tenorio

Manuel Delgado Tenorio es asesor independiente de estrategia de datos e inteligencia artificial, profesor e investigador. Cofundó y vendió una *startup* de inteligencia artificial, fue socio de la práctica de Data & Analytics para servicios financieros en EY y responsable de Data Analytics para el sur de Europa en AWS. Es profesor asociado en IE University, donde imparte la primera asignatura de grado sobre Customer Lifetime Value en España, investiga sobre el CLV en su doctorado en la Universidad Complutense de Madrid y es autor del libro "Customer Lifetime Value (CLV): No todos tus clientes valen lo mismo".

manueldelgado.com · **LinkedIn** · **X**

Recibe cada artículo en tu correo con la newsletter Ex Datis, Claritas: manueldelgado.com/newsletter/suscribete