

The background is a vibrant yellow. It is decorated with a complex network of black and white dots of various sizes. These dots are interconnected by thin black lines, some of which are solid and others dashed. The lines and dots form a series of overlapping circles and arcs, creating a sense of movement and connectivity. The overall effect is reminiscent of a stylized molecular structure or a network diagram.

LAURENT DÉSÉCHALLIERS

CONTEXTO

Cree un contexto que permita a la IA
comprender lo que importa,
conectar sus conocimientos
y ponerlos al servicio de sus objetivos.

Le Contexte

Cómo construir un contexto que permita a la IA entender lo que importa, conectar tus conocimientos y capitalizar lo que aprendes al servicio de tus objetivos.

Laurent Deséchalliers

Nota sobre la traducción

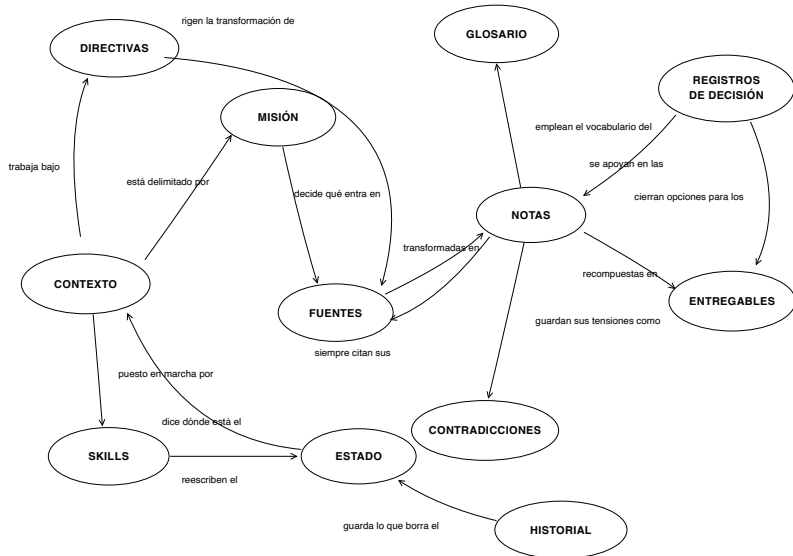
Esta edición es una traducción automática, producida por una inteligencia artificial a partir del texto francés.

No ha sido revisada por un traductor humano. Las ideas, la estructura y los ejemplos son fieles al original; en cambio, algunos giros pueden sonar torpes, y ciertos pasajes conservarán la huella de la lengua de partida.

Es el precio que aceptamos para que esta obra exista en varios idiomas y no solo en francés. En caso de duda sobre una formulación, la edición francesa es la referencia.

Definición del Contexto

Un contexto es un sistema delimitado por una misión humana, dentro del cual una IA organiza y pone en juego lo que se sabe, de dónde se sabe, cómo se relacionan los elementos y en qué punto está el trabajo, para ayudar a la persona a alcanzar su objetivo sin empezar de cero. Al hacerlo, el sistema constituye un capital informativo duradero y lo hace crecer: el conocimiento producido sigue disponible, se enriquece con el tiempo y puede reutilizarse para otros fines.



Introducción

Nunca hemos tenido tanta información a nuestro alcance. Y sin embargo seguimos dedicando buena parte de nuestra energía a volver a encontrarla, a reordenarla y a reconstruir lo que ya habíamos entendido unos días antes.

Una idea útil rara vez llega en una forma limpia. Nace de una conversación, se concreta en un documento, se topa con una objeción en una reunión, produce una decisión y luego se convierte en un artículo, una especificación o un proyecto. Nuestras herramientas conservan cada uno de esos rastros. Lo que les cuesta es conservar el hilo que los une.

La inteligencia artificial cambia una parte del problema. Sabe leer, relacionar, reformular y transformar materiales muy distintos. Pero una conversación con una IA sigue siendo un momento frágil: el tema avanza, la discusión se alarga y luego hay que recordar el objetivo, las decisiones tomadas, las fuentes utilizadas y lo que queda por hacer.

Para responder a esa dificultad construí el Contexto.

Un Contexto es un espacio de trabajo dedicado a un tema. Un humano y una IA conservan en él el objetivo perseguido, las fuentes, las ideas, las reglas y el estado del trabajo, para producir resultados sin tener que reconstruirlo todo en cada sesión.

La IA puede entonces trabajar desde una historia real. Encuentra el origen de una idea, recuerda una elección, señala una contradicción, advierte una prueba que falta, retoma un borrador donde se había quedado. **Ayuda a organizar el pensamiento y a ponerlo a prueba. El humano conserva la responsabilidad del sentido y de las decisiones.**

Este sistema nació en mi trabajo diario, de un solo archivo y de una petición muy simple dirigida a una IA: organiza lo que voy a contarte. Se fue desarrollando a base de intentos, y cada dificultad hacía visible una carencia. Sigue evolucionando: algunas de las soluciones que se presentan aquí han aguantado en varios usos, otras siguen siendo respuestas provisionales a problemas con los que todavía me encuentro. Así que este libro no parte de una teoría acabada. Parte de problemas vividos: quinientos correos que se habían vuelto imposibles de seguir, ideas dispersas entre varias herramientas, documentos de clientes a los que había que responder sin perder el trabajo ya hecho, textos retomados tras una interrupción.

Se dirige a quienes transforman información en comprensión, en decisiones y en entregables: product managers, consultores, redactores, investigadores, analistas y, en general, trabajadores del conocimiento. Propone una manera de hacer de la IA un compañero de trabajo duradero sin perder el control sobre el propio material.

El libro está organizado en dos libros complementarios. **Libro I — Comprender el contexto** cuenta cómo nació el sistema y qué cambia en la relación con la IA. **Libro II — Construir y mantener vivo el contexto** explica cómo construirlo y usarlo a lo largo del tiempo.

Pueden leerse uno detrás de otro o por separado. El capítulo **En qué orden leer este libro** presenta los distintos recorridos posibles.

Nota sobre los ejemplos

Las reflexiones y los ejemplos de este libro parten de más de treinta años de experiencia profesional en el sector informático. Proceden de situaciones vividas en proyectos y entornos profesionales diversos, y de numerosas conversaciones con product managers, CTO, engineering managers y otros profesionales de distintas empresas. Abarcan, por tanto, varias organizaciones y diferentes etapas de mi trayectoria.

Para poner estas situaciones al servicio del libro, las he modificado, generalizado o recompuesto. Los casos de uso, los datos numéricos y las métricas ilustran problemas recurrentes y lo que he aprendido de ellos. Son construcciones: ninguna describe a un cliente, un producto o una organización reales, ni refleja los datos, las prácticas o las posiciones de mi empleador.

Los ejemplos tomados de mi vida personal — la comunidad de vecinos, mis artículos y mis propios textos — se apoyan en situaciones que viví de verdad. Algunos detalles que podrían identificar a personas han sido, sin embargo, modificados. Los análisis, métodos y opiniones que aparecen en este libro son personales y no comprometen a nadie más que a mí.

Sobre el autor

Llevo varias décadas trabajando en el sector del software y hoy ejerzo como product manager. Mi recorrido me llevó del desarrollo y la arquitectura a la organización de equipos, y después al producto, a sus operaciones y a su marketing. Esos cambios de oficio desplazaron mi relación con la información. Ya no bastaba con conocer una solución técnica ni con encontrar un documento. Había que reunir señales dispersas, entender qué decían juntas, elegir entre varias direcciones y escribir textos adaptados a lectores muy distintos.

Mi interés por la inteligencia artificial y por los sistemas distribuidos viene de lejos. Hoy se encuentra con una pregunta muy concreta: **¿cómo trabajar con una IA sin cederle el control de la propia información, de las propias decisiones y del propio pensamiento?**

He colaborado con la prensa especializada y he trabajado como revisor técnico para M2 Édition y Eyrolles. Hoy escribo en malorean.net sobre product management, IA, métodos de trabajo y sistemas de información.

También colaboro con *Proposition Marketing*, una revista independiente dedicada al marketing estratégico y operativo. Escribir para lectores que vienen del marketing y de la dirección de empresa me obliga a explicar las cuestiones de

producto sin dar por supuesto un vocabulario común. Ese mismo rigor es el de este libro.

Me haría mucha ilusión intercambiar con otras personas que construyen sistemas parecidos, distintos o complementarios: el tema merece discutirse, confrontarse y enriquecerse con otras prácticas. También con quienes investigan cuestiones vecinas en la universidad — pasé algunos años en dos centros de investigación del CNRS, y volver a ese mundo sería un placer.

Laurent Deséchalliers

- laurent@malorean.net
- <https://malorean.net>

Agradecimientos

Mis primeros agradecimientos van para ti.

Para ti, que tuviste la idea algo loca de abrir este libro: un libro un poco más grueso de lo previsto, que seguramente contiene algunas repeticiones y trata de un tema abstruso. Hasta ahora he trabajado en el Contexto de forma bastante solitaria. Espero, sin embargo, que esa soledad no dure: que el tema despierte en otros las ganas de explorarlo, de discutirlo, de criticarlo y de hacerlo avanzar. Quizá los próximos artículos publicados en malorean.net sean entonces la ocasión de dar las gracias a muchos colaboradores y colaboradoras.

Doy las gracias sobre todo a mi mujer, Christine.

Ella vio cómo las tardes y los fines de semana se alargaban delante de este libro. Me apoyó durante toda su escritura, sin dejar de recordarme que era tarde y que ya iba siendo hora de cerrar ese maldito ordenador.

Prometido, solo fue un paréntesis.

Y, mucho más allá de este libro, gracias por acompañarme todos estos años en esta hermosa existencia que compartimos los dos.

En qué orden leer este libro

El libro sigue el orden en que las cosas me fueron ocurriendo: un problema vivido, una solución improvisada y, al final, un sistema. Por eso la descripción completa del Contexto no llega hasta el Libro II.

El recorrido «comprender» es ese orden. Lo lees de principio a fin, y cada pieza del sistema aparece justo cuando se acaba de mostrar la carencia que la produjo.

El recorrido «construir» está pensado para quien ya ha vivido el problema y quiere la mecánica, no el relato.

1. **Definición del contexto**, unas líneas para saber de qué hablamos.
2. **La parte 5 — Bajo el capó de un Contexto**, en el orden de sus seis capítulos: el mapa, abrir, alimentar, transformar, producir y decidir, guardar.
3. **La parte 6 — El Contexto en su ecosistema**, para conectar lo que acabas de construir con el resto de tus sistemas de información.
4. **El Libro I**, cuando una decisión de diseño te parezca arbitraria: ahí encontrarás el problema que la provocó.

Ese atajo cuesta algo. El Libro II se escribió para un lector que ya tiene en la cabeza las situaciones del Libro I, y remite a ellas. Así que te cruzarás con ejemplos a los que les faltará

un poco de contexto: ese es el precio, y sigue saliendo a cuenta si ya sabes por qué estás aquí.

Probar antes de leer

Este libro describe un método, y un método se cuenta mal. Puedes ver uno funcionando antes de leer la teoría, o mientras la lees: suele ser el mejor momento para entender para qué sirve cada pieza.

Hay tres repositorios abiertos en github.com/laurent-malorean.

Contexte Nano Notion es el más rápido de instalar. No exige entorno de desarrollo ni conocimientos técnicos: basta con un ChatGPT y un Notion. ChatGPT lleva la inteligencia del Contexto, Notion le sirve de memoria: ahí se escriben y se vuelven a leer las fuentes, las notas, las decisiones y el estado de una conversación a otra. En unos minutos tienes un Contexto que se sostiene.

Contexte V2 es la implementación completa, la que uso a diario y con la que se escribió este libro. Se instala con Claude Code o con Codex, uno u otro a elección, y pide algo más de preparación que la anterior.

LLM Wiki no es un Contexto. Es una biblioteca de conocimiento construida a partir de documentos —libros, normas, artículos, cursos— de los que la IA extrae fichas breves y enlazadas entre sí, que luego se consultan en lenguaje corriente. El Contexto mantiene el hilo de una misión; el LLM Wiki conserva en masa lo que se ha leído. La parte 6 explica cómo trabajan juntos.

Nada obliga a empezar por ahí. Pero si en algún momento la lectura te parece abstracta, es el camino más corto para que vuelva a ser concreta.

Del mismo autor



Malorean Corp — Hasta hace poco, convertir una idea en un prototipo podía llevar semanas. Hoy, una persona que cuenta con la ayuda de la inteligencia artificial puede conseguirlo en pocos días. El cambio es espectacular. Pero la verdadera transformación podría producirse en otro lugar: en nuestra forma de trabajar juntos.

Publicación prevista: finales de noviembre de 2026.

Más información en <https://malorean.net>.

Índice

LIBRO I

Comprender el Contexto

Parte 1 — El nacimiento de una necesidad	21
1.1 Quinientos correos y una historia que desaparece	22
1.2 Una primera definición	31
Parte 2 — Lo que cambia el Contexto	39
2.1 De la conversación a un trabajo que se puede retomar	40
2.2 Dos clientes, un informe técnico y una contradicción que conservar	47
Parte 3 — Tres usos concretos	57
3.1 Contestar un cuestionario del cliente sin perder el trabajo hecho . 58	
3.2 Escribir un artículo a partir de un pensamiento aún informe	64
3.3 Madurar un tema de producto en medio del caos	72
Parte 4 — Lo que los entregables dejan detrás	81
4.1 La respuesta cabe en una celda, el razonamiento no	82
4.2 Seis meses después, la respuesta ya no basta	86

LIBRO II

Construir y mantener vivo el Contexto

Parte 5 — Bajo el capó de un Contexto 93

5.1 El mapa completo 94

5.2 Abrir: recuperar la misión y el presente 105

5.3 Alimentar: hacer entrar el material 116

5.4 Transformar: pasar de las fuentes a las ideas 123

5.5 Producir y decidir 133

5.6 Guardar y retomar 143

Parte 6 — El Contexto en su ecosistema 151

6.1 Consultar conocimientos estructurados 152

6.2 Consultar el producto real 159

6.3 El día en que tuve tres Contextos 168

6.4 Leer en casa del vecino sin mudarse con él 174

6.5 Lo que cuesta la circulación 181

Antes de cerrar

Cierre 189

7.1 Epílogo 190

7.2 Parentescos y vecindades 193

LIBRO I

Comprender el Contexto

- 1 El nacimiento de una necesidad
- 2 Lo que cambia el Contexto
- 3 Tres usos concretos
- 4 Lo que los entregables dejan detrás

Parte 1

El nacimiento de una necesidad

El Contexto no nació de una idea sobre la inteligencia artificial. Nació de un buzón de correo que se había vuelto inmanejable y de algo que tardé en comprender: el verdadero problema no era el número de mensajes.

Esta parte cuenta ese origen y extrae de él una primera definición: lo bastante sencilla para recordarla y lo bastante completa para guiarnos durante el resto del libro.

Quinientos correos y una historia que desaparece

La primera vez que creé un Contexto no buscaba inventar un método de trabajo con la inteligencia artificial. Solo intentaba dejar de perder el hilo de mi comunidad de vecinos.

Formo parte de la junta de la comunidad. Por eso me llegan los mensajes del administrador, de los propietarios, de los inquilinos y de los proveedores:

- un correo anuncia un problema nuevo,
- el siguiente concreta un presupuesto,
- un tercero aplaza una intervención,
- una semana después alguien contesta en el mismo hilo para hablar de otro edificio.

Mientras tanto han avanzado otros tres asuntos.

El volumen no era el único problema. Unos quinientos correos se repartían entre unos quince asuntos:

- obras,
- cuentas,
- seguros,
- juntas generales,
- morosos,
- concursos,
- incidencias y peticiones varias

- ...

Cada uno avanzaba a su ritmo, a veces durante meses.

Quien sigue una obra a diario lleva su historia en la cabeza sin esfuerzo. Sabe qué proveedor intervino, por qué cambió el calendario y qué decisión sigue esperando respuesta. Cuando yo abría un mensaje nuevo, no siempre tenía esa continuidad. Tenía que remontar el hilo, recuperar los intercambios anteriores y rehacer la situación antes incluso de poder entender la información nueva.

Quería poder hacer preguntas sencillas:

- ¿cómo va lo del tejado?
- ¿qué proveedor mandó el último presupuesto de los canalones?
- ¿qué decidimos sobre aquella filtración?
- ¿qué esperamos todavía del administrador?

Mi buzón tenía las respuestas. Lo que no me daba era la visión de conjunto.

Un archivo y una frase

Empecé con el montaje más sencillo posible: un archivo y una instrucción dada a una IA.

▮ *Reúne la información y divídela por temas.*

Le pasaba los correos. Ella los asociaba a temas y mantenía un resumen de cada uno.

- Todo lo de las obras acababa junto.
- Las cuestiones de dinero formaban otro bloque.

- Los proveedores, los seguros y las juntas fueron encontrando su sitio.

Ese primer intento ya aportaba algo. Podía preguntar cómo iba un tema sin releer todos los mensajes que lo trataban. La IA ya no buscaba solo una coincidencia en el buzón: me presentaba lo que había entendido de la situación.

El archivo se reorganizaba a medida que llegaba nueva información.

- Aparecía un apartado cuando un tema nuevo cobraba peso.
- Dos apartados se separaban cuando empezaban a mezclar realidades distintas.

La información ya no se limitaba a acumularse. Tomaba una forma con la que se podía trabajar.

Luego ese primer sistema dio una respuesta exacta y aun así insuficiente.

La obra estaba terminada, pero su historia había desaparecido

Cuando una obra llegaba a su fin, su resumen decía: «obra terminada, recepción conforme».

Ese era su estado presente. También era una pérdida.

- ¿Cuándo había empezado la obra?
- ¿En qué momento se había retrasado?
- ¿Por qué había cambiado el calendario?
- ¿Cuándo se había firmado el anexo?

Todas esas etapas habían contado. La última actualización las había borrado en favor de la situación final.

Yo había pedido un resumen. La IA lo había mantenido al día sustituyendo el estado antiguo por el nuevo. Había seguido la instrucción, pero la instrucción no protegía la historia.

Necesitaba, pues, dos vistas del mismo asunto.

- La primera tenía que responder deprisa a la pregunta «¿cómo vamos?».
- La segunda tenía que conservar los hechos en su orden: inicio de la obra, retraso, nuevo acuerdo, intervención, recepción.
- El estado presente permitía actuar.
- La historia permitía entender cómo se había llegado a ese estado y comprobar un punto en caso de duda.

Esa distinción nació de un fallo muy concreto. La tomaba por una mejora de orden. Se convirtió en uno de los principios más importantes de mi trabajo con la IA: **lo que es cierto ahora y lo que ha pasado no cumplen la misma función.**

La IA no podía adivinar mis reglas

Otro problema apareció con los nombres:

- dos proveedores compartían apellido.
 - Para distinguirlos había que entender de qué iba el correo.
- una propietaria aparecía unas veces con su apellido de soltera y otras con el de casada.

Sin ninguna indicación, la IA podía crear dos personas donde solo había una.

Otras diferencias venían de nuestra manera de trabajar.

- Los morosos por debajo de cien euros no se seguían como los demás.
- Ese límite no venía de ninguna regla universal que la IA pudiera conocer.
- Venía de cómo hacíamos las cosas.

Empecé a formular ese conocimiento como reglas duraderas, escritas con todas sus letras en el archivo:

```
# Reglas de tratamiento
```

- Martín citado a propósito del tejado designa al tejador;
Martín citado a propósito
de las zonas verdes designa al jardinero.
- Dupont-Lemoine y Lemoine designan a la misma propietaria,
vivienda 14.
- Presentar a los morosos por debajo de cien euros en una
lista aparte.

Nada más sofisticado que eso. Tres frases que habría tenido que repetir en cada conversación y que no volví a decir ni una sola vez.

Cada regla venía de un error observado o de una ambigüedad real. Recogía un conocimiento que yo llevaba en la cabeza y que la IA no podía reconstruir con certeza.

Así el sistema empezaba a guardar dos tipos de información:

- la información sobre la comunidad,
- la manera en que quería que se tratara.

El archivo que ayudaba empezó a estorbar

Con las semanas, el archivo creció. Pasó de quinientas líneas, luego de ochocientas, luego de mil.

Ese crecimiento demostraba su utilidad:

- los asuntos se seguían mejor,
- los proveedores quedaban identificados,
- las decisiones se recuperaban.

Pero lo que había hecho posible el sistema empezaba a limitar su uso.

- Demasiada información distinta compartía el mismo espacio.
- Las reglas se mezclaban con los hechos.
- Las cuentas convivían con las obras, las personas, los seguros y los concursos.

Encontrar la información adecuada volvía a resultar difícil.

Primero separé las reglas de tratamiento. No describían la comunidad; decían cómo trabajar con la información que la afectaba. Aislarlas permitía encontrarlas y aplicarlas más fácilmente.

Después repartí la información entre varios temas. Las obras tuvieron su propio espacio, igual que las cuentas, los proveedores, los acuerdos de la junta o los seguros. Una vista general decía qué contenía cada uno. La IA podía entonces abrir el tema pertinente para responder, en lugar de recorrerlo todo cada vez.

Por último agrupé el vocabulario particular: términos jurídicos, siglas y expresiones propias de una comunidad de vecinos. No se trataba de construir una enciclopedia. Se trataba de dar un sentido estable a las palabras de las que dependía entender los correos.

Seguía sin tener ninguna gran arquitectura en la cabeza. Añadía una pieza cuando un problema hacía necesaria esa pieza.

Confiar sin perder la fuente

La IA se equivocaba a veces. Podía juntar dos informaciones que no trataban exactamente del mismo asunto o elegir al proveedor equivocado entre dos homónimos. El sistema tenía que acelerar la comprensión sin convertirse en la única versión disponible de los hechos.

Conservé todos los correos originales. Cuando necesitaba comprobar una respuesta o había algo importante en juego, podía volver a la información tal como había llegado. El resumen daba acceso rápido. La fuente permitía comprobar. Esa elección es lo que hacía viable el uso. Podía aceptar algún error puntual al organizar los correos porque los mensajes originales seguían accesibles. El mismo grado de confianza no serviría para un documento bancario, una obligación normativa o una decisión irreversible. El riesgo depende siempre de lo que se le confía al sistema y de la posibilidad de comprobarlo.

Así que el Contexto no sustituía a mi buzón. Convertía su contenido en una memoria de trabajo y me dejaba una puerta de vuelta a los mensajes originales.

Lo que había aparecido

En unas semanas, el punto de partida —un archivo y una frase— se había convertido en un conjunto capaz de cubrir varias necesidades: mostrar el estado presente, conservar la historia, aplicar reglas propias de la comunidad, distinguir los asuntos, estabilizar el vocabulario y volver a las fuentes.

Cuando llegaba un correo nuevo, podía fijarme en lo que cambiaba en vez de reconstruir todo lo anterior. Cuando surgía una pregunta, podía plantearla en los términos del problema en vez de buscar el mensaje adecuado en el hilo adecuado. Cuando la IA se equivocaba, podía corregir una regla o volver al texto original.

No había diseñado el Contexto de una sola pieza. Lo había visto aparecer a través de una sucesión de carencias: el resumen borraba la historia, los nombres se confundían, las reglas se perdían, el archivo se hacía demasiado grande. Cada corrección hacía el trabajo más duradero y más fácil de retomar.

Ese origen cuenta. El Contexto no nació del afán de clasificar más. Nació de la necesidad de entender una información que cambia, conservar las pruebas necesarias para comprobarla y continuar el trabajo sin empezar de cero.

PARA RECORDAR — DOS VISTAS DEL MISMO ASUNTO

El estado responde a «¿cómo vamos?». La historia responde a «¿cómo hemos llegado hasta aquí?». Un resumen que se actualiza aplastando el anterior conserva la primera y pierde la segunda. Hacen falta las dos: una para actuar, la otra para entender y comprobar.

Una primera definición

El sistema nacido de aquellos quinientos correos ya puede recibir una primera definición.

Un Contexto es un espacio de trabajo dedicado a un tema, donde una persona y una IA guardan lo que necesitan para entender, tomar decisiones y producir resultados a lo largo del tiempo.

Esta definición es deliberadamente sencilla. El Contexto que describe este libro tiene una organización precisa, que detallará la segunda mitad. Para entender lo que aporta, al principio bastan seis referencias:

- un objetivo,
- información,
- unas ideas,
- un estado del trabajo,
- unas reglas propias,
- unos resultados que producir.

Un tema y un objetivo

El Contexto de la comunidad no pretendía contener todo lo que yo sabía. Perseguía un objetivo: entender los asuntos en curso, conservar su historia y encontrar la información necesaria para actuar en la junta de la comunidad.

Ese objetivo formaba una frontera. Un correo sobre el tejado tenía su sitio. Una lectura interesante sin relación con la comunidad, no. Ese límite evitaba que el sistema se convirtiera en un depósito donde se acumulara toda información que pareciera interesante.

Un Contexto empieza, pues, por una intención. Puede servir para:

- escribir un libro,
- explorar un problema de producto,
- preparar una decisión,
- contestar cuestionarios del cliente,
- seguir una entidad de negocio,
- ...

Información cuyo origen se conserva

La información entra en un Contexto de dos maneras.

La primera recibe lo que ya existe antes de la sesión de trabajo:

- un correo,
- un artículo,
- un libro,
- una página web,
- la transcripción de una entrevista
- ...

Son las **fuentes documentales**. Llegan como objetos ya formados y se mantienen estables. En el caso de la

comunidad, el correo que anuncia el aplazamiento de una intervención pertenece a esta primera categoría.

La segunda recibe lo que se formula durante el trabajo con la IA. Cuando explico una situación, corrijo una interpretación, afinó un matiz o expreso una convicción salida de mi experiencia, lo que digo se convierte en una **fente de sesión**. Se construye poco a poco a lo largo del intercambio. Cuando explico que dos nombres designan a la misma propietaria, lo que digo entra primero por esta segunda puerta.

La diferencia está, pues, en el momento y la manera en que aparece la información.

- El documento existe antes de la sesión y se trae a ella.
- La fuente de sesión nace durante la sesión y crece con ella.

Después, las dos cumplen la misma función: conservar lo recibido o dicho antes de interpretarlo.

- No garantizan que todo lo que contienen sea cierto.
- Permiten recuperar lo que se recibió o se dijo y valorar después su fiabilidad.

A partir de esas fuentes, el Contexto puede extraer una idea, una regla o un estado del trabajo. El conocimiento obtenido tras interpretar y organizar la información se guarda en otro lugar. Si una fecha parece incorrecta, si se ha confundido a dos personas o si una interpretación se vuelve discutible, se puede así volver a su origen.

Ideas hechas utilizables

Acumular fuentes no basta. Quinientos correos bien guardados siguen siendo quinientos correos por leer.

Por eso el Contexto va convirtiendo esas fuentes en conocimiento utilizable. Hace aparecer:

- temas,
- hechos importantes,
- reglas,
- distinciones,
- preguntas abiertas,
- relaciones.

Un intercambio aislado se suma a la historia de la obra a la que pertenece. Varias menciones de un mismo proveedor enriquecen un conocimiento común. Una corrección impide que una confusión se repita.

Esta etapa intermedia cuenta porque hace utilizable la información.

- Las fuentes conservan lo que se recibió o se dijo.
- El conocimiento organizado permite entender y actuar.

En trabajos más complejos, algunas ideas podrán servir varias veces: un mismo conocimiento de producto alimenta una decisión, una especificación y una documentación; un mismo hecho comprobado ayuda a contestar varios cuestionarios del cliente. El resultado cambia, mientras que la idea y su procedencia siguen disponibles.

Un estado que permite continuar

Un trabajo en curso siempre tiene un estado actual: lo que se sabe, lo que sigue siendo incierto, lo que se ha decidido y lo que debe venir después.

En la comunidad, el estado respondía a una pregunta inmediata: ¿cómo va hoy esta obra? Su historia respondía a otra: ¿cómo hemos llegado hasta aquí? El sistema necesitaba las dos para permitir que el trabajo se reanudara de verdad.

Esta distinción va más allá del seguimiento de una obra.

Cuando interrumpo la escritura de un texto o la exploración de un problema de producto, quiero encontrar algo más que un montón de documentos. Quiero saber:

- qué está asentado,
- qué punto sigue siendo frágil,
- qué acción puede volver a poner el trabajo en marcha.

El Contexto conserva esa continuidad fuera de la conversación.

- La sesión con la IA puede detenerse.
- El trabajo guarda un estado al que la sesión siguiente podrá volver.

Reglas propias del trabajo

La IA dispone de conocimientos generales. En cambio ignora buena parte de lo que hace particular a una situación:

- los dos proveedores que comparten apellido,
- la manera en que un equipo califica una señal de un cliente,
- el nivel de prueba exigido antes de publicar una afirmación
- el tono elegido para un libro,

- ...

El Contexto guarda esas reglas junto al trabajo que rigen. Se pueden corregir cuando la experiencia muestra su límite.

Después se aplican sin que la persona tenga que reformularlas en cada petición.

Esas reglas no le dan a la IA la responsabilidad del trabajo. Hacen explícito lo que la persona espera de ella. La persona fija el objetivo, aporta o valida las fuentes, corrige las interpretaciones y toma las decisiones. La IA incorpora la información, establece relaciones, organiza, pregunta, reformula y prepara resultados dentro de ese marco.

Resultados adaptados a una necesidad

El Contexto sigue centrado en aquello que debe hacer posible.

En la comunidad, el resultado podía ser:

- una respuesta a una pregunta,
- un balance de las obras,
- la preparación de una conversación.

En otros usos será:

- un artículo,
- un cuestionario del cliente,
- una documentación.

El mismo espacio puede producir varios resultados porque conserva las fuentes antes de darles una forma final. Las fuentes no quedan encerradas en el primer documento que las

usa. Las ideas siguen disponibles para otra pregunta u otro público.

Esa capacidad distingue al Contexto de un simple archivo.

- Un archivo permite conservar.
- El Contexto organiza cómo se retoma y se transforma lo conservado.

Una forma que puede empezar de manera muy sencilla

En mi práctica, un Contexto toma la forma de archivos que puedo leer, mover y conservar al margen de la herramienta que me ayuda a trabajarlos. Esa forma responde a mi necesidad de control y de duración. Son posibles otras realizaciones si preservan las mismas funciones.

Sin embargo, el primer Contexto no tenía nada de sofisticado. Cabía en un archivo. Creció cuando sus límites se hicieron visibles. Esa progresión sigue siendo importante: no hace falta construir todo el sistema que describe este libro para empezar a probar la idea.

Bastan un tema, un objetivo y unas primeras fuentes.

- La IA ayuda a organizarlos.
- La persona observa lo que falta, añade una regla, conserva una fuente o separa dos informaciones que no tienen el mismo papel.
- La arquitectura aparece al ritmo de los problemas que van saliendo.

Así que el Contexto no designa tanto una estructura de carpetas concreta como una relación de trabajo duradera

entre una persona, una IA y un conjunto de información organizado en torno a un objetivo.

1

La parte siguiente examinará qué cambia esa relación. Una conversación con una IA ya puede ser fecunda. El paso al Contexto empieza cuando esa conversación tiene lugar en un entorno capaz de conservar su objetivo, su historia y el estado del trabajo más allá del momento presente.

PARA RECORDAR — LAS SEIS REFERENCIAS DE UN CONTEXTO

Un objetivo que traza una frontera, información cuyo origen se conserva, ideas hechas utilizables, un estado que permite continuar, unas reglas propias y resultados que producir. Esas seis referencias bastan para empezar. La arquitectura viene después, cuando un problema la hace necesaria.

Parte 2

Lo que cambia el Contexto

Una conversación con una IA ya puede aportar mucho. Explica, reformula, propone y a menudo da una respuesta mejor que la que habrías encontrado solo.

Esta parte se pregunta, pues, qué añade un Contexto. La respuesta no está en la calidad de las respuestas obtenidas, sino en lo que se conserva de una sesión a otra: un trabajo que continúa donde se había detenido y documentos que mantienen sus desacuerdos en vez de limarlos.

De la conversación a un trabajo que se puede retomar

Una conversación con una IA puede empezar de forma muy sencilla.

Le doy una idea todavía confusa. Ella la reformula. Pido un guion, luego otro. Me plantea una objeción en la que no había caído. Concreto mi intención, descarto una vía, profundizo en otra. En unos minutos la conversación hace avanzar algo que quizá habría quedado mucho tiempo delante de una página en blanco.

Esa es una de las grandes bazas del chat: basta una pregunta para empezar. Esta forma le va muy bien a una tarea corta. Pido una síntesis, una explicación, unas cuantas opciones. Llega la respuesta, la uso y la conversación puede terminar. La relación cambia cuando el trabajo dura.

Cinco frases antes de poder escribir

Vuelvo a abrir una conversación de hace seis días para retomar un texto empezado la semana anterior. Antes de poder escribir una sola línea, tecleo esto:

Retoma el texto sobre la calidad de los partes de intervención. Está dirigido a responsables de operaciones, no a desarrolladores. Habíamos descartado acortar el formulario, es pronto para decidirlo. El dato sobre las

auditorías viene de una sola entrevista, trátalo como hipótesis. Íbamos por la tercera sección.

Cinco frases solo para volver a situar a la IA en un trabajo que ya conocía seis días antes.

Todo lo que acabo de teclear existía antes de que lo tecleara: el enfoque, el público, la decisión, el punto frágil y el lugar en el texto estaban en la conversación anterior. Simplemente los recuperé de mi propia memoria, porque la conversación solo los había conservado en orden cronológico.

El trabajo sigue otro orden.

- Una instrucción dada al principio tiene que seguir valiendo al final.
- Una fuente añadida ayer tiene que poder sostener un párrafo escrito mañana.
- Una decisión antigua tiene que impedir que se reabra la misma discusión.
- Un punto pendiente tiene que reaparecer en el momento oportuno.

Cuanto más crece la conversación, más tiene la persona que sostener ese orden en la cabeza. Cuando empieza una sesión nueva, reconstruye la situación para la IA: este es el objetivo, esto es lo que ha cambiado, este es el tono, estas son las decisiones, aquí nos quedamos.

El chat sigue siendo útil. La carga de la continuidad, en cambio, recae sobre quien lo usa.

Dar un lugar a lo que debe durar

Un Contexto cambia la relación al darle al trabajo un espacio distinto de la conversación.

La discusión sigue siendo el lugar del intercambio. Ahí se pregunta, se explora, se afina y se corrige. Lo que debe durar pasa al espacio de trabajo:

- el objetivo,
- las fuentes,
- las reglas,
- las ideas asentadas,
- las decisiones,
- los textos,
- el punto en que está el trabajo.

La diferencia puede parecer pequeña. Es comparable a la que separa una reunión del proyecto del que habla. Una buena reunión puede hacer avanzar el proyecto, pero el proyecto no cabe entero en la memoria de la reunión. Los documentos, las decisiones y las acciones siguen existiendo cuando termina. Guardar la transcripción completa del chat no produciría el mismo efecto. Es valiosa cuando se quiere recuperar exactamente lo que se dijo, pero sigue la forma de la conversación: una pregunta, una respuesta, una corrección, una digresión, una vuelta al asunto. Para retomar el trabajo habría que releer esa secuencia y decidir qué cuenta ahora. El Contexto, en cambio, transforma lo que se dijo.

- Una corrección va al sitio donde podrá impedir que el error vuelva.
- Una decisión sigue visible como decisión.
- Una fuente sigue ligada a lo que sostiene.
- Un texto existe en su última versión completa.

Así, la memoria se organiza para facilitar la acción. No busca solo conservar el pasado. Prepara la intervención siguiente.

Una página para recuperar treinta y dos textos

Medí lo que eso cambia al volver al trabajo sobre este libro.

El manuscrito tenía entonces treinta y dos textos. Algunas partes estaban escritas, otras todavía vacías. Una sección se había dividido porque ocupaba un espacio desproporcionado.

Dos pasajes esperaban mi relectura. Una afirmación no se apoyaba en ningún ejemplo concreto y había que rehacerla.

Por separado, ninguno de esos elementos es difícil de entender. Sin embargo, juntos forman una situación que hay que reconstruir antes de poder escribir una sola línea útil.

Podría abrir los treinta y dos textos, releer el guion, recorrer mis conversaciones antiguas y buscar las observaciones que quedaron pendientes. Ese trabajo de arqueología llevaría tiempo y dejaría una duda: ¿habré olvidado justamente el punto que debía impedir un error?

En su lugar, abro el Contexto del libro. Me tiende una ficha para volver al trabajo, que aquella mañana decía más o menos esto:

Volver al trabajo

Dónde estamos

Existen treinta y dos textos, del guion a la conclusión.
Las partes técnicas están escritas; las dos primeras siguen
siendo demasiado teóricas.

Qué hay que saber antes de continuar

El lector al que nos dirigimos descubre el tema: pierde el
hilo en cuanto un concepto aparece antes de un ejemplo
concreto.

La afirmación sobre el fracaso de los sistemas documentales
no se apoya todavía en ninguna experiencia personal.

La próxima acción

Reescribir la apertura en torno al caso de los quinientos
correos de la comunidad.

Qué ha quedado abierto

Dos pasajes esperan mi relectura.
Varias fuentes han entrado sin haber sido convertidas en
notas.

Una página, leída en menos de un minuto. Sé qué hacer, y no
he abierto ninguno de los treinta y dos textos.

Continuar no es volver al trabajo

**Continuar consiste en conocer la acción siguiente. Volver
al trabajo consiste en recuperar su sentido antes de
realizar esa acción.**

Una lista de tareas puede decirme que hay que tratar una
fuente. No me explica por qué esa fuente es prioritaria, qué
parte del libro debe iluminar ni qué riesgo debe permitir
corregir.

Para volver al trabajo necesito un conjunto más rico:

- el objetivo que persigue el trabajo;
- el punto al que ha llegado realmente;
- las decisiones que siguen orientando lo que viene;
- los puntos frágiles que no deben olvidarse;
- una próxima acción lo bastante precisa para volver a empezar.

El archivo, de hecho, se lee en ese orden. «Treinta y dos textos» da una medida. «Pierde el hilo en cuanto un concepto aparece antes de un ejemplo concreto» marca una dirección. La próxima acción se desprende de ambas y nombra el caso concreto que debe abrir el libro.

Lo recuperado fue un razonamiento, no solo una posición en una lista.

Tampoco trata de recrear en mi cabeza cada detalle de la sesión anterior. Un resumen demasiado corto borra la razón de una elección. Un informe demasiado largo aplaza el momento en que el trabajo vuelve a empezar.

Una interrupción no elige su momento

Al principio imaginaba la vuelta al trabajo como consecuencia de un cierre limpio. Una sesión terminaría, su estado quedaría cuidadosamente resumido y la próxima acción anotada.

La práctica rara vez sigue ese guion.

- Empieza una reunión.
- Se acaba la batería.
- El día termina antes de lo previsto.
- Otra urgencia ocupa el lugar del tema.

El momento en que esa memoria resulta más útil es justo aquel en que no se preparó el final.

El Contexto mantiene, pues, durante el trabajo su capacidad de retomar. Las decisiones encuentran su sitio cuando se toman, las tareas cuando se formulan, las fuentes cuando entran.

Retomar pasa a ser una propiedad del sistema en vez de una disciplina heroica del usuario.

Aun así, la persona conserva un papel decisivo. Puede juzgar que la próxima acción propuesta ya no es prioritaria, que una decisión antigua debe reabrirse o que un hecho nuevo cambia el objetivo. La memoria no congela la trayectoria. Hace visible la trayectoria desde la que se decide modificarla.

Las cinco frases que volvía a teclear al reabrir mi conversación ya no las tecleo. Están escritas en alguna parte, y alguien que no soy yo las relee en mi lugar al empezar cada sesión.

PARA RECORDAR — EL CHAT, EL ENTORNO Y LA VUELTA AL TRABAJO

El chat es el lugar del intercambio; el Contexto es el lugar de lo que debe durar. Una conversación conserva la información en el orden en que se dijo; el trabajo necesita una organización distinta. Continuar es conocer la acción siguiente; volver al trabajo es recuperar su sentido antes de realizarla. Una interrupción no avisa: la capacidad de retomar se mantiene durante el trabajo, no al cerrarlo.

Dos clientes, un informe técnico y una contradicción que conservar

Hasta aquí he descrito el Contexto y varios efectos que produce. Pero ¿qué aspecto tiene de verdad cuando se abre? ¿Qué se ve en los archivos? ¿Cómo hacen varias fuentes que aparezcan ideas que pueden ser ciertas por separado y aun así difíciles de conciliar?

El tema de este Contexto inventado es sencillo: entender por qué algunos técnicos entregan partes de intervención incompletos, antes de decidir si el producto debe cambiar. Su misión cabe en unas líneas:

Entender qué degrada la calidad de los partes de intervención. Distinguir las dificultades de cumplimentación, las restricciones de organización y las necesidades que vienen del uso de los datos. No elegir una solución antes de haber confrontado varias fuentes.

Esa misión le da una dirección al trabajo. Pide entender un problema, no todavía diseñar una funcionalidad.

Un primer testimonio revela una idea

El primer documento que entra en el Contexto es la transcripción de una entrevista con una responsable de operaciones.

Explica:

*«Los técnicos rellenan sus partes a menudo el viernes.
Sobre el terreno, el formulario les parece demasiado largo.
Al final de la semana han olvidado parte de los detalles.»*

2

La IA conserva esa entrevista como **fuentes**. Después saca de ella una idea en una ficha aparte, lo que llamo una **nota atómica**:

Una cumplimentación demasiado exigente puede llevar a algunos técnicos a aplazar el parte y a perder detalles.

Esa formulación no dice que todos los técnicos actúen así ni que la longitud del formulario sea la única causa. Conserva la idea central del testimonio: en la organización descrita, el informe se rellena más tarde y parte de la información se pierde entre la intervención y su redacción.

La fuente sigue disponible. La nota no la sustituye; hace que su idea principal sea más fácil de encontrar y de confrontar con lo que venga después.

El segundo testimonio enriquece la misma nota

Unas semanas más tarde entra en el Contexto una segunda entrevista.

Esta vez un responsable de operaciones cuenta:

«Hicimos obligatorios doce campos para mejorar la calidad de los datos. Ahora algunos técnicos escriben «sin novedad» o ponen un guion solo para poder validar.»

Ese testimonio no describe exactamente el mismo comportamiento. Los técnicos de la primera entrevista aplazan la cumplimentación. Los de la segunda rellenan el formulario, pero esquivan lo que exige.

Las dos fuentes tratan, sin embargo, de la misma idea central: una restricción pensada para obtener más información puede acabar reduciendo la calidad real de lo que se introduce.

La segunda entrevista se suma, pues, a la primera **nota atómica**. No sustituye a la primera fuente ni crea una idea rival. Enriquece la nota al revelar otra forma del mismo problema: después del aplazamiento, una apariencia de información completa.

La nota ya puede afirmar:

Una cumplimentación demasiado exigente puede llevar a algunos técnicos a aplazar o a esquivar el parte, en detrimento de la información recogida.

Hay dos fuentes referenciadas. Cada una conserva su papel y sus límites. La nota gana solidez sin pretender que dos testimonios basten para medir el alcance del fenómeno.

El informe técnico revela otra idea

Añado después un informe técnico dedicado al uso de los datos de intervención.

El documento defiende otra exigencia. Para entender las averías recurrentes, comparar equipos y preparar ciertas auditorías, una empresa necesita partes suficientemente estructurados. Si cada técnico escribe lo que quiere en un campo libre — o no rellena nada —, la cumplimentación quizá se vuelva más fácil, pero los datos pierden buena parte de su utilidad colectiva.

Esa **fuerza** no resta validez a los testimonios anteriores. Los técnicos descritos siguen aplazando o esquivando. El PDF revela otra realidad: suprimir toda restricción también tendría un coste.

La IA podría verse tentada a fundir estos elementos en la primera nota y a reescribirla como un compromiso general. Sería un error. El informe técnico no resta validez a lo observado en las entrevistas; plantea una afirmación diferente.

Produce, pues, una segunda **nota atómica**:

Hacen falta partes estructurados para comparar las intervenciones y usar los datos de intervención.

El Contexto contiene ya dos ideas distintas:

- la restricción de cumplimentación puede degradar la información realmente recogida;
- la falta de estructura puede degradar el uso posterior de esa información.

Las dos notas pueden ser ciertas a la vez. Sin embargo, orientan la decisión de producto en direcciones opuestas.

La oposición debe seguir visible entre las notas

El papel del Contexto no es modificar la primera nota hasta que la contradicción desaparezca. Debe conservar las dos proposiciones y hacer explícita su relación.

Una **nota temática** puede entonces expresar lo que se comprende al leerlas juntas:

La calidad de un parte depende de una tensión entre el esfuerzo pedido al cumplimentarlo y la estructura necesaria para usarlo.

Esa tercera nota no es una media diplomática. No sustituye a ninguna de las dos notas atómicas. Muestra el problema nuevo que aparece cuando se leen juntas.

La contradicción se suma también al estado del trabajo: ¿cómo obtener información estructurada sin provocar el aplazamiento o el esquivar de la cumplimentación? Ninguna de las tres fuentes permite responder todavía.

Esto es lo que se ve en pantalla

El Contexto usado para este ejemplo existe como una pequeña carpeta de archivos. Su estructura no es el concepto en sí: es mi manera actual de darle una forma concreta.

```
contexte-qualite-comptes-rendus/  
├─ 00-frame/  
│   ├── mission.md  
│   └── directives.md  
├─ 01-state/  
│   ├── state.md  
│   └── contradictions.md  
├─ 03-sources/  
│   ├── SRC00001-entretien-exploitation.md  
│   ├── SRC00002-entretien-exploitation.md  
│   └── SRC00003-livre-blanc-donnees-intervention.md  
├─ 04-notes/  
│   ├── atomic/  
│   │   ├── AT000001-la-contrainte-de-saisie.md  
│   │   └── AT000002-des-comptes-rendus-structures.md  
│   └── thematic/  
│       └── THM00001-la-qualite-depend-de-la-saisie.md  
└─ 05-outputs/  
    └── questions-prochain-entretien.md
```

Ahora puedes conectar las palabras usadas desde el principio del libro con objetos visibles.

- La **misión** explica por qué existe la carpeta.
- Las **fuentes** conservan lo recibido antes de interpretarlo.
- Las **notas atómicas** asientan ideas distintas.
- La **nota temática** hace inteligible su tensión.
- El **estado** dice qué deja abierto esa tensión.
- El **entregable** prepara la acción siguiente.

Nada de esto exige un programa misterioso. VS Code (aunque podría ser la aplicación Codex o Claude Cowork) es aquí la ventana por la que accedo al Contexto. El Contexto reside en los archivos, en la función de cada uno y en las relaciones conservadas entre ellos.

La respuesta se apoya en el trabajo realizado

Unos días después retomo el trabajo y le propongo a la IA:

«Quizá deberíamos hacer opcionales todos los campos.»

En una conversación aislada, la IA podría defender esa idea, proponer una interfaz más ligera o enumerar sus ventajas. En el Contexto se encuentra con las dos notas atómicas y con la tensión que las une.

Puede entonces responder:

Esa propuesta reduciría probablemente parte de la fricción descrita en las entrevistas. Choca, en cambio, con la necesidad de datos estructurados señalada en el informe técnico. Las fuentes muestran dos restricciones reales; todavía no permiten elegir entre simplificación, cumplimentación por etapas, dictado por voz o adaptación de los campos al tipo de intervención.

Esa respuesta no viene de un conocimiento general del product management. Se apoya en el trabajo en curso. La IA sabe qué pregunta intentamos entender, qué fuentes sostienen qué ideas y qué contradicción sigue abierta. No decide por mí. Solo impide que mi nueva intuición borre una de las realidades ya establecidas.

El primer entregable prepara la fuente siguiente

El estado del Contexto no dice: «diseñar un formulario nuevo».

Separa lo que parece asentado de lo que sigue siendo incierto.

Lo que parece asentado *Las restricciones de cumplimentación pueden provocar aplazamiento y evasión. Sigue haciendo falta una estructura suficiente para usar los partes.*

Lo que sigue abierto *Todavía no sabemos qué fricción pesa más ni qué información debe seguir estructurada a toda costa.*

Próxima acción *Preparar una entrevista que distinga las causas del problema en lugar de presentar enseguida una solución.*

El primer entregable es, pues, modesto: una guía de preguntas para las nuevas entrevistas. Debe permitir entender el momento de la cumplimentación, las restricciones del terreno, la utilidad real de los distintos campos y las formas de esquivarlo.

El entregable no resuelve la contradicción. Prepara la información que quizá permita resolverla más adelante.

Un objeto de trabajo, no una máquina de fundirlo todo

Este ejemplo muestra también un error que la IA puede cometer: querer meter una fuente nueva en una nota existente porque los temas se parecen.

Una **nota atómica** puede crecer cuando varias fuentes tratan de la misma proposición. Es lo que ocurre con las dos entrevistas. Pero el informe técnico no es una prueba nueva

de esa proposición. Sostiene una idea distinta, que merece su propia nota.

La distinción cuenta. Si cada contradicción reescribe la nota anterior, *el conjunto de ideas* acaba limando las tensiones. El conocimiento parece coherente, pero ya no corresponde a lo que las fuentes aportaron de verdad.

El Contexto no garantiza que la IA elija siempre la frontera correcta. La persona debe poder rechazar una fusión, pedir una nota nueva y comprobar la relación creada entre ellas. La estructura hace posible y duradera esa corrección; no la decide automáticamente.

Ya has visto el objeto:

- **fuentes** que quedan intactas,
- **notas atómicas** que conservan cada una su centro,
- una **nota temática** que acerca sin fundir,
- una **contradicción** que queda abierta y un **entregable** que prepara lo que viene.

Los capítulos siguientes podrán volver a los usos del Contexto y después abrir con más detalle su arquitectura a partir de estos elementos ya visibles.

PARA RECORDAR — UNA CONTRADICCIÓN SE CONSERVA

Dos fuentes pueden ser ciertas a la vez y orientar la decisión en direcciones opuestas. Una nota atómica crece cuando una fuente refuerza la misma proposición; una proposición distinta merece su propia nota. Fundir para hacer desaparecer la tensión produce un conocimiento coherente que ya no corresponde a las fuentes.

Parte 3

Tres usos concretos

Las dos partes anteriores han dicho qué cambia el Contexto. Esta parte lo muestra en funcionamiento, en tres usos que practico de verdad.

Contestar un cuestionario del cliente, escribir un artículo a partir de un pensamiento aún informe, madurar un tema de producto en medio de una jornada fragmentada. Tres duraciones, tres tipos de material, tres destinatarios, y el mismo movimiento cada vez: recuperar lo que ya se había entendido, transformarlo y producir sin agotar lo que ha servido.

Contestar un cuestionario del cliente sin perder el trabajo hecho

El primer cuestionario del cliente que traté con este sistema tenía cuarenta y tres preguntas.

Había que responder sobre la arquitectura del producto, su seguridad, las copias de seguridad, la continuidad del servicio y la protección de datos. Parte de la información estaba en nuestra documentación. Otra estaba enterrada en archivos de configuración. Para algunas cosas había que recurrir al equipo que explotaba el servicio.

Tuvimos que reunir siete fuentes técnicas y hacer una ida y vuelta con el equipo. Ese trabajo se sumaba a la redacción propiamente dicha: entender qué pedía el cliente potencial, encontrar la información, comprobar que seguía siendo cierta, decidir qué podíamos comunicar y devolver después la respuesta al formato esperado.

No era una situación excepcional. Cada nuevo cliente potencial mandaba su propio documento. Una vez era una hoja de cálculo; otra, un archivo Word o un PDF. Los repositorios de referencia cambiaban, las frases también, pero volvían los mismos asuntos.

Y sin embargo cada vez teníamos la impresión de empezar otra vez. Contestar llevaba entre tres y seis horas y ocupaba a

varias personas, sin ninguna garantía de que el cliente potencial se convirtiera en cliente. Después el documento salía y el trabajo quedaba encerrado en el archivo enviado. Mi primer objetivo era, pues, sencillo: cuando llegara el cuestionario siguiente, no quería volver a empezar de cero.

Una respuesta verosímil, pero falsa

La solución más evidente consistía en guardar los cuestionarios antiguos y reutilizar sus respuestas.

Descubrimos pronto por qué no bastaba.

Algunos documentos recibidos ya traían respuestas. Podían venir de una plantilla, de otro proveedor o de un intercambio anterior. En uno de ellos, una respuesta decía que la certificación ISO 9001 era de 2017. La frase parecía perfectamente aprovechable. Era precisa, ya redactada y colocada frente a la pregunta correcta.

La fecha correcta era 2018.

El error se reducía a un año, pero dejaba a la vista un problema mucho mayor. Una respuesta antigua se parece a un conocimiento. Tiene una forma afirmativa y tranquilizadora.

Cuando una IA se la encuentra en un documento, puede recogerla sin esfuerzo. Y sin embargo su presencia no dice quién la escribió, ni en qué se apoya, ni si describe realmente nuestro producto.

Entendí entonces que no debía construir una biblioteca de respuestas que copiar. Había que construir una memoria de lo que sabíamos responder de verdad.

El cuestionario recibido conservaba un valor: mostraba qué quería saber el cliente. Pero por sí solo no podía probar la respuesta. Esa distinción se convirtió en una regla del Contexto: el documento del cliente guarda la pregunta; la respuesta tiene que seguir ligada a una información que hayamos comprobado.

Guardar los hechos en lugar de los formularios

Empezamos a extraer de los cuestionarios lo que pudiera servir al siguiente.

Las preguntas ya no se ordenaban solo por cliente o por número. Las agrupábamos por asunto: copias de seguridad, autenticación, cifrado, subcontratistas, alojamiento, continuidad del servicio. Una pregunta llamada «14.6.l» en un cliente potencial podía así juntarse con una pregunta llamada «CL-072» en otro si ambas trataban del mismo asunto.

Para cada uno de esos asuntos, el Contexto guardaba los hechos validados y su origen. La respuesta podía apoyarse en una documentación oficial, en una configuración comprobada, en una prueba o en la confirmación del equipo competente. Guardábamos también la fecha de validación y los límites de lo que podíamos anunciar.

Ese trabajo no siempre producía una respuesta completa. También sacaba a la luz los huecos.

Cuando faltaba una información, el sistema tenía que decirlo. Preparaba una pregunta precisa para el equipo implicado e indicaba qué líneas del cuestionario permitiría completar esa

respuesta. Una información ausente dejaba así de esconderse detrás de una frase plausible. Se convertía en una acción identificable.

Separábamos por último lo que servía para comprobar una respuesta de lo que podía enviarse al cliente. El detalle de una configuración podía ser indispensable para quien revisaba dentro y no tener sitio en el documento público. El Contexto preparaba, pues, la formulación destinada al cliente potencial y conservaba al lado las pruebas y las reservas necesarias para aprobarla.

Poco a poco cambió el objetivo. Ya no se trataba solo de terminar el cuestionario presente. **Cada respuesta comprobada debía reducir el trabajo necesario para los siguientes.**

El cuarto cuestionario

El cuarto cuestionario tenía ciento quince preguntas, muchas más que el primero.

Esta vez no partíamos de un conjunto de archivos dispersos. El Contexto ya conocía los asuntos recurrentes y sabía dónde encontrar los hechos que les correspondían. Noventa y seis respuestas completas pudieron producirse directamente a partir de la memoria acumulada y de las fuentes del producto, sin nueva recogida.

El sistema clasificó también el resto del documento. Algunas respuestas eran sólidas, pero pedían una última

comprobación. Otras seguían siendo inciertas. Unas pocas no podían darse con la información disponible.

Sobre todo, los huecos habían cambiado de naturaleza. Ya no pasábamos horas buscando al azar quién sabía qué. Los puntos aún abiertos correspondían a decisiones comerciales — por ejemplo la duración de un compromiso — o a confirmaciones técnicas puntuales. El trabajo restante era visible, asignable y limitado.

3

Ese resultado me permitió ver qué significa de verdad contestar «automáticamente» a un cuestionario. No se trata de pedirle a una IA que rellene todas las casillas con aplomo. Se trata de preparar de inmediato lo que se pueda, de mostrar cuánta confianza merece cada respuesta y de aislar con claridad lo que sigue exigiendo un juicio humano.

Diez minutos no significa cien por cien de certeza

En mi uso, la preparación de un cuestionario ha pasado de tres a seis horas a unos diez minutos para producir el 80-90 % del documento.

Esa cifra podría dar una impresión engañosa. No significa que el 80-90 % de las respuestas se vuelvan al instante seguras y listas para comprometer a la empresa. Significa que el documento está en buena parte preparado: las respuestas conocidas están puestas, las fuentes recuperadas, los puntos frágiles señalados y la información que falta convertida en peticiones precisas.

La relectura humana sigue siendo indispensable. Se dirige, eso sí, a los lugares adecuados. En vez de dedicar el tiempo a hurgar en archivos antiguos y a molestar otra vez a las mismas personas, quien relee puede comprobar los compromisos, decidir qué se divulgará y concentrarse en las respuestas nuevas o inciertas.

Así que la ganancia más importante no es solo pasar de varias horas a unos minutos. Es el cambio de naturaleza del trabajo.

El primer cuestionario había exigido una investigación. Los siguientes se beneficiaron de todo lo que esa investigación había permitido aprender. Un error de fecha se convirtió en una regla de prudencia. Una confirmación obtenida de un equipo se convirtió en un hecho reutilizable. Una pregunta sin respuesta se convirtió en un hueco conocido y no en un olvido.

El cliente potencial sigue recibiendo su documento.

La empresa, en cambio, ya no pierde lo que aprendió al prepararlo.

PARA RECORDAR — UNA RESPUESTA VALE POR SU PRUEBA

Una respuesta antigua se parece a un conocimiento: es afirmativa, está bien formada y calla su origen y su fecha. Conservar los hechos comprobados y su procedencia, y no los formularios ya rellenos. Un hueco explícito vale más que una frase plausible: se convierte en una pregunta dirigida a alguien.

Escribir un artículo a partir de un pensamiento aún informe

Los artículos que publico en mi blog rara vez empiezan delante del ordenador, ante una página en blanco cuidadosamente preparada.

Una idea aparece después de una dificultad, de una herramienta probada, de una conversación o de un cambio de opinión. A veces llega andando, en el transporte público o en mi sofá. Noto que merece un texto, pero todavía no conozco ni su enfoque exacto ni el orden en que habrá que contarla.

En ese momento suelo sacar el móvil.

No lo uso solo para grabar una nota de voz o dictar un texto que copiaré más tarde. **Abro una conversación con la IA.**

Me escucha, me responde y me da réplica mientras pienso. Esa conversación va al mismo entorno de trabajo que volveré a encontrar en el ordenador. Puedo, pues, empezar lejos de mi mesa sin crear una nota suelta que luego habrá que trasladar y reinterpretar.

Descargar un pensamiento mientras se forma

Creo un Contexto con un objetivo sencillo: escribir el artículo. Después le doy todo lo que tengo en la cabeza.

A veces durante dos horas hablo de un primer punto, luego de otro, y vuelvo al primero. Cuento una experiencia, salto a una

conclusión, añado un detalle, abro una vía nueva y dejo la anterior un rato. Puedo repetir la misma idea cuatro veces porque me vuelve con formas distintas. Puedo también afirmar algo y decir lo contrario una hora después porque mi pensamiento ha cambiado durante la conversación.

Ese desorden no es un accidente que hubiera que evitar antes de empezar. Suele ser la forma real de una idea cuando todavía está naciendo.

Hablar con la IA me permite seguir sin imponer una estructura demasiado pronto. Una respuesta me hace concretar un recuerdo. Una pregunta revela que acabo de saltarme un paso. Una reformulación me enseña que lo que creía evidente todavía no lo es. Corrijo, matizo y salgo en otra dirección.

El Contexto conserva esas palabras como fuente de sesión. No se queda solo con las frases más limpias. Guarda también los retrocesos, las vacilaciones y las contradicciones que permitirán ver cómo se ha desplazado el pensamiento.

La IA acerca después las cuatro formulaciones de una misma idea en vez de tratarlas como cuatro ideas distintas. Señala dos afirmaciones incompatibles, recupera el matiz que llegó una hora más tarde y junta un ejemplo con el principio que ilumina. **No hace desaparecer el desorden: lo vuelve trabajable.**

Esa primera etapa me libra de una elección prematura.

Todavía no tengo que decidir qué frase abrirá el artículo, si tal

idea merece una sección o en qué orden contar los hechos. Lo que busco primero es vaciar lo que tengo en la cabeza, sin dejar escapar las ideas que solo la atraviesan.

El Contexto, eso sí, sabe por qué está ahí ese material. No recibe una conversación sin destino. Recibe el material todavía desordenado de un artículo por construir.

Usar la IA como compañera de maduración

Una vez depositado ese primer material, le pregunto a la IA qué falta.

Detecta una afirmación demasiado rápida, un supuesto nunca explicado, un pasaje que solo se sostiene si el lector ya conoce mi experiencia. Pide un ejemplo. Acerca dos ideas formuladas en momentos distintos. A veces propone una vía que parece plausible y que rechazo porque no corresponde a lo que viví.

Respondo, concreto, corrijo. Esa conversación todavía no produce el artículo. Densifica su material.

El papel de la IA es importante, pero su sitio sigue siendo claro. Dispone de todo lo que he volcado y busca ahí tensiones más deprisa que yo. En cambio ignora qué quiero defender de verdad mientras no lo haya formulado. Ella abre la pregunta; yo decido si esa pregunta merece mover el texto. Ese diálogo acorta el tiempo necesario para tropezar con una debilidad. Solo, habría descubierto dos días después que faltaba un ejemplo o que mi conclusión se apoyaba en un

paso nunca explicado. La IA provoca ese hallazgo mientras la idea aún es fácil de rehacer.

Añadir fuentes cuando la experiencia ya no basta

Mis artículos parten a menudo de lo que he vivido. El material exterior conserva, sin embargo, su papel:

- una referencia le da nombre a lo que he observado;
- una documentación comprueba un funcionamiento;
- otro autor defiende una posición vecina, o muestra un límite con el que yo no me había encontrado;
- una búsqueda adicional revela que mi experiencia solo cubre un caso particular.

Esos documentos entran en el Contexto como **fuentes documentales**. Su papel se distingue del de mis primeras palabras. La **fente de sesión** conserva lo que afirmo y lo que entiendo de mi experiencia. Las **fuentes documentales** aportan un material que puedo confrontar con esa experiencia.

El artículo puede entonces distinguir lo que he observado, lo que una fuente establece y lo que sigue siendo una interpretación. Esa distinción protege el texto de una tentación frecuente: presentar una experiencia personal como una regla general.

Convertir las fuentes en ideas reutilizables

A estas alturas el Contexto no pasa directamente de las fuentes al guion del artículo. Primero convierte el material bruto en **notas atómicas**.

Cada nota asienta una idea autónoma salida de la conversación o de un documento:

- una afirmación que quiero defender;
- un ejemplo que le da forma concreta;
- un matiz que limita su alcance;
- una objeción a la que el artículo tendrá que responder;
- un hecho exterior que confirma o contradice mi experiencia.

La nota conserva su origen. Puedo así entender la idea sin releer toda la conversación y volver luego a la fuente si tengo que comprobar una formulación o recuperar su contexto.

Esa transformación cambia la manera de preparar el artículo.

- La conversación seguía el movimiento de mi pensamiento; las notas agrupan ahora lo que habla de la misma idea, aunque esos elementos aparecieran con una hora de diferencia o en dos fuentes distintas.
- Una nota central puede hacer aparecer el enfoque del artículo.
- Otras le aportan ejemplos, pruebas, objeciones o consecuencias.

Esas notas no son todavía ni párrafos ni un guion prefabricado. **Siguen siendo neutrales frente al entregable:**

la misma idea podrá alimentar este artículo, una presentación u otro trabajo.

Pasar por el guion antes de redactar

Cuando el material resulta suficiente, pido un guion.

Esa etapa me impide pasar directamente de una conversación rica a un texto que se limitara a seguir su orden. Una discusión avanza por retrocesos, correcciones y asociaciones. Un artículo tiene que llevar a un lector que no conoce ni la discusión ni su historia.

El guion fija esa progresión. Indica qué hay que entender primero, dónde colocar el ejemplo principal, qué objeción tratar y qué puede afirmar realmente la conclusión.

La IA propone una estructura a partir del material disponible. La muevo, la aprieto, rechazo las partes que le dan demasiada importancia a una vía secundaria. Dos o tres intercambios bastan a menudo para obtener una dirección clara.

La redacción viene después.

A estas alturas la IA no crea el texto a partir de una instrucción vaga. Trabaja con un corpus de ideas y de temas extraídos de mi experiencia contada, con respuestas a preguntas, con las fuentes y con un guion validado. Agrupa las repeticiones, recoloca los ejemplos y convierte un habla sinuosa en prosa legible.

Releo el resultado buscando tres desviaciones concretas:

- un matiz que ha desaparecido;
- un pasaje que se ha vuelto demasiado afirmativo;

- una formulación que traiciona la idea.

Las corrijo y decido después la versión publicada.

Lo que el Contexto aporta a un trabajo corto

Un artículo puede escribirse en uno o dos días. No parece necesitar la misma memoria que un producto seguido durante años o que un libro de decenas de capítulos.

Justamente ese caso ligero permite ver que la duración no es lo único que está en juego.

3

El Contexto mantiene juntas las primeras palabras, las fuentes añadidas, las preguntas hechas, el guion elegido y el texto completo. Si el trabajo se interrumpe, recupero su estado. Si una frase se vuelve discutible, puedo volver a su origen. Si una idea aparecida durante la escritura desborda el artículo, puede asentarse aparte y alimentar un trabajo futuro.

No todo ese material merece por fuerza conservarse indefinidamente. Un Contexto pequeño puede alcanzar su objetivo, producir su artículo y luego archivarse. Su valor está en la continuidad que le dio al trabajo mientras se formaba y en el rastro que deja si más adelante hay que entender cómo el texto tomó esa dirección.

La publicación misma puede desencadenar otras tareas: preparar un resumen, adaptar el texto a varios idiomas o producir lo necesario para ponerlo en línea. Esas operaciones prolongan el mismo resultado, bajo reglas que el Contexto ya conoce.

El movimiento completo sigue siendo sencillo: una experiencia se vuelve palabra, esa palabra se encuentra con preguntas y fuentes, el material se vuelve un guion, y después el guion se vuelve un texto.

La IA acelera cada transformación, y el Contexto sostiene el hilo que lleva de una a otra.

Nada de ese movimiento pertenece en exclusiva a la escritura larga. Empieza en cuanto una idea merece algo más que una nota de voz olvidada en un móvil.

PARA RECORDAR — EL DESORDEN ES MATERIAL

Un pensamiento que nace se repite, se contradice y vuelve atrás: esa es su forma normal, y no un defecto que corregir antes de empezar. El Contexto conserva esas palabras como fuente de sesión y después las vuelve trabajables. El guion se intercala entre la conversación y el texto, porque un lector no sigue el orden de una discusión.

Madurar un tema de producto en medio del caos

Los dos capítulos anteriores siguieron un solo tema cada uno, de principio a fin. Este capítulo describe cómo es de verdad un día cualquiera de un product manager: varios temas, peticiones y ritmos avanzan a la vez.

Soy el único product manager de mi empresa. Eso significa que buena parte de la información ligada al producto acaba pasando por delante de mí.

En un mismo día, un cliente dice una frase que me parece importante.

- Cabe en unas pocas palabras.
- Todavía no sé si revela un problema compartido por otros usuarios o solo una dificultad particular, pero no quiero perderla.

Un poco más tarde recojo un informe técnico sobre cómo evoluciona una cuestión del oficio.

- Esta vez el material ya no cabe en una frase, sino en un PDF de varias decenas de páginas.
- Todavía no sé qué me será útil ni si ese documento confirmará lo que observo sobre el terreno.

Después los desarrolladores me hablan de una funcionalidad en construcción.

- Hay que responder a una pregunta, concretar un comportamiento o volver sobre una hipótesis tomada unas semanas antes.

Mientras tanto, el CPO quiere hacer balance de dos apuestas ligadas a otra parte del producto.

Una conversación con el soporte revela por fin que falta una documentación funcional.

- Parte de esa documentación solo existe todavía en mi cabeza.

Al final del día he pasado por tres o cuatro temas distintos (o más). He recibido una cita de unas pocas palabras, un documento entero, una pregunta sobre un desarrollo en curso, una petición de arbitraje y una necesidad de documentación.

No es un día especialmente malo: es un día corriente.

Entradas que no tienen ni el mismo tamaño ni el mismo destino

El caos del product management no viene solo de la cantidad de información. Viene de su diferencia de naturaleza.

Cada una de esas cinco entradas mira a un momento distinto de la vida del producto:

- una frase de un cliente es una señal débil, de la que aún no sé si anuncia otras;
- un PDF da una visión general de un oficio;
- una discusión con los desarrolladores trata de lo que estamos construyendo ahora;

- una petición del soporte afecta a lo que ya existe y hay que explicar;
- una pregunta del CPO obliga a comparar varios futuros posibles.

Esos elementos no están hechos para entrar en un documento único.

- No hablan todos del mismo tema y no se sitúan en el mismo momento de la vida del producto.

3

Si los reúno sin más en una lista de notas, conservo la información, pero aplazo el trabajo de comprensión al día en que lo necesite.

Los Contextos me permiten mantener varios hilos en paralelo.

- La cita se suma al tema al que parece pertenecer, aunque su significado siga siendo incierto.
- El PDF se conserva con su origen y puede analizarse para extraer las pocas ideas (**notas atómicas**) realmente útiles.
- La conversación con los desarrolladores se suma al trabajo sobre la funcionalidad en curso.
- La falta señalada por el soporte se convierte en un trabajo por hacer en lugar de quedarse en una observación oída entre dos reuniones.

No todo se transforma enseguida en conocimiento:

- la frase del cliente quizá siga siendo ambigua,
- el informe técnico poco pertinente,
- la intuición desmentida a la semana siguiente.

El Contexto no le da inteligencia a lo que entra. Le da un sitio, un origen y una posibilidad de ser recuperado en el momento oportuno.

El Contexto contiene el caos sin pretender abolirlo.

Mientras entra información, las peticiones de salida continúan

El día no se reduce a recibir y clasificar material: también hay que producir.

El soporte me pide, por ejemplo, una documentación funcional sobre una funcionalidad en la que aún estamos trabajando.

- Hay que explicar cómo funciona, concretar los casos importantes y darles a quienes acompañan a los clientes lo necesario para responder correctamente.

El mismo día puedo querer terminar un artículo para el blog de la empresa.

- Ese artículo no trata necesariamente de la misma funcionalidad.
- Puede, en cambio, tratar del mismo tema de oficio: el trabajo al que responde el producto, una dificultad recurrente de los usuarios o un cambio en sus prácticas.

Los dos documentos recurren entonces a parte del mismo conocimiento sin contar lo mismo.

- La documentación destinada al soporte explica el producto tal como funciona.
 - Tiene que ser precisa, operativa y ajustada a las preguntas de los usuarios.

- El artículo desarrolla una experiencia, un problema de oficio o una reflexión.
 - Puede movilizar algunas de las mismas ideas, pero las organiza para otro lector y con otro fin.

No necesito reconstruir dos veces todo lo que sé del tema.

- El Contexto conserva un material intermedio (las **notas atómicas** y las **notas temáticas**) que no es todavía ni la documentación ni el artículo.
- A partir de ese material, la IA puede preparar varios montajes: uno centrado en el uso del producto, el otro en la cuestión de oficio más amplia.

Releo, claro, los dos resultados.

- Una documentación para el soporte compromete cómo se entiende el producto.
- Un artículo compromete mi punto de vista.

Pero dedico mi tiempo a comprobar el fondo y la orientación, en vez de recuperar los mismos elementos y escribirlos otra vez desde cero.

Ese movimiento puede ocurrir en el mismo día. Mientras llega información nueva, un conocimiento ya organizado permite producir los documentos que la empresa o yo necesitamos.

Una frase pequeña puede convertirse en un tema de producto

La cita oída por la mañana no desencadena necesariamente una decisión.

- Tomada sola, puede no probar nada.
- Otro cliente expresará quizá una dificultad parecida unas semanas después.
- Una entrevista aportará un matiz.
- El informe técnico recogido ese día le dará quizá nombre al cambio observado.
- Un dato del soporte mostrará que el problema vuelve más a menudo de lo que pensábamos.

Esos elementos llegan en momentos distintos y bajo formas distintas.

- El Contexto permite acercarlos sin borrar su origen.
- Puede señalar
 - que una observación nueva se parece a una señal ya conservada,
 - hacer aparecer una contradicción,
 - preguntar qué permitiría decidir entre dos interpretaciones.

Poco a poco, una frase suelta puede convertirse en un tema que merece trabajarse. Para entonces ya dispongo de parte de su historia: las primeras señales, los documentos que la iluminan, lo que los confirma, lo que los contradice y las preguntas aún abiertas.

El Contexto ayuda a que ese material madure.

- No decide que el problema es importante ni elige la solución por mí.

La apuesta de producto sigue siendo responsabilidad mía.

- En cambio, puedo tomar esa decisión con algo más que un recuerdo vago o una serie de documentos dispersos.

El mismo sistema acompaña después el trabajo:

- añadir entrevistas,
- confrontar una hipótesis con los hechos,
- preparar escenarios,
- producir un argumentario para el CPO,
- redactar especificaciones,
- ...

Una señal pequeña captada en mitad de un día puede así seguir viva lo bastante para convertirse en una decisión real.

Una idea abandonada puede seguir madurando

No todas las exploraciones llegan a producción.

He trabajado en usos de la inteligencia artificial que parecían interesantes pero que la tecnología disponible no permitía todavía realizar bien.

- La idea no era necesariamente mala.
- Lo era el momento.

En un funcionamiento clásico, la exploración habría producido unos documentos y luego habría desaparecido en una carpeta tras la decisión de abandonarla.

- Uno o dos años después habría hecho falta recuperar esos archivos, acordarse de las hipótesis de partida y entender por qué nos habíamos parado.

Con un Contexto, la exploración puede archivarse con su recorrido: las señales iniciales, los escenarios imaginados, los intentos, los límites encontrados y la razón de la parada.

- Cuando la tecnología evoluciona, puedo retomar ese trabajo sin convertir un antiguo resultado negativo en una idea definitivamente mala.

Recupero lo que queríamos hacer, pero también por qué no funcionaba. Puedo entonces comprobar si el obstáculo sigue ahí, conservar lo que siga valiendo y seguir la reflexión a partir de ese punto.

La maduración de un tema de producto no es, pues, siempre un trayecto continuo.

- Puede empezar con una frase, enriquecerse durante meses, interrumpirse y luego reanudarse cuando el producto, el mercado o la tecnología han cambiado.

Hacer aparecer orden sin frenar el día

Mi día sigue estando fragmentado.

- Los clientes no coordinan sus observaciones con las peticiones del soporte.
- Los desarrolladores no eligen el momento en que acabo de terminar la lectura de un PDF.
- Las decisiones por preparar no se detienen mientras escribo una documentación o un artículo.

El Contexto no convierte esa realidad en una sucesión perfecta de tareas.

Permite que cada fragmento conserve una continuidad.

- Una frase pequeña y un documento largo pueden entrar sin recibir el mismo tratamiento.
- Varios temas pueden avanzar sin confundirse.
- Un conocimiento ya trabajado puede servir a varios entregables sin quedar encerrado en el primero.
- Una idea interrumpida puede esperar sin perder toda su historia.

3

Así es como lo uso en product management: no para que mi día se agite menos, sino para evitar que su movimiento disperse lo que aprendo.

El caos sigue entrando.

Lo que ya no hace es salir en forma de pérdida.

Estos tres usos tienen un punto en común que aún no he nombrado: cada uno termina con algo que sale — una respuesta enviada, un artículo publicado, una decisión tomada.

La parte siguiente mira lo que queda una vez que el entregable se ha ido.

PARA RECORDAR — VARIOS HILOS EN PARALELO

Una frase de un cliente y un PDF de cincuenta páginas piden un tratamiento y un destino distintos. El Contexto le da a cada fragmento un sitio, un origen y una posibilidad de ser recuperado en el momento oportuno. Una exploración abandonada conserva sus razones de parada: vuelve a ser retomable cuando la tecnología o el mercado cambian.

Parte 4

Lo que los entregables dejan detrás

Cada uno de los tres usos anteriores termina con un entregable remitido. Esta parte mira lo que queda una vez que se ha ido.

Sirve de hilo una misma pregunta, hecha dos veces con seis meses de diferencia. La primera vez, la respuesta cabe en una celda de hoja de cálculo. La segunda, esa respuesta existe todavía pero ya no basta: lo que falta es el razonamiento que la había producido, y las relaciones entre los documentos que lo sostenían.

4.1

La respuesta cabe en una celda, el razonamiento no

Volvamos a uno de los cuestionarios de la parte anterior. La pregunta llega en la celda D47 de una hoja de cálculo:

| ¿Sus copias de seguridad están cifradas en reposo?

La respuesta final ocupará tres líneas. Para escribirlas, sin embargo, tengo que reunir varios elementos:

- nuestra documentación describe cómo funcionan las copias de seguridad en general;
- un archivo de configuración indica el ajuste realmente activo;
- el equipo de Ops confirma cómo se aplica ese ajuste en producción;
- el departamento jurídico aclara qué podemos comunicar a un cliente potencial.

Paso de la hoja de cálculo a la documentación, de la documentación a la configuración, y luego de un intercambio con el equipo de Ops a una decisión jurídica. Vuelvo por fin a la celda D47 y escribo ahí la respuesta.

El documento sale.

Tres líneas al final de una cadena

La celda contiene el resultado del trabajo. No muestra el camino que lo produjo.

No dice:

- qué documentación consulté;
- qué ajuste comprobé;
- quién confirmó que se aplicaba;
- qué formulación se autorizó;
- qué reservas se sacaron del texto enviado al cliente potencial.

Esos elementos existen, sin embargo. Han determinado cada una de las tres líneas y, mientras respondo, su relación está clara en mi cabeza.

Una vez enviado el documento, esa unidad desaparece. La hoja de cálculo conserva la respuesta, pero no el razonamiento que permitía fiarse de ella.

Cada herramienta hizo su trabajo

El problema no viene de un mal orden.

- La documentación está donde tiene que estar, en el espacio documental.
- La configuración sigue en el producto.
- La respuesta del equipo permanece en su hilo de discusión.
- La decisión jurídica tiene su propio soporte.
- La celda D47 está en su sitio en el documento del cliente potencial.

Cada herramienta conservó el objeto que se le había confiado. Ninguna tenía por función guardar el razonamiento entero. Para reconstruirlo hay que saber qué elementos buscar, dónde buscarlos y qué papel desempeñó cada uno. Ese conocimiento sigue en manos de quien preparó la respuesta.

El entregable aplanar lo que lo produjo

Un entregable tiene que presentar un resultado adaptado a su destinatario.

- El cliente potencial espera una respuesta clara, no el diario completo de la investigación hecha para obtenerla.

Esa simplificación es necesaria. Se vuelve costosa cuando el entregable es el único rastro del trabajo.

El resultado final siempre es más corto que el camino que lleva a él: la respuesta enviada, el artículo publicado o la decisión de producto. Si solo queda el resultado, el trabajo siguiente tiene que rehacer parte de ese camino.

Darle una dirección al razonamiento

El Contexto no sustituye ni a la hoja de cálculo, ni a la documentación, ni a la configuración, ni a los intercambios con los equipos.

- Le da una dirección común al trabajo que los une.

La pregunta D47, los elementos consultados, las validaciones obtenidas y la respuesta producida pertenecen al mismo asunto. Pueden conservar su forma y su ubicación de origen y seguir ligados dentro del Contexto.

La celda sigue cumpliendo su función: contestarle al cliente potencial. El Contexto cumple otra: conservar lo que permitirá entender, comprobar y retomar esa respuesta.

Seis meses después, esa diferencia se vuelve decisiva.

PARA RECORDAR — EL ENTREGABLE NO CONSERVA EL RAZONAMIENTO

Una respuesta puede ocupar tres líneas y depender aun así de varias fuentes, validaciones y reservas. Las herramientas conservan cada una su objeto; quien sigue cargando con las relaciones entre ellas es la persona. El Contexto le da una dirección al trabajo que une esos elementos, sin sustituir a sus herramientas de origen.

Seis meses después, la respuesta ya no basta

Seis meses después de la celda D47, un nuevo cliente potencial hace la misma pregunta sobre el cifrado de las copias de seguridad.

- Recupero la respuesta antigua en diez segundos.
- Es clara, está bien escrita y lista para copiarse.

Y sin embargo me quedo delante de ella sin poder usarla.

Bastan tres preguntas para bloquearme:

- ¿de dónde venía esa respuesta: de la documentación, del ajuste observado o de lo que el equipo de Ops me había explicado aquel día?
- ¿el jurídico había validado esa formulación exacta o una versión más prudente?
- ¿el ajuste sigue siendo el mismo hoy?

La respuesta ha sobrevivido; lo que permitía fiarse de ella ha desaparecido.

Encontrar no es todavía retomar

La celda antigua me devuelve un texto. No me devuelve:

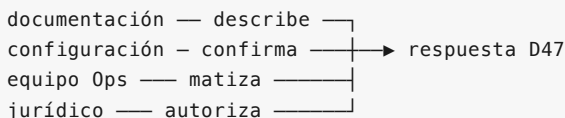
- su nivel de prueba,
- sus límites,
- ni las comprobaciones necesarias antes de volver a usarla.

Sin otra memoria, tengo que rehacer toda la investigación del capítulo anterior, en orden inverso. No empiezo del todo de cero, puesto que ya tengo una respuesta. Pero empiezo sin saber qué parte del trabajo sigue valiendo.

Es el error de la certificación fechada en 2017 en vez de 2018, visto esta vez desde el lado de nuestras propias respuestas: la frase es nítida, viene de nosotros y su prueba se ha vuelto invisible.

Recuperar el papel de cada elemento

En el Contexto, la respuesta antigua no se queda sola. Conserva las relaciones que necesitaba el razonamiento que la produjo:



Los verbos cambian lo que puedo hacer con cada elemento.

- La documentación describe el funcionamiento general; por sí sola no prueba el ajuste activo.
- La configuración confirma ese ajuste en un momento dado; no decide qué puede comunicarse.
- El equipo de Ops aporta un matiz que viene de la explotación real.
- El jurídico autoriza una formulación destinada al cliente potencial.

Ya no recupero solo cuatro objetos cercanos a una respuesta. Recupero la función que cada uno cumplía.

Esa precisión impide que una fuente pruebe más de lo que establece realmente. Muestra también qué comprobación hay que rehacer cuando ha pasado el tiempo.

Comprobar lo que puede haber cambiado

El Contexto no convierte la respuesta antigua en verdad permanente: prepara su comprobación.

- La documentación puede seguir siendo válida mientras la configuración ha cambiado.
- La configuración puede seguir igual mientras la formulación autorizada debe revisarse.
- Un matiz aportado por el equipo de Ops puede haber desaparecido tras una evolución del producto.

Las relaciones hacen visibles esas diferencias. Puedo repasar los elementos que pueden haber cambiado sin rehacer toda la investigación al azar.

La respuesta final quizá sea idéntica a la de hace seis meses. La diferencia está en que sé por qué puedo seguir usándola.

Conservar las relaciones que sirven al trabajo

No se trata de enlazar todos los documentos entre sí.

- Dos archivos pueden compartir una palabra sin participar en el mismo razonamiento.

Una relación merece conservarse cuando cambia la manera de entender, comprobar, decidir o producir.

- En el caso de D47, bastan cuatro relaciones: describir, confirmar, matizar y autorizar.

Permiten responder a preguntas concretas:

- ¿en qué se apoya esta respuesta?
- ¿qué parte hay que comprobar hoy?
- ¿quién había validado lo que podíamos comunicar?
- ¿qué dejaría esta respuesta sin valor?

Así que el Contexto no busca fabricar un mapa total de todo lo que sé. Conserva el tejido necesario para un trabajo determinado.

Abrir la máquina

El Libro I ha mostrado cómo nació el Contexto, qué cambia en la relación con la IA y cómo acompaña varios usos. El caso D47 deja ver lo que queda por entender.

- ¿Cómo conservar la misión del trabajo sin repetirla en cada sesión?
- ¿Dónde colocar las fuentes, las ideas, las decisiones y los entregables para que conserven papeles distintos?
- ¿Cómo cargar lo que es útil ahora sin meter todo el material en la conversación?

El Libro II va a abrir esa máquina. Seguirá el recorrido completo de un material, desde su llegada a un Contexto hasta su transformación, lo que produce y su vuelta al trabajo.

PARA RECORDAR — RECUPERAR LO QUE PERMITE FIARSE

Recuperar una respuesta no basta: hay que recuperar lo que todavía permite usarla. Una relación nombrada indica el papel preciso de una fuente, de una validación o de una reserva. El Contexto prepara la comprobación mostrando qué sostiene la respuesta y qué puede haber cambiado.

LIBRO II

Construir y mantener vivo el Contexto

- 5 Bajo el capó de un Contexto
- 6 El Contexto en su ecosistema

Parte 5

Bajo el capó de un Contexto

Hasta aquí el Contexto se ha descrito por lo que permite hacer. Esta parte lo abre.

Presenta primero su árbol completo y sigue después una sesión de trabajo de principio a fin: abrir y recuperar el punto en que está el trabajo, hacer entrar material, convertirlo en ideas, producir un entregable, arbitrar y guardar para que la sesión siguiente vuelva a empezar sin reconstruir nada. El ejemplo que atraviesa estos seis capítulos es enteramente ficticio: entender por qué algunos técnicos de intervención entregan partes de intervención incompletos, antes de decidir si el producto debe cambiar.

5.1

El mapa completo

En la parte 2 dejé el Contexto dedicado a la calidad de los partes de intervención:

- tres fuentes,
- dos notas atómicas,
- una nota temática,
- una guía para la próxima entrevista.

Entonces miraba lo que ese Contexto permitía entender. Ahora abro la carpeta misma.

Este es el árbol completo de la versión que uso en este libro:

5

```
contexte-qualite-comptes-rendus/
├─ README.md           navegación general, generada
├─ offline.md          el buzón de fuera de sesión
├─
├─ 00-frame/           lo que marca una dirección duradera
│   ├── README.md      navegación de la zona, generada
│   ├── mission.md     el objetivo, la pregunta, los límites
│   ├── directives.md  las reglas propias de la misión
│   └─ loading.md      lo que se leerá al abrir
├─ 01-state/           el presente del trabajo
│   ├── README.md      navegación de la zona, generada
│   ├── state.md        lo establecido, dudas, deudas
│   ├── synthesis.md    lo que se ha vuelto duradero
│   ├── todo.md         las acciones pendientes
│   ├── decisions.md    las opciones que estructuran el Contexto
│   ├── contradictions.md lo que sigue en tensión
│   ├── handoff.md      cómo retomar el trabajo
│   ├── checkpoints.md  la serie compacta de los estados
│   └─ health.md        el resultado del último control
├─ 02-history/         el rastro factual y fechado
│   ├── README.md      el papel de este rastro, generado
│   └─ YYYY-MM-DD.md   los hechos del día
```

└─ 03-sources/	lo que se ha recibido o dicho
└─ README.md	la lista de fuentes, generada
└─ ingestion-log.md	qué se leyó, qué produjo
└─ documents/	
└─ README.md	la lista de documentos, generada
└─ raw/	los archivos originales
└─ fulltext/	su texto íntegro utilizable
└─ SRC00001-...md	identidad, procedencia, estado
└─ sessions/	
└─ README.md	la lista de sesiones, generada
└─ SESSION-...md	lo dicho durante el trabajo
└─ 04-notes/	el conocimiento sacado de las fuentes
└─ README.md	el mapa de las notas, generado
└─ atomic/	una idea por nota – AT000001-...md
└─ thematic/	un ángulo que enlaza – THM00001-...md
└─ glossary/	un término fijado – GL000001-...md
└─ decision-records/	un arbitraje – DRE00001-...md
└─ 05-outputs/	lo que se produce para un uso
└─ README.md	la lista de resultados, generada
└─ _draft/	los entregables en curso
└─ questions-entretien.md	una guía para continuar
└─ note-de-cadrage.md	una síntesis para decidir
└─ presentation-equipe.md	lo ya sabido, reformulado

Los números imponen un orden de lectura, no una jerarquía de valor:

- el marco y el estado orientan el trabajo;
- la historia permite controlar cómo se ha desarrollado;
- las fuentes guardan el material, las notas lo convierten en conocimiento;
- los entregables le dan una forma para un destinatario.

Seis zonas, seis funciones

00-frame/ responde a la pregunta «¿por qué estoy aquí y bajo qué reglas?». Esta zona cambia pocas veces, y con razón:

modificar la misión transformaría mi manera de juzgar todas las fuentes, todas las tareas y todos los resultados.

```

contexte-qualite-comptes-rendus/
├─ 00-frame/                lo que marca una dirección duradera
|   └─ README.md            navegación de la zona, generada
|   └─ mission.md           el objetivo, la pregunta, los límites
|   └─ directives.md        las reglas propias de la misión
|   └─ loading.md           lo que se leerá al abrir

```

01-state/ responde a «¿dónde estoy ahora?». Esta zona se mueve con el trabajo y me permite distinguir un hecho establecido de una hipótesis, una acción futura de una decisión ya tomada, una dificultad pasajera de una contradicción que merece seguir a la vista.

```

contexte-qualite-comptes-rendus/
├─ 01-state/                el presente del trabajo
|   └─ README.md            navegación de la zona, generada
|   └─ state.md             lo establecido, dudas, deudas
|   └─ synthesis.md         lo que se ha vuelto duradero
|   └─ todo.md              las acciones pendientes
|   └─ decisions.md         las opciones que estructuran el Contexto
|   └─ contradictions.md    lo que sigue en tensión
|   └─ handoff.md           cómo retomar el trabajo
|   └─ checkpoints.md       la serie compacta de los estados
|   └─ health.md            el resultado del último control

```

02-history/ responde a «¿qué ha pasado realmente?». Esta zona no cuenta la sesión con elegancia: guarda sus hechos útiles, una fuente añadida, una pista abandonada, una decisión tomada, una nota creada. La consulto pocas veces, pero se vuelve decisiva el día en que el estado presente me parece incoherente.

```

contexte-qualite-comptes-rendus/
├── 02-history/                el rastro factual y fechado
│   ├── README.md             el papel de este rastro, generado
│   └── YYYY-MM-DD.md         los hechos del día

```

03-sources/ responde a «¿qué material he recibido?». Un documento de fuera entra congelado; mi propia palabra entra a lo largo de las sesiones. En ambos casos, la fuente sigue separada de lo que el Contexto entenderá de ella.

```

contexte-qualite-comptes-rendus/
├── 03-sources/                lo que se ha recibido o dicho
│   ├── README.md             la lista de fuentes, generada
│   ├── ingestion-log.md      qué se leyó, qué produjo
│   └── documents/
│       ├── README.md         la lista de documentos, generada
│       ├── raw/              los archivos originales
│       ├── fulltext/         su texto íntegro utilizable
│       └── SRC00001-....md    identidad, procedencia, estado
└── sessions/
    ├── README.md             la lista de sesiones, generada
    └── SESSION-....md        lo dicho durante el trabajo

```

5

La ficha SRC00001-....md no contiene ni el documento ni su resumen. Responde a todo lo que necesito saber sobre él:

- de dónde viene y por qué lo añadí;
- dónde encontrar el original y su texto íntegro;
- si la extracción salió bien;
- qué nivel de confianza darle;
- si su lectura ya ha producido notas.

Así que hace visible el estado de la fuente sin tocar el material en sí.

04-notes/ responde a «¿qué he aprendido?».

- Una nota atómica fija una idea.
- Una nota temática enlaza varias ideas.
- El glosario concreta una palabra.
- Un decision record guarda las razones de un arbitraje.

```

contexte-qualite-comptes-rendus/
├─ 04-notes/                el conocimiento sacado de las fuentes
│   ├── README.md          el mapa de las notas, generado
│   ├── atomic/            una idea por nota – AT000001-....md
│   ├── thematic/          un ángulo que enlaza – THM00001-....md
│   ├── glossary/          un término fijado – GL000001-....md
│   └─ decision-records/   un arbitraje – DRE00001-....md

```

05-outputs/ responde a «¿qué voy a hacer con este conocimiento?». Los borradores y los resultados viven ahí, con la forma que espera su público:

- una guía de entrevista;
- un artículo;
- una especificación;
- una presentación;
- una decisión, entre otras formas posibles.

```

contexte-qualite-comptes-rendus/
└─ 05-outputs/              lo que se produce para un uso
    ├── README.md          la lista de resultados, generada
    ├── _draft/             los entregables en curso
    ├── questions-entretien.md una guía para continuar
    ├── note-de-cadrage.md   una síntesis para decidir
    └─ presentation-equipe.md lo ya sabido, reformulado

```

El mundo de las ideas: piezas para varias construcciones

La separación entre 04-notes/ y 05-outputs/ sostiene una de las fuerzas principales del Contexto. Las notas y el glosario forman

ahí un *mundo de las ideas*: un conjunto de piezas de conocimiento, reutilizables sin límite, que ya no son material en bruto y todavía no son un resultado destinado a alguien.

```

03-sources/
  |
  ▼
04-notes/           el mundo de las ideas
├─ AT000001         una idea fijada
├─ AT000002         otra idea fijada
└─ THM00001         una lectura de su relación
    |
    ├──> questions-entretien.md
    ├──> note-de-cadrage.md
    └──> presentation-equipe.md

```

En mi ejemplo, la fricción al rellenar el parte y la necesidad de datos estructurados quedaron fijadas antes de que yo supiera con qué forma se presentarían.

- la guía de entrevista saca de ahí preguntas,
- la nota de encuadre saca de ahí lo que está en juego,
- la presentación saca de ahí un relato dirigido al equipo.

Estos tres entregables no poseen el conocimiento que usan.

- Seleccionan y recomponen una parte para una necesidad, un público y un formato concretos.
- Así que un entregable no agota las notas de las que está hecho: las piezas siguen disponibles para otra construcción.

Sin ese mundo intermedio, cada resultado nuevo me obligaría a reabrir las fuentes, a rehacer los cruces y a reformular lo mismo otra vez.

El Contexto no conserva solo lo que ya se ha producido: construye una capacidad duradera de volver a producir.

No todos los archivos cambian igual

El árbol solo se vuelve de verdad legible cuando sé cómo cambia cada uno de sus elementos.

Algunos archivos se reescriben para mostrar la mejor situación disponible ahora:

- `state.md`;
- `handoff.md`;
- los borradores de `05-outputs/_draft/`.

Otros solo se alargan, para conservar el pasado:

- las fuentes de sesión;
- la historia fechada;
- la síntesis;
- la serie de los checkpoints.

Los documentos recibidos, en cambio, quedan congelados, y una nota sustituida por otra sigue pudiendo consultarse. Los archivos `README.md`, por último, se generan a partir de lo que existe: corregirlos a mano crearía una segunda versión de la realidad, que acabaría separándose.

Esa diversidad explica por qué un solo archivo grande ya no me bastaba. Habría habido que pedirle al mismo texto:

- mostrar el presente,
- conservar el pasado,
- recibir material en bruto,

- fijar ideas,
- llevar un resultado terminado.

Un esqueleto y no una receta que copiar

Este esqueleto no es una documentación colocada al lado del

Contexto: es una carpeta de verdad, que copio.

- Cada archivo llega con las secciones que debe contener y algunas indicaciones, de modo que dos Contextos creados con meses de diferencia comparten las mismas referencias.

Aun así, la estructura sigue siendo adaptable. Un Contexto puede recibir:

- una regla particular,
- un entregable propio de su oficio,
- una fuente de un tipo nunca visto.

Sobre todo, la base común conserva las tres fronteras que mi trabajo necesita:

- lo estable y el presente,
- lo bruto y lo interpretado,
- el conocimiento y su puesta en forma.

Hablar al Contexto

Hasta aquí he mostrado carpetas y archivos. Podría deducirse que tengo que abrir cada una de ellas, saber dónde escribir y mover yo mismo cada información. No trabajo así.

Hablo a la IA:

Abre el Contexto dedicado a la calidad de los partes de intervención.

| *Añade este documento como fuente.*

| *Convierte esta fuente en notas.*

| *Guarda el trabajo.*

Detrás de cada una de estas peticiones hay un procedimiento que escribí una vez y que la IA puede retomar en cada sesión.

- En Claude Code y Codex, ese tipo de procedimiento reutilizable tiene un nombre: un **skill**.

El skill de apertura indica, por ejemplo:

- qué hay que absorber antes de empezar;
- qué archivos leer siempre;
- qué debe quedarse en el disco hasta que la tarea lo exija;
- cómo señalar un problema;
- qué información devolverme para que pueda actuar enseguida.

No necesito dar esos pasos cada vez.

- Solo pido que se abra el `Contexto`.

La conversación lleva mi intención del momento.

- El skill lleva la manera fiable de ejecutar los gestos que se repiten.

Mi sistema tiene así unos cuantos verbos estables:

- crear un `Contexto`;
- abrirlo y recuperar el trabajo en curso;
- añadir una fuente o conservar una sesión;
- convertir las fuentes en notas;

- guardar el estado del trabajo;
- controlar su coherencia;
- archivarlo cuando ha alcanzado su objetivo.

No todos mis intercambios con la IA pasan por un skill. Sigo pensando, preguntando, corrigiendo y arbitrando en lenguaje corriente.

Los skills solo enmarcan las operaciones que exigen respetar siempre los mismos pasos.

- Impiden, por ejemplo, que una fuente se transforme sin guardar su origen o que una sesión termine sin preparar la vuelta al trabajo.

El procedimiento sigue siendo estable, pero el juicio no se vuelve automático. Dos modelos pueden formular una nota de otra manera. Aun así, deben consultar los mismos elementos, respetar las mismas fronteras y hacer los mismos controles.

Compartir los skills no le da a nadie, por tanto, mi Contexto. Le da la misma manera de crear el suyo y de mantenerlo vivo.

No diré más sobre su mecánica.

Los skills no son un invento ni una particularidad del Contexto. Claude Code y Codex los ofrecen precisamente para describir procedimientos reutilizables, y su documentación explica mejor de lo que yo lo haría aquí cómo crearlos y usarlos.

Lo que pertenece al Contexto es la elección de las capacidades que se le dan: crear, abrir, alimentar, transformar, guardar, controlar y archivar. Es también el orden en que se encadenan y las fronteras que cada una debe respetar.

Los archivos conservan lo que el Contexto sabe. Los skills organizan los gestos por los que cambia.

El mapa está abierto. La visita puede empezar ahora como empieza una sesión de verdad: abriendo el Contexto.

PARA RECORDAR — EL ESQUELETO DE UN CONTEXTO

Un Contexto nuevo llega ya amueblado: seis zonas, cada una de las cuales responde a una pregunta distinta. Los números imponen un orden de lectura, no una jerarquía de valor. Lo que hace legible el árbol no es su forma, sino saber cómo cambia cada archivo. Claude Code y Codex aportan el mecanismo de los skills; el Contexto define los gestos recurrentes, su orden y sus límites.

5.2

Abrir: recuperar la misión y el presente

Han pasado dos semanas desde mi última sesión sobre los partes de intervención — esos documentos en los que un técnico anota lo que ha observado y lo que ha hecho tras una intervención en un equipo.

- Las entrevistas y las notas siguen ahí, en su sitio.

Pero si los reabro al azar, voy a pasar media hora reconstruyendo lo que ya sabía, y lo reconstruiré mal.

- La apertura empieza por tanto con un núcleo mucho más pequeño: el marco, el estado y la vuelta al trabajo.

No tengo que pedir todo eso. Llamo al skill de apertura y él desarrolla su procedimiento.

Primero recupero por qué estoy aquí

El primer archivo que la IA lee (mediante su skill de apertura de contexto) es `00-frame/mission.md`. Su plantilla plantea cinco preguntas:

```
# Misión

## El objetivo
¿Qué debe permitir este `Contexto`?

## La pregunta central
¿Qué pregunta guía la selección y la exploración?

## Lo que entra en el tema
¿Qué material puede entrar?

## Lo que queda fuera del tema
¿Qué se rechaza, aunque parezca interesante?

## El entregable esperado
¿Qué se quiere producir, si ya se sabe?
```

En el mío, el principio está rellenado así:

```
## El objetivo
Entender qué degrada la calidad de los partes de
intervención.

## La pregunta central
¿Cómo recoger información utilizable sin que rellenarla
resulte disuasorio?

## Lo que queda fuera del tema
- Elegir una interfaz antes de haber confrontado varias
fuentes.
- Generalizar a partir de una sola entrevista.

## El entregable esperado
Preparar las decisiones de producto y las próximas
entrevistas.
```

Una misión que dijera solo «mejorar los partes de intervención» no me orientaría gran cosa. La pregunta central, en cambio, ya hace que unos documentos sean más

pertinentes que otros, y lo que queda fuera del tema impide que una idea seductora convierta en silencio mi exploración en un proyecto de solución.

- El entregable, por su parte, puede seguir siendo desconocido.
- Escribo esa incertidumbre en vez de inventar una salida prematura: el Contexto sabe entonces que todavía explora.

Las reglas que había puesto me esperan

Dos misiones parecidas pueden exigir precauciones distintas.

`directives.md` guarda esas reglas junto al trabajo:

- confidencialidad,
- nivel de prueba,
- vocabulario,
- validación o forma esperada
- ...

La plantilla es deliberadamente sencilla: una regla, una fecha, una formulación imperativa.

Directivas

- (2026-09-11) Anonimizar cada entrevista antes de integrarla: sustituir el nombre de la persona entrevistada por su función, pero conservar el nombre de la empresa y el país.
- (2026-09-11) Distinguir lo que las fuentes muestran de lo que nosotros deducimos.
- (2026-09-11) No proponer ninguna solución antes de haber documentado los usos reales.

Una directiva solo vale si puede guiar una acción futura.

- «Ser prudente» me deja toda la interpretación por rehacer;
- «distinguir la observación de la deducción» me da un control preciso a la hora de escribir una nota o un entregable.

Pero una directiva no es solo una consigna que encuentro al abrir un archivo. Es una regla persistente que el Contexto debe aplicar cada vez que una acción entra en su campo.

Añado, por ejemplo, la transcripción de una entrevista nueva. Le pido sin más al Contexto que la integre (mediante el skill dedicado a eso), sin acordarme de señalar que contiene el nombre de la persona entrevistada. La directiva de anonimización interviene aun así durante la incorporación: antes de guardar la transcripción como fuente, el skill sustituye ese nombre por la función de la persona entrevistada, conservando a la vez el nombre de la empresa y el país.

No tengo que acordarme de la regla en el momento preciso en que se vuelve necesaria. El Contexto se acuerda por mí. Así que las directivas no dejan de actuar una vez terminada la apertura. Siguen aplicándose durante toda la sesión y acompañan los distintos gestos del Contexto:

- al añadir una fuente, la regla de anonimización protege la identidad de la persona entrevistada;
- al crear una nota, la regla sobre el nivel de prueba separa lo que la fuente muestra de lo que nosotros deducimos;

- al preparar un entregable, la última directiva impide presentar demasiado pronto una solución como cosa hecha.

El skill que ejecuta una acción debe empezar por identificar las directivas que la afectan y comprobar después que su resultado las respeta.

```
petición
↓
directivas aplicables
↓
acción del skill
↓
control del resultado
```

Las directivas rodean así la acción: la limitan antes de su ejecución y sirven para controlar su resultado después. No actúan solas y no lo saben todo. Son los skills los que las hacen operativas en el momento adecuado.

La misión indica adónde ir. Las directivas acompañan cada gesto del Contexto, para que no tenga que recordar, en cada paso, las precauciones ya decididas.

No todo merece que se vuelva a abrir

Mi carpeta contiene decenas de fuentes y de notas. Cargarlas todas en cada apertura ahogaría las prioridades en el montón.

loading.md elige por tanto un núcleo estable:

```
## Siempre
```

```
- `00-frame/mission.md`
- `00-frame/directives.md`
- `01-state/state.md`
- `01-state/todo.md`
- `01-state/handoff.md`
```

```
## A petición
```

```
- las fuentes necesarias para una comprobación
- las notas ligadas al tema trabajado
- el borrador activo
```

Todo sigue estando accesible, pero no todo se carga en el contexto de la IA. `loading.md` designa los elementos indispensables para entender la misión, interpretar una petición y actuar sin perder las reglas ya establecidas.

5

- El resto se queda en el disco y solo se consulta si hace falta.
- Una fuente antigua puede ser indispensable para controlar una afirmación sin ocupar sitio en el contexto activo durante todas las sesiones.

El contenido de `loading.md` depende del Contexto.

- Un libro cargará su plan en cada apertura; una exploración de producto, su nota temática central.
- El principio sigue siendo el mismo:
 - empezar con información suficiente para actuar,
 - e ir a buscar el resto cuando la tarea lo justifique.

En qué punto estaba de verdad

La misión se mantiene estable; `01-state/state.md`, en cambio, está hecho para cambiar. Cabe en tres bloques:

```
# Estado
```

```
## Lo que sabemos
```

- Rellenar más tarde el parte provoca una pérdida de detalles en la primera entrevista.
- Algunos campos obligatorios se esquivan en la segunda.

```
## Lo que sigue siendo incierto
```

- No sabemos qué campos se vuelven a usar realmente después de rellenarlos.

```
## Las deudas
```

- El informe técnico se ha añadido, pero falta calificar su nivel de fiabilidad.

El estado no cuenta toda la historia ni resume todas las notas: expone la mejor comprensión que tengo del trabajo mismo. `state.md` se dirige a la vez a la IA y al humano.

Para la IA constituye una memoria de trabajo persistente.

- Cargado en cada apertura, le indica lo establecido que puede movilizar, las incertidumbres que debe preservar y las deudas que no debe olvidar.
- Sin él, tendría que reconstruir ese estado a partir de las fuentes y de las notas en cada sesión nueva.

Para mí hace el papel de un cuadro de mando.

- En unas líneas puedo comprobar qué da el `Contexto` por establecido, qué considera todavía incierto y qué trabajos quedan incompletos.
- Puedo entonces corregir ese estado si no se corresponde con mi comprensión.

Unas decenas de líneas bastan así para que la IA retome el trabajo sin confundir lo establecido con las hipótesis, y para que yo controle la comprensión en la que se apoya.

`todo.md` extrae de él las acciones:

```
## En curso
- T-001 – Preparar una entrevista sobre el uso real de los
  campos obligatorios.

## Por hacer
- T-002 – Comprobar qué datos sirven para las auditorías.

## Hecho
- T-003 – Comparar las dos primeras entrevistas. –
  terminada el 2026-09-10
```

No abro necesariamente `todo.md` para modificarlo yo mismo.

Puedo decir sin más:

Añade una tarea para comprobar qué datos sirven para las auditorías.

El Contexto crea una línea, le asigna un identificador y la coloca en la sección adecuada.

Una tarea también puede aparecer durante el trabajo.

- Si el análisis de una fuente hace surgir una comprobación necesaria, la IA me propone añadirla en vez de dejarla desaparecer en la conversación.
- No convierte en silencio cada idea en una obligación nueva: cuando una tarea cambia las prioridades o supone trabajo adicional, el arbitraje sigue siendo mío.

Cuando el trabajo está hecho, puedo pedir:

Cierra la tarea T-002 (o una descripción aproximada de la tarea y la IA encontrará la referencia correcta).

El Contexto no borra la línea. La traslada a Hecho, añade su fecha de cierre y, cuando resulta útil, indica el resultado o el elemento que permite darla por terminada.

- Una tarea terminada no desaparece.
- Su presencia me evita rehacer el mismo trabajo y me explica por qué una pista dejó de estar activa.

Así que `todo.md` no es una lista escrita de una vez para siempre. Es la cola de acciones del Contexto: la IA la consulta para saber qué queda por hacer, la actualiza cuando el trabajo avanza y conserva las tareas terminadas para no volver a empezarlas.

Por dónde vuelvo a empezar

`handoff.md` es el último archivo del núcleo. Se reescribe en cada guardado para responder a una situación muy concreta: alguien llega ahora, una persona o una IA, sin nada de la conversación anterior. ¿Qué debe saber para volver a arrancar sin equivocarse?

```
# Vuelta al trabajo

## En qué punto estamos
Tres fuentes han hecho aparecer una tensión entre facilidad
de registro y aprovechamiento.

## Lo que hay que saber antes de seguir
Las dos entrevistas describen situaciones particulares,
mientras que el informe técnico defiende un punto de
vista general.
Este corpus hace aparecer una tensión, pero todavía no
permite medir su amplitud ni concluir que existe en todas
las organizaciones.

## La próxima acción
Preparar las preguntas que permitirán identificar los
campos realmente utilizados.

## Lo que ha quedado abierto
El grado de estructura que necesitan las auditorías sigue
siendo desconocido.
```

La próxima acción es única e inmediatamente ejecutable. Una lista de diez posibilidades me devolvería el trabajo de priorización que este archivo existe precisamente para ahorrarme.

Así que la orden de apertura absorbe lo que se ha depositado durante el sueño del Contexto, lee ese núcleo y me anuncia después esa acción.

- Recupera su finalidad antes de recuperar sus detalles, y mis dos semanas de ausencia ya solo cuestan un minuto de lectura.
- La sesión puede ahora recibir material nuevo sin perder su dirección.

PARA RECORDAR — EL NÚCLEO DE APERTURA

Reabrir un Contexto no consiste en releer su contenido, sino en cargar un núcleo pequeño: la misión, las directivas, el estado y la vuelta al trabajo. La misión dice adónde ir; las directivas determinan las condiciones del trayecto. Una vuelta al trabajo útil señala una sola acción, inmediatamente ejecutable.

5.3

Alimentar: hacer entrar el material

Empieza la sesión y llegan tres cosas a la vez.

- Un responsable de operaciones me envía el acta de una entrevista,
- se me ocurre una idea durante el intercambio,
- una nota que había dejado la víspera espera en una bandeja de entrada (gestor de notas, correo...).

Estos tres materiales no entran en el Contexto de la misma manera, y cada uno tiene su skill.

Un documento llega como un objeto terminado

Un documento externo se conserva con su archivo de origen, su texto utilizable y una ficha que permite seguirle la pista.

```
03-sources/documents/  
├─ SRC00001-entretien-exploitation.md  
├─ raw/  
│   └─ SRC00001-entretien-exploitation.pdf  
└─ fulltext/  
    └─ SRC00001-entretien-exploitation.md
```

La ficha SRC00001-entretien-exploitation.md puede tomar esta forma:

```
---  
added: 2026-09-11  
kind: pdf  
intention: entrevista  
raw: raw/SRC00001-entretien-exploitation.pdf  
fulltext: fulltext/SRC00001-entretien-exploitation.md  
extraction: ok  
reliability: medium  
---
```

Entrevista con la responsable de operaciones

Por qué está aquí

Entender qué impide a los técnicos producir partes de intervención utilizables.

Interpretación – sin analizar

La responsable describe una oposición entre la rapidez al rellenar el parte sobre el terreno y la necesidad de datos estructurados para el análisis.

Cobertura

- Registro en el móvil
- Campos obligatorios
- Reutilización de los datos

Lo que ha producido

- Por completar tras el análisis

Esta ficha no resume forzosamente el documento. Responde ante todo a preguntas de gestión:

- ¿de dónde viene?,
- ¿por qué lo he añadido?,
- ¿puedo leer su contenido?,
- ¿qué confianza darle?

La sección Interpretación – sin analizar pone en cuarentena una primera lectura. Evita que una impresión formulada al importar el documento se trate enseguida como un conocimiento establecido.

Cuando una fuente llega en forma de PDF, de página web o de otro formato difícil de aprovechar directamente, la extracción produce una versión textual legible para el humano y para la IA.

- Esta operación no produce todavía ningún conocimiento: solo hace accesible el material sin modificar el documento original.

La transformación viene después.

- El texto extraído se analiza para hacer aparecer notas atómicas, notas temáticas, términos de glosario o arbitrajes.
- Estas notas alimentan el *mundo de las ideas*, donde el material de las fuentes se convierte en un conjunto de piezas de conocimiento reutilizables.

El nivel de fiabilidad no juzga las fuentes añadidas.

- Indica sencillamente cómo usar ese material: testimonio aislado, dato verificado, documento contractual o hipótesis por confirmar.

No todo el material útil existe ya en un documento. Una parte está en mi experiencia, mis recuerdos, mis observaciones o las distinciones que formulo mientras trabajo.

Cuando las expreso durante una sesión de intercambio con la IA, también ellas entran en el Contexto.

- No se convierten automáticamente en hechos establecidos
- Conservan su origen y podrán confrontarse con otras fuentes.

Así que añadir material consiste tanto en integrar un documento como en conservar lo que la persona dueña del Contexto aporta a la conversación.

El conocimiento de la persona entra por la sesión

Una sesión no es un documento entregado de una vez para siempre. Se construye a lo largo del intercambio. Queda registrada en `03-sources/sessions/SESSION-2026-09-12.md`, que puede tomar esta forma:

```
# Sesión del 12 de septiembre de 2026
```

```
## 09:41 – IA
```

```
¿Los campos obligatorios mejoran realmente la calidad o solo trasladan el esfuerzo al técnico?
```

```
## 09:42 – Autor
```

```
En las entrevistas, los campos obligatorios garantizan sobre todo que se ha introducido un valor.
```

```
No garantizan ni su precisión ni su utilidad.
```

```
Cuando el técnico no entiende para qué servirá después un campo, lo rellena lo más deprisa posible para seguir adelante.
```

```
La obligación mejora entonces la completitud aparente, pero puede trasladar el esfuerzo sin mejorar la información realmente aprovechable.
```

```
...
```

No todo lo que dice la IA necesita conservarse. En cambio, una pregunta, una objeción o una reformulación que cambia el trabajo sí lo merece.

- Mi palabra y la de la IA siguen separadas.
- Así puedo encontrar el origen de una idea y distinguir mi intención de una propuesta del sistema.

Una sesión puede ser desordenada.

- Su papel es captar el movimiento real del pensamiento, no presentar una síntesis perfecta.

El buzón habla al Contexto cuando está cerrado

Se me ocurre una idea fuera de sesión, en el tren o en medio de una reunión y sin acceso a internet.

- Entonces entra en el buzón.
- Ese buzón es el archivo `offline.md`, colocado en la raíz del Contexto.
- Me basta con escribir en ese archivo Markdown.

Puede tomar esta forma:

Preguntar si los técnicos esquivan los campos obligatorios con valores genéricos.

En la siguiente apertura del contexto, esa nota se relee, se pone en relación con el trabajo en curso y se absorbe en el sitio adecuado. Puede convertirse en una tarea, en una pregunta de entrevista, en una nota, o desaparecer si ya no viene al caso.

Añadir no es todavía entender

Una vez entrado ese material, mi árbol contiene varias fuentes y el rastro de la sesión en curso.

```
03-sources/  
├── documents/  
│   ├── SRC00001-entretien-exploitation.md  
│   ├── SRC00002-entretien-exploitation.md  
│   └── SRC00003-livre-blanc-donnees-intervention.md  
└── sessions/  
    └── SESSION-2026-09-11.md
```

Estos archivos demuestran que el material está disponible. No demuestran que se haya entendido.

El archivo `03-sources/ingestion-log.md` distingue las dos operaciones. Para cada fuente indica si el análisis está completo, es parcial o hay que revisarlo, con una huella del contenido y unas notas de tratamiento.

Esa distinción se vuelve esencial cuando se sustituye un documento, cuando cambia una extracción o cuando una sesión se interrumpe. El Contexto puede entonces retomar el análisis sin fingir que lo sabe todo.

El material ha entrado. Me queda convertirlo en ideas lo bastante precisas para poder enlazarlas, discutir las y reutilizarlas.

PARA RECORDAR — LA FUENTE

Un documento, una palabra de sesión y una nota de fuera de sesión entran por tres puertas distintas y cada una conserva su forma de origen. Una fuente conserva el material y la prueba, nunca la interpretación que se sacará de ellos. Añadir no es entender: el seguimiento de ingesta distingue lo que está presente de lo que está asimilado.

Transformar: pasar de las fuentes a las ideas

Tengo ahora tres fuentes delante.

- Dos entrevistas describen la fricción que provocan los formularios,
- Un informe técnico defiende la necesidad de datos estructurados.

La tentación sería zanjar enseguida y escribir que los campos obligatorios perjudican la calidad. Sería ir demasiado rápido: el Contexto debe preservar esas aportaciones sin fundirlas en una conclusión única.

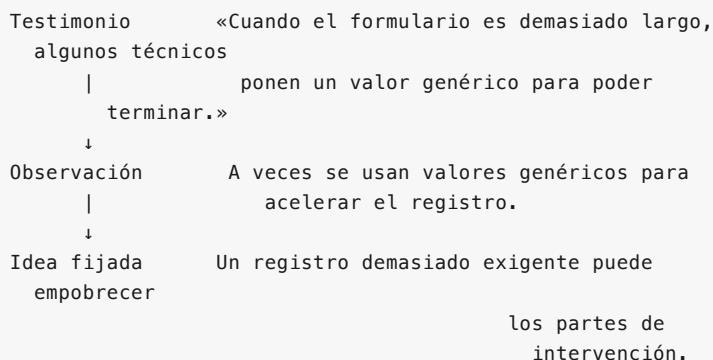
Llamo por tanto al skill que crea y actualiza las notas: lee lo que aún no se ha tratado y saca ideas de ello.

Extraigo una idea que se sostiene sola

Una fuente contiene pocas veces una idea ya lista para reutilizarse.

- Una entrevista mezcla hechos, ejemplos, opiniones, vacilaciones y detalles propios de la situación contada.
- Transformarla no consiste en resumirlo todo, sino en extraer las ideas.

El camino consta de tres etapas:



Una idea fuerte no es una fórmula espectacular ni una verdad definitiva.

- Tiene un centro de gravedad claro y firme,
- Puede entenderse fuera del texto que la hizo nacer
- Sigue siendo lo bastante delimitada para ser confirmada, matizada o contradicha.
- Conserva además su origen y sus límites, para no convertir un testimonio concreto en regla general.

La IA lee la fuente a la luz de la misión y de las directivas, y la confronta después con las notas ya presentes. Crea una nota nueva cuando la idea es nueva; si no, enriquece la existente con el matiz o el límite adicional.

Hace ese trabajo sin pedirme que valide cada formulación.

Puedo después releer y corregir las notas.

- Cuando aparece una ambigüedad durante la sesión, me pregunta;

- Si descubre más tarde una contradicción que no puede resolver, la conserva en vez de zanjar por mí.

```
04-notes/  
└─ atomic/  
    └─ AT000001-la-contrainte-de-saisie-peut-appauvrir-les-comptes-rendus.md
```

id: AT000001

status: active

sources: [SRC00001, SRC00002]

Un registro demasiado exigente puede empobrecer los partes de intervención

Idea principal

Cuando el formulario ralentiza la intervención, los técnicos pueden introducir valores genéricos o reducir su descripción al mínimo esperado.

Por qué importa

Un dato formalmente estructurado puede parecer limpio y describir peor la realidad del terreno.

Matices y límites

Las dos entrevistas no permiten medir la frecuencia del fenómeno.

Origen / prueba

- SRC00001: ejemplos de valores genéricos usados para terminar más deprisa.
- SRC00002: caída de precisión observada tras añadir campos obligatorios.

Enlaces

- Contraste con AT000002.

El título lleva una afirmación, no solo un tema. La nota sigue siendo comprensible sin que tenga que reabrir los documentos de los que procede, conservando a la vez los enlaces que me permiten comprobar su origen.

Una nota atómica no queda fijada en el primer intento.

- Puede ganar un matiz, una prueba nueva o una relación sin perder su identidad.

El informe técnico produce una segunda nota: los datos estructurados son necesarios para comparar, agregar y automatizar.

- Esa idea sigue separada de la primera porque afirma otra cosa.

Me niego a zanjar demasiado pronto

Las dos ideas parecen oponerse, y todavía no tengo material suficiente para arbitrarlas. Abro por tanto una contradicción.

C-001 – Estructurar sin degradar el registro

- Por un lado: los campos impuestos pueden empobrecer las descripciones de campo.
- Por otro: el aprovechamiento a escala exige datos comparables.
- Lo que está en juego: determinar qué informaciones deben estructurarse en el momento del registro y cuáles pueden estructurarse después.
- Abierta el: 2026-09-11

5

La contradicción me evita dos errores simétricos: elegir demasiado pronto un bando, o suavizar el desacuerdo en una formulación vaga que no molesta a nadie.

- Sigue abierta hasta que una decisión, una fuente nueva o una reformulación la resuelva.

No toda incertidumbre merece este tratamiento.

- Lo reservo a las tensiones que pesan de verdad sobre el resto del razonamiento o sobre la calidad de una decisión.

Enlazo lo que estas ideas dicen juntas

Durante la ingesta, la IA no crea solo notas atómicas.

- Examina también las relaciones entre las ideas extraídas y las notas ya presentes.
- Cuando varias de ellas forman un conjunto que no puede reducirse a una sola afirmación, crea en la misma operación una nota temática y pone los enlaces que las unen.

No tengo que lanzar una segunda orden.

- La creación de las notas atómicas, su acercamiento y la aparición eventual de una nota temática forman parte de una misma ingesta.

```
---
id: THM00001
status: active
sources: [AT000001, AT000002]
---
```

La calidad depende del registro y del aprovechamiento

Ángulo

La calidad de un parte de intervención no puede evaluarse solo por el número de campos rellenados.

Síntesis

El sistema debe obtener bastante estructura para aprovechar los datos sin volver el registro artificial o costoso.

Notas enlazadas

- AT000001 – La exigencia del registro puede empobrecer los partes de intervención.
- AT000002 – Los datos estructurados son necesarios para el aprovechamiento.

Tensiones

- C-001 – Estructurar sin degradar el registro.

La nota temática no yuxtapone solo resúmenes. Propone una lectura de sus relaciones y muestra lo que queda abierto.

Me doy cuenta de que una palabra no significa lo mismo para todos

Parte del desacuerdo viene a veces del vocabulario. El glosario fija entonces el sentido adoptado en este Contexto.

Dato estructurado

Información registrada en una forma explícita que permite compararla o tratarla automáticamente.

No confundir con: campo que el usuario está obligado a rellenar en el momento de la intervención.

Ejemplos: tipo de avería, tiempo de parada, equipo afectado.

Esta definición local no aspira a volverse universal.

- Simplemente impide que dos notas, o dos personas, usen la misma palabra para hablar de cosas distintas.

Piezas neutras frente al entregable

Una nota pertenece al *mundo de las ideas* mientras busca fijar un conocimiento en vez de producir un efecto en un público.

- No debe por tanto convertirse demasiado pronto en el gancho de un artículo, en una diapositiva, en un argumento comercial o en un trozo de especificación.

Esa neutralidad no significa que la nota siga siendo vaga.

- Al contrario: su título afirma algo preciso, su contenido expone los matices y sus enlaces permiten controlar los cimientos.
- Lo que queda neutro es su forma: no está organizada para servir a un solo resultado.

AT000001 podrá así convertirse en una pregunta sobre las evasiones en la guía de entrevista, en un riesgo en la nota de encuadre o en un ejemplo en la presentación al equipo.

AT000002 podrá sostener una pregunta sobre los usos del dato, justificar una exigencia de producto o explicarle al equipo por qué el texto libre no siempre basta.

La nota temática no funde esas piezas en un texto definitivo.

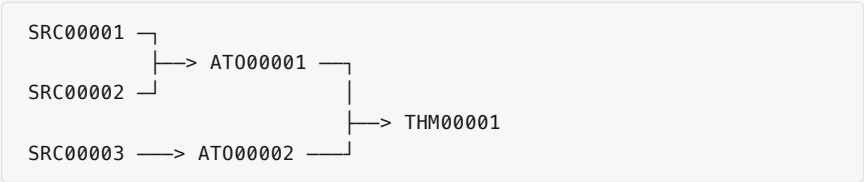
- Indica lo que permiten entender juntas y conserva la tensión que habrá que tratar de otra manera según el entregable.

Esta capa intermedia evita volver a empezar el análisis para cada producción.

- Las fuentes siguen siendo las pruebas,
- las notas se vuelven conocimientos movilizables,
- cada entregable compone con ellas según su propia necesidad.

Puedo volver de cada idea a su prueba

A estas alturas, las relaciones se representan así:



Los enlaces nombrados me permiten distinguir una fuente que confirma una idea de otra que la matiza o la contradice.

- Un nivel de confianza hace visible lo que se apoya en varias observaciones y lo que sigue siendo hipotético.

La cobertura de ingesta muestra qué fuentes han contribuido realmente a las notas. Me impide dar por asimilado un documento solo porque está en la carpeta.

Estas notas no están escritas para un entregable concreto.

- Forman un conocimiento común que podrá alimentar varios resultados sin ser copiado ni encerrado en uno de ellos.

El mundo de las ideas es precisamente ese capital reutilizable: un conjunto de piezas lo bastante estables para volver a encontrarlas y lo bastante independientes para recombinarlas.

PARA RECORDAR — LA NOTA ATÓMICA

Una idea fijada lleva una afirmación y no un tema, y se entiende sin reabrir el documento del que procede. Una contradicción se conserva mientras pese sobre el razonamiento, en vez de zanjarse demasiado pronto o suavizarse. Cada nota guarda el enlace hacia sus fuentes, lo que permite volver de la idea a su prueba.

Producir y decidir

Mi nota temática ha hecho aparecer una pregunta precisa:
¿qué informaciones hay que rellenar al final de la intervención
y cuáles pueden completarse después?

Ahora hay tres personas que esperan algo de mí.

- Un técnico al que debo entrevistar,
- un comité que debe decidir,
- un equipo al que tengo que explicárselo.

No voy a hacer tres investigaciones: voy a recurrir tres veces a las mismas notas.

Es el único momento de la sesión en el que ningún skill me ayuda: producir un entregable es un trabajo de escritura, no un gesto que se repite.

Recurro tres veces al mismo fondo

Pasar de las notas a los entregables no es copiar. Cada resultado selecciona las piezas que le son útiles, las ordena y las reformula en función de su destinatario.

Pieza común	Guía de entrevista	Nota de encuadre	Presentación al equipo
AT000001 — Un registro demasiado exigente	Preguntar qué campos se esquivan y cómo	Describir el riesgo de pérdida de calidad	Mostrar un ejemplo de fricción en el terreno

Pieza común	Guía de entrevista	Nota de encuadre	Presentación al equipo
puede empobrecer los partes de intervención			
AT000002 — Los datos estructurados son necesarios para el aprovechamiento	Preguntar qué datos se reutilizan realmente	Establecer la necesidad de comparación y de automatización	Explicar el valor esperado para el equipo
THM00001 — La calidad depende del registro y del aprovechamiento	Organizar la entrevista en torno a las dos exigencias	Formular lo que está en juego y los criterios de arbitraje	Construir el hilo del relato
C-001 — Estructurar sin degradar el registro	Convertir la tensión en preguntas por comprobar	Conservarla como problema por resolver	Hacerla visible sin anunciar una solución prematura

El mismo fondo produce así tres montajes, sin que ningún entregable lo recoja todo ni sustituya a las notas que lo alimentan.

Esa separación me permite sobre todo corregir el conocimiento una sola vez.

- Si una fuente nueva matiza AT000001, mis próximos entregables partirán de esa pieza enriquecida, en vez de reproducir una formulación antigua dispersa en varios documentos.

Empiezo por lo más urgente: la próxima entrevista

Mi primera necesidad es preparar la entrevista con un técnico. El entregable empieza en el espacio de los borradores.

```
05-outputs/  
└─ _draft/  
    └─ questions-prochain-entretien.md
```

Preguntas para la próxima entrevista

1. ¿Qué campos le frenan más durante una intervención?
2. ¿Qué anota cuando la información pedida no está disponible?
3. ¿Qué informaciones podría añadir después de la intervención?
4. ¿Qué datos deben ser exactos de inmediato para que sirvan al aprovechamiento?

Las notas buscan conservar una idea reutilizable. El entregable, en cambio, elige un destinatario, una forma y una acción esperada.

Una vez releído y validado, el archivo sale de `_draft/` y pasa directamente a `05-outputs/`. El traslado indica un nivel de madurez, no una diferencia de género documental.

Hasta aquí no he hecho más que producir: he dado forma a lo que ya sabía, para tres públicos. Lo que sigue de este capítulo es un gesto muy distinto. En algún momento hay que dejar de restituir y zanjar — y lo que zanjo tiene que poder encontrarse dentro de seis meses.

El decision record guarda la elección que moldea el resultado

«Entrevistar a un técnico antes de redactar una especificación» no es un arbitraje sobre lo que se entregará.

- Si esa entrevista es sencillamente la próxima acción, pertenece a `01-state/todo.md`.
- Si todos los trabajos de descubrimiento deben respetar esa regla, se convierte en una directiva en `00-frame/directives.md`.

5

En ninguno de los dos casos justifica un decision record.

El decision record aparece cuando el trabajo hace surgir varias direcciones posibles y hay que elegir la que estructurará el producto, el texto, el método u otro resultado del Contexto. Guarda entonces lo que se ha decidido concebir o entregar, las alternativas descartadas, las razones de la elección y las consecuencias aceptadas.

En mi ejemplo, el arbitraje no trata por tanto de la organización de las entrevistas. Trata de una decisión funcional:

- exigir todas las informaciones al final de la intervención,
- conservar únicamente texto libre,

- o permitir que el parte se complete en un segundo momento.

Un decision record útil no delega el núcleo de la elección en la persona que lo llevará a cabo. Tres personas que lo lean deben entender el mismo perímetro, los mismos comportamientos esperados y los mismos límites. Los detalles técnicos que no estructuran nada pueden quedar abiertos, pero ninguno de ellos debe poder modificar en silencio la decisión.

```
04-notes/  
└─ decision-records/  
    └─ DRE00001-limiter-la-cloture-a-quatre-informations.md
```

Esto es lo que da una vez escrito. La plantilla es larga porque conserva todo el razonamiento que hace la decisión comprensible y revisable — quien ya conozca los decision records puede saltar directamente al comentario que lo sigue.

5

```
---  
uuid: 0d5efbe2-1eac-4ad7-94c1-29f68adcf001  
id: DRE00001  
created: 2026-09-11  
updated: 2026-09-11  
status: accepted  
decided_by: equipo de producto  
decided_on: 2026-09-11  
supersedes: null  
superseded_by: null  
sources: [AT000001, AT000002, THM00001]  
---  
  
# Limitar el cierre de una intervención a cuatro
```

informaciones obligatorias

Decisión

Para que una intervención se considere cerrada deben estar presentes exactamente cuatro informaciones:

1. El equipo afectado, ya indicado en la ficha de intervención o introducido en el momento del cierre.
2. La fecha y la hora reales de fin de la intervención.
3. El estado del equipo al final de la intervención: devuelto al servicio, mantenido en funcionamiento degradado o dejado fuera de servicio.
4. Un resumen no vacío de los trabajos realizados o de la razón por la que el equipo no ha podido devolverse al servicio.

Ninguna otra información condiciona el cierre.

Tras el cierre, el parte puede completarse con la naturaleza del fallo, su causa identificada, la naturaleza de los trabajos realizados, las piezas sustituidas y el tiempo de indisponibilidad del equipo, sin poner en cuestión el cierre de la intervención.

El técnico puede introducir esas informaciones complementarias, pero el responsable de operaciones debe completarlas o validarlas en veinticuatro horas.

Un parte todavía incompleto sigue pudiendo consultarse. Se señala como pendiente de completar, y después se avisa al responsable de operaciones pasadas veinticuatro horas.

Contexto

Los técnicos deben poder cerrar su trabajo en el terreno sin recorrer un formulario de análisis completo.

El departamento de operaciones necesita, sin embargo, información comparable para analizar los fallos, sus causas, los trabajos realizados, las piezas sustituidas y los tiempos de indisponibilidad.

Hipótesis

- La ficha de intervención identifica normalmente el equipo antes del inicio de los trabajos.
- El técnico conoce el estado del equipo y puede describir brevemente los trabajos realizados en el momento del cierre.
- El responsable de operaciones dispone del tiempo y de las informaciones necesarias para validar el parte completo en veinticuatro horas.

Alternativas

- Exigir las nueve informaciones en el momento del cierre.
- Conservar solo un resumen libre y extraer de él automáticamente la información sobre las intervenciones a posteriori.
- Hacer todas las informaciones opcionales y controlar únicamente la calidad después.

Razones

Las cuatro informaciones inmediatas bastan para identificar la intervención, fechar su fin, conocer el estado del equipo y entender los trabajos realizados.

Las cinco informaciones de análisis siguen siendo necesarias, pero no deben impedir al técnico terminar su trabajo en el terreno.

El segundo momento atribuye explícitamente al responsable de operaciones la responsabilidad de completar o validar el parte.

Base

- AT000001 describe las evasiones ligadas a un registro demasiado exigente.
- AT000002 establece la necesidad de datos comparables.
- THM00001 enlaza las dos exigencias.

Consecuencias

Positivas

- El cierre solo pide cuatro informaciones conocidas de inmediato.
- Las informaciones necesarias para los análisis de las intervenciones siguen disponibles una vez completado el parte.
- La responsabilidad de los datos que faltan es visible.

Contrapartidas

- Los indicadores de intervención pueden quedar incompletos durante las veinticuatro horas siguientes a una intervención.
- El responsable de operaciones recibe una carga de validación adicional.

Riesgos

- Pueden acumularse partes incompletos.
- Las informaciones añadidas a posteriori pueden reflejar con menos fidelidad la situación observada en el terreno.

Criterios de conformidad

- Una intervención no puede cerrarse si falta una de las cuatro informaciones inmediatas.
- Una intervención puede cerrarse sin ninguna de las cinco informaciones complementarias.
- Las informaciones complementarias pueden modificarse sin cambiar el estado de cierre de la intervención.
- El parte se presenta como «pendiente de completar» o «completo y validado» en todos los usos que se hacen de él.
- Se avisa al responsable de operaciones veinticuatro horas después del cierre si el parte no está completo y validado.

Decisiones relacionadas

- Resuelve C-001 – Estructurar sin degradar el registro.

Fuentes

- SRC00001
- SRC00002
- SRC00003

El estado `accepted` indica aquí que el arbitraje está validado y es lo bastante preciso para guiar la implementación.

- Mientras las informaciones esperadas, las responsabilidades o los criterios de conformidad sigan indeterminados, el decision record debe conservar el estado `proposed`.
- Una vez aceptado, queda fijado como rastro histórico y ya no cambia en silencio.

Si una decisión nueva lo sustituye, el decision record antiguo sigue pudiendo consultarse y los dos archivos quedan enlazados por `supersedes` y `superseded_by`.

Esa continuidad conserva la cronología completa de los arbitrajes: lo que se había decidido, sobre qué informaciones, y qué elemento llevó después a elegir de otra manera.

Seis meses más tarde, por tanto, no recupero solo la regla actual. Puedo entender por qué una primera dirección había parecido acertada, qué la volvió insuficiente y cómo se construyó la decisión siguiente.

Sin esa cronología, una sucesión de elecciones acaba pareciendo una incoherencia. Con ella, vuelve a ser la historia comprensible de un producto que evoluciona.

- Las alternativas muestran lo que se descartó.
- Las contrapartidas hacen visible lo que se sacrificó.

- La base enlaza la decisión con los conocimientos y las pruebas que la sostienen.

Producir no agota lo que sé

La guía de entrevista, la nota temática y el decision record tratan del mismo problema sin duplicarse, porque cumplen tres funciones distintas.

- La nota conserva lo que el Contexto cree saber.
- La decisión conserva una elección y sus razones
- La guía convierte ese conocimiento en acción ante un destinatario concreto.

Otro entregable podrá retomar el mismo conocimiento común bajo otra forma. **Producir no vacía por tanto el Contexto de su sustancia: revela lo que esa sustancia permite hacer.**

Me queda registrar el estado alcanzado, para que la próxima sesión retome sin tener que reconstruir el camino recorrido.

5

PARA RECORDAR — EL DECISION RECORD

Las mismas notas alimentan varios entregables, y ningún entregable consume el conocimiento que usa. Un decision record guarda un arbitraje que moldea lo que se concebirá o se entregará, no una acción por realizar. Conserva lo que se descartó y lo que se sacrificó, para que la elección siga siendo revisable.

Guardar y retomar

El decision record acaba de aceptarse y dentro de tres minutos empieza una reunión. No he terminado ni el estudio ni siquiera una etapa que se pareciera a un final.

Es el caso corriente: otra tarea reclama mi atención, la máquina debe reiniciarse, cambio de ordenador, o la ventana de contexto de la IA se carga demasiado.

- Guardar no responde por tanto a una necesidad del trabajo mismo, del tipo «el estudio está terminado».
- Es una garantía técnica: el estado del Contexto debe sobrevivir a la sesión en la que se ha producido.

Debo poder interrumpirme ahora, reabrir más tarde y encontrar el trabajo exactamente donde lo dejé.

De eso se encarga un skill, y debe resolverse en unos segundos: un guardado que sale caro no se lanzará en el momento en que uno tiene prisa, es decir, en el momento en que más sirve.

La IA fija lo que se sabe en este instante

El archivo `01-state/state.md` resume lo que es lo bastante estable para orientar lo que sigue.

```
contexte-qualite-comptes-rendus/  
└─ 01-state/  
   └─ state.md
```

En el instante del guardado puede tomar esta forma:

```
# Estado actual
```

```
## Conocimiento fijado
```

- Un registro demasiado exigente puede empobrecer los partes de intervención.
- El aprovechamiento a escala exige sin embargo datos estructurados.

```
## Decisión tomada
```

- El cierre exige el equipo afectado, la fecha y la hora de fin, el estado del equipo y un resumen de los trabajos realizados.
- La naturaleza y la causa del fallo, la naturaleza de los trabajos, las piezas sustituidas y el tiempo de indisponibilidad se completan tras el cierre y las valida en veinticuatro horas el responsable de operaciones.

```
## Incertidumbre principal
```

- Medir si esta separación reduce realmente el tiempo de cierre sin degradar la calidad de los datos.

```
## Deuda de conocimiento
```

- Ningún piloto ha medido todavía el tiempo de cierre ni la proporción de partes completos pasadas veinticuatro horas.

- Una tarea terminada puede desaparecer de la lista de tareas.
- Una contradicción no resuelta sigue visible en el estado.

La síntesis solo contiene lo que el trabajo ha fijado lo bastante. El estado se reescribe para describir el presente,

mientras que la síntesis se enriquece poco a poco para conservar la comprensión adquirida.

Preparo una sola acción para mi yo futuro

El archivo de vuelta al trabajo no cuenta toda la historia: prepara el próximo gesto útil.

```
contexte-qualite-comptes-rendus/  
└─ 01-state/  
   └─ handoff.md
```

El archivo 01-state/handoff.md puede tomar esta forma:

```
# Vuelta al trabajo  
  
## Estado actual  
  
DRE00001 está aceptado y C-001 está resuelta.  
Las cuatro informaciones de cierre y las cinco  
informaciones complementarias quedan fijadas.  
  
## Punto de vigilancia  
  
No añadir una información obligatoria al cierre sin abrir  
un decision record nuevo que sustituya a DRE00001.  
  
## Próxima acción  
  
Arrancar la puesta en práctica de DRE00001.  
  
## Abierto  
  
- Definir los rótulos que se presentan a los técnicos para  
  los tres estados posibles del equipo tras la  
  intervención.  
- Preparar la medida del tiempo de cierre y de la  
  proporción de partes completos pasadas veinticuatro  
  horas.
```

Una vuelta al trabajo útil señala una sola próxima acción. Una lista de diez posibilidades obligaría a la sesión siguiente a rehacer el trabajo de decisión que acabo justamente de hacer.

Guardo un rastro para comparar mis reanudaciones

El archivo `01-state/checkpoints.md` conserva una serie de fotografías compactas de la situación.

```
contexte-qualite-comptes-rendus/
└─ 01-state/
   └─ checkpoints.md
```

El archivo puede tomar esta forma:

```
## 2026-09-11
```

- Fuentes: 3
- Notas atómicas: 2
- Notas temáticas: 1
- Contradicciones abiertas: 0
- Decision records aceptados: 1
- Próxima acción: arrancar la puesta en práctica de DRE00001.

```
## 2026-09-04
```

- Fuentes: 1
- Notas atómicas: 1
- Notas temáticas: 0
- Contradicciones abiertas: 0
- Próxima acción: obtener el punto de vista del área de explotación.

Estas fotografías me muestran la evolución del Contexto sin obligarme a comparar todos los archivos. También me hacen

detectar un estancamiento, o una deuda que crece sin que me dé cuenta.

La IA anota lo que ha pasado, no lo que ella opina

El historial registra los acontecimientos importantes sin intentar reconstruir cada intercambio. En mi ejemplo, los acontecimientos del día se añaden a `02-history/2026-09-11.md`.

```
contexte-qualite-comptes-rendus/  
└─ 02-history/  
   └─ 2026-09-11.md
```

El bloque añadido puede tomar esta forma:

```
## 2026-09-11  
  
- Añadido de SRC00003-livre-blanc-donnees-intervention.  
- Creación de AT000002 y THM00001.  
- Apertura de C-001.  
- Validación de `questions-prochain-entretien.md`.  
- Aceptación de DRE00001 y resolución de C-001.
```

5

- El estado dice en qué punto estamos.
- El historial dice lo que ha pasado.
- La vuelta al trabajo dice qué hacer después.

Dejo que la máquina compruebe lo que yo no vería

En el momento de guardar, unos cuantos controles mecánicos reducen los olvidos. Su resultado se escribe en `01-state/health.md`.

```
contexte-qualite-comptes-rendus/  
└─ 01-state/  
   └─ health.md
```

- ¿Existen todos los archivos a los que se hace referencia?
- ¿Son únicos los identificadores?
- ¿Indica cada nota sus fuentes?
- ¿Se señala una fuente añadida pero no analizada?
- ¿Están las decisiones sustituidas enlazadas con las que las sustituyen?
- ¿Es explícita la próxima acción?

Estas comprobaciones no sustituyen a mi juicio. Simplemente impiden que un error de estructura se haga pasar por un problema de pensamiento.

5

El ciclo completo y lo que permite

El movimiento general se vuelve ahora visible.

```
apertura
↓
lectura del buzón
↓
carga de la misión, del estado y de la vuelta al trabajo
↓
añadido de una fuente
↓
creación o actualización de las notas
↓
producción de un borrador
↓
control, validación o decisión
↓
actualización del estado, de la síntesis y de la vuelta al
trabajo
↓
checkpoint e historial
↓
control de guardado
↓
próxima apertura
```

5

No espero al final de esa cadena para guardar.

- Lo hago tras añadir una fuente, en mitad de la creación de una nota, antes de una decisión o mientras preparo un entregable, cuando veo que mi ventana de contexto revienta...

Ese es precisamente el papel del guardado: **hacer que cada uno de esos estados pueda retomarse, con independencia de la conversación en curso.**

Así que mi reunión puede empezar.

El contexto no es solo un árbol de archivos: es un sistema para pensar a lo largo del tiempo, que hace entrar

material, preserva las tensiones, fija ideas, produce varios resultados y prepara la vuelta al trabajo.

Su forma sigue siendo deliberadamente sencilla y portátil — unos archivos legibles bastan para hacer el trabajo observable, comprobable y transmisible.

Una vez entendido este motor, aparece otra pregunta: ¿cómo dialoga el Contexto con las herramientas, las bases de conocimiento y el producto real que existen fuera de él?

PARA RECORDAR — EL GUARDADO

Guardar no marca el final de una etapa: es la garantía de que el estado sobrevive a la sesión que lo ha producido. El estado dice en qué punto estamos, el historial lo que ha pasado, la vuelta al trabajo qué hacer después. El ciclo puede interrumpirse en cualquier punto, y todos esos puntos pueden retomarse.

Parte 6

El Contexto en su ecosistema

Un Contexto bien llevado da ganas de meter en él todo el conocimiento disponible.

Esta parte examina lo que deja fuera y por qué. Se mantiene pequeño para conservar su capacidad de razonar. Todo lo que guardara por simple precaución envejecería sin avisar mientras el original siguiera cambiando en otro sitio. Así que trabaja con vecinos: herramientas cuya función es conservar, medir o ejecutar, y otros Contextos que investigan preguntas distintas de la suya.

Consultar conocimientos estructurados

Los partes de intervención esperarán: esta mañana trabajo en otro asunto del mismo producto, las personas que abandonan a mitad del registro. Una agencia acaba de entregarme su estudio y lo añado al Contexto: varios participantes dudan ante las reglas de la contraseña.

Esa observación me recuerda una idea con la que ya me había encontrado: exigir un esfuerzo importante antes de que el usuario perciba el valor del producto puede llevarlo a abandonar.

- Estoy casi seguro de tener material sobre el tema.
- Simplemente ya no sé dónde está ni qué palabras había empleado.

6

¿Hablé de esfuerzo inicial, de fricción, de activación o de onboarding? La idea quizá haya sido

- sacada de un libro,
- desarrollada en una nota de product management,
- enlazada con otro recorrido de usuario,
- ...

Podría probar varias búsquedas por palabras clave y abrir los resultados uno a uno. En vez de eso, le pregunto al Contexto:

¿Qué hemos aprendido ya sobre los esfuerzos pedidos antes de que el usuario perciba el valor del producto?

El Contexto consulta una memoria más amplia

La respuesta no está en el Contexto activo. En cambio, puede estar en mi LLM Wiki, una biblioteca común donde ya están estructurados los miles de notas sacadas de libros, artículos, normas y otros documentos.

El LLM Wiki y el Contexto no son de la misma naturaleza.

El wiki es una base de conocimientos.

- Conserva notas estructuradas (en mi caso he reutilizado la idea de las notas atómicas) a partir de lo que otros han escrito: una definición, un principio, el resultado de un estudio, una exigencia de una norma.
- Cada nota dice lo que se sabe y de dónde viene.
- Puede contener decenas de miles, acumuladas durante años y sobre temas sin relación entre sí.

Ningún Contexto contendrá nunca esa cantidad, y no es su papel.

Pero el wiki se detiene ahí.

- No dice por qué ese conocimiento cuenta en el trabajo que llevo hoy, ni qué he hecho con él.
- No enlaza una idea con una decisión, ni una decisión con lo que me costó.

El LLM Wiki no es un sistema de pensamiento: conserva conocimiento a gran escala, pero no razona.

El razonamiento vive en el Contexto. Es él quien guarda el porqué del cómo:

- cuál es la misión,

- qué directivas y decisiones están acordadas,
- por qué se retuvo esta idea para esta misión,
- qué alternativas he descartado,
- qué he aceptado perder,
- qué contradicción sigue abierta.

Solo lo consigue porque se mantiene pequeño — unas decenas de notas que un humano puede releer.

Cada uno paga el precio de su fuerza.

- **El wiki gana la escala y pierde el hilo.**
- **El Contexto guarda el hilo y renuncia a la escala.**

Justamente por eso deben hablarse en vez de fundirse.

Describo la idea que busco

No copio todo el wiki en el Contexto dedicado al registro.

Tampoco ejecuto una consulta RAG seguida de una búsqueda en un grafo.

Le planteo sin más mi pregunta al Contexto:

¿Qué hemos aprendido ya sobre los esfuerzos pedidos antes de que el usuario perciba el valor del producto?

La formulo con mis propias palabras.

- No necesito conocer ni el título de una nota, ni dónde está, ni el vocabulario empleado cuando se escribió.

El Contexto transmite esa idea al LLM Wiki por su API. Todo lo demás pasa dentro del wiki.

El wiki busca por el sentido y luego alrededor del resultado

El LLM Wiki empieza por recuperar las notas cuyo sentido se acerca a mi pregunta.

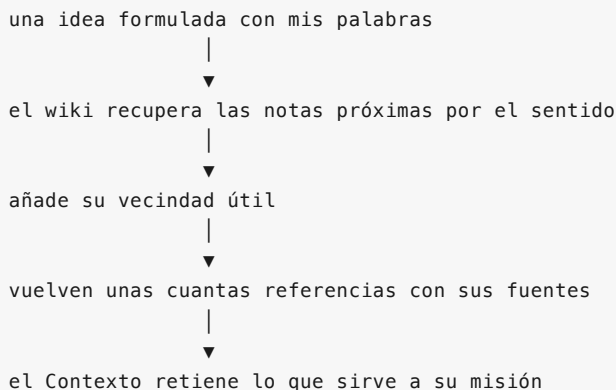
- Una nota puede hablar de fricción inicial, de activación o de tiempo hasta el primer valor sin emplear la palabra «esfuerzo».

Eso es lo que su RAG le permite hacer: encontrar conocimientos ya estructurados próximos a mi idea sin exigir el mismo vocabulario.

Después el wiki mira alrededor de los mejores resultados. Su grafo semántico sigue las relaciones ya establecidas entre las notas y añade las vecinas que pueden aclarar el tema:

- una definición,
- un límite,
- una contradicción,
- una idea complementaria.

Una de esas notas vecinas quizá nunca habría respondido directamente a mi pregunta. Se vuelve preciosa de todos modos porque muestra en qué condiciones se aplica la idea, o en qué punto deja de ser cierta.



El RAG recupera lo que yo no habría sabido nombrar. El grafo hace aparecer lo que no se me habría ocurrido buscar.

Para mí, esas dos operaciones forman una sola. Pido referencias que respondan a un problema; el LLM Wiki busca la idea y su vecindad, y luego me devuelve una lista corta.

6

El Contexto recibe referencias, no una respuesta hecha

El wiki devuelve notas existentes con su contenido, su origen y sus relaciones. No fabrica una síntesis que mezclaría en silencio varios resultados.

Así que el Contexto puede leer cada resultado, compararlo con las observaciones sobre el registro y volver a su fuente cuando hay que comprobar una prueba.

- Algunas notas se refieren efectivamente a un esfuerzo impuesto antes del acceso al producto.
- Otras hablan de un esfuerzo pedido más tarde, cuando el usuario ya conoce su valor.

Su proximidad con mi pregunta no basta para hacerlas equivalentes.

El Contexto hace la selección a partir de su misión. Puede:

- retener una idea que aclara el abandono durante el registro,
- conservar una contradicción,
- descartar una nota demasiado general.

Una nota retenida entra en el Contexto con su origen y las fuentes en las que se apoya. El resto se queda en el wiki.

Dispongo así de una memoria mucho más amplia sin convertir cada Contexto en una copia de toda mi biblioteca.

- No necesito aprender cómo está indexada esa memoria para servirme de ella: describo la idea que busco y luego compruebo las referencias que me trae.

Esta cadena da acceso al conocimiento ya estructurado.

- Todavía no dice qué hace realmente el producto hoy.

Para responder a esa otra pregunta, el Contexto debe salir de la biblioteca y consultar directamente el código, las pruebas y el producto en funcionamiento.

PARA RECORDAR — DOS SISTEMAS, UNA PUERTA

El LLM Wiki gana la escala y pierde el hilo; el Contexto guarda el hilo y renuncia a la escala. Describo una idea con mis propias palabras; el wiki recupera las notas próximas por el sentido y añade su vecindad útil. El Contexto recibe unas cuantas referencias acompañadas de sus fuentes y solo retiene lo que sirve a su misión.

Consultar el producto real

Estoy releendo el estudio sobre los abandonos durante el registro cuando llega una pregunta:

¿Un usuario invitado tiene que crear una contraseña como los demás?

La pregunta parece minúscula. Sin embargo, la respuesta la espera una persona de soporte que debe contestar a un cliente.

Podría:

- interrumpir mi lectura,
- abrir un entorno de pruebas,
- preparar dos cuentas,
- repetir los recorridos,
- comparar lo que pasa.

Es lo que hacía antes.

- Media hora después tendría probablemente la respuesta.
- También habría perdido el hilo de mi trabajo.

Esta vez le hago la pregunta al Contexto.

El Contexto consulta el producto

El Contexto trabaja con las notas salidas del estudio de usuarios y conserva ese estudio como fuente de sus afirmaciones.

- Lleva además la misión en curso.
- Entiende por qué la pregunta cuenta, pero esos elementos no le permiten conocer el comportamiento actual del producto.

Ya no mantengo una especificación encargada de describir ese comportamiento a largo plazo.

- El producto cambia demasiado a menudo para que un documento escrito al lado siga siendo su copia fiel.

La única verdad al día es lo que está entregado y funcionando en producción: el código fuente de la aplicación.

En mi sistema, unos skills externos al Contexto saben consultar el código fuente y generar documentación a partir de la implementación. El Contexto llama al que responde a las preguntas de producto y le transmite mi petición.

6



El skill empieza por encontrar el recorrido de invitación en el código. Después sigue ese recorrido hasta la regla de creación de la cuenta.

El código muestra el comportamiento programado. Una prueba confirma que un usuario invitado puede terminar su registro gracias al enlace recibido, sin crear una contraseña de inmediato. La configuración indica que ninguna variante propia de un cliente está activa para ese recorrido.

El skill me devuelve una respuesta funcional:

Un usuario invitado puede acceder a su cuenta desde su enlace de invitación sin crear una contraseña durante esa etapa. Si luego decide crear una, se aplican las reglas ordinarias. Las pruebas confirman el recorrido de invitación y la creación voluntaria de la contraseña. El recorrido de recuperación de una contraseña olvidada no se ha examinado.

Puedo pasarle las dos primeras frases a soporte. También puedo abrir las referencias que aporta el skill si quiero controlar su análisis con un desarrollador.

Por último, puedo registrar esa respuesta en el Contexto, con sus pruebas, la fecha de la comprobación y el límite señalado.

- Responder a soporte resuelve un caso y se acaba ahí.
- Registrarla en el Contexto hace entrar un hecho verificado sobre el producto en el *mundo de las ideas* de la misión: la próxima pregunta sobre el recorrido de invitación no partirá de cero, y la nota salida del estudio gana lo que la observación sola no daba — el momento exacto en que cada recorrido pide la contraseña.

Una respuesta vale por las pruebas que la acompañan

El skill no se ha limitado a producir una respuesta plausible.

Para cada afirmación indica qué ha consultado:

- la regla presente en el código,
- la prueba que verifica el comportamiento,
- la configuración que determina la variante activa.

Ese rastro protege contra una respuesta convincente pero falsa. También permite ver enseguida lo que falta.

Aquí, el skill no ha comprobado la recuperación de una contraseña olvidada. Lo dice en vez de extender su conclusión a un recorrido que no ha leído.

Según la pregunta, otros elementos pueden volverse necesarios.

- El código contiene las reglas posibles, pero un feature flag indica cuál está activa.
- La estructura de la base muestra lo que el producto puede registrar, mientras que los datos autorizados permiten entender la situación de una cuenta concreta.
- El historial Git y las pull requests ayudan a encontrar cuándo cambió una regla y las explicaciones conservadas con ese cambio.
- ...

El skill reúne solo lo que necesita para responder a la pregunta planteada. No absorbe todo el repositorio ni convierte cada archivo en fuente del Contexto.

La respuesta sigue siendo legible para soporte, para ventas o para un product manager. Las pruebas siguen disponibles para quien tenga que comprobarla o seguir con la investigación.

Encontrar la regla no decide cambiarla

Vuelvo a mi estudio sobre el registro. Ya me había enseñado que los dos recorridos no presentan la creación de la contraseña en el mismo momento.

El skill añade lo que el estudio no podía ver. El usuario invitado no esquiva la regla de la contraseña: se topa con ella más tarde, y entonces es exactamente la misma.

Así que la fricción no ha desaparecido para él, ha cambiado de sitio. Esa precisión desplaza mi pregunta. Ya no busco saber quién sufre la regla, sino en qué momento se topa cada cual con ella: antes de haber percibido el valor del producto, o después.

El código me ha enseñado lo que hace el producto.

- No me dice si la regla es buena, si protege realmente las cuentas o si su coste para los usuarios es aceptable.

Para decidir, sigo necesitando los demás elementos del

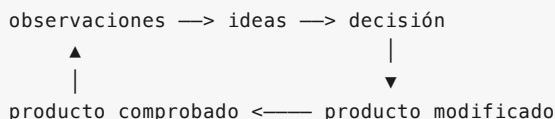
Contexto:

- las observaciones del estudio y su alcance;
- los objetivos de seguridad;
- las razones que habían llevado a adoptar esa regla;
- las consecuencias esperadas de un cambio.

El skill establece un hecho sobre el producto. El Contexto enlaza ese hecho con las notas y las decisiones ya presentes, conservando a la vez las fuentes que permiten controlar su origen. **La decisión sigue siendo mía.**

Si decido modificar la creación de la contraseña, el decision record conserva la elección, su alcance y las razones que la sostienen. Después de la implementación, el skill puede volver al producto y comprobar que el comportamiento nuevo se corresponde con la decisión.

Así que el movimiento funciona en los dos sentidos. El producto aclara la reflexión, y luego la decisión transforma el producto.



Guardar el porqué, reconstruir el qué

EL ENFOQUE CODE-CENTRIC

El código ya no es solo el resultado final del desarrollo. Se convierte en la fuente a partir de la cual puedo reconstruir lo que describe el comportamiento actual del producto. El Contexto conserva lo que el código no puede explicar solo: las fuentes, el razonamiento, las decisiones, las alternativas descartadas y las razones de la elección.

Esa separación cambia el lugar de la documentación.

Conservo lo que explica por qué el producto tomó una dirección: las entrevistas, las restricciones, las hipótesis y las decisiones. Reconstruyo desde el producto lo que describe su comportamiento actual:

- una respuesta a soporte,
- una documentación funcional,
- un registro de cambios,
- ...

Ya no mantengo manualmente dos versiones del mismo comportamiento, una en el código y otra en un documento que empieza a envejecer desde su publicación (e incluso mucho antes).

Aun así, no todo viene del código.

- Un estudio de usuarios no está ahí.
- Una obligación reguladora guarda su propia fuente.
- Una decisión no se reduce al comportamiento que resulta de ella.

El código registra lo que se ha implementado. El Contexto guarda el porqué y permite discutirlo cuando la situación cambia.

Volver seis meses más tarde

La respuesta dada a soporte sigue ligada a la versión del producto examinada hoy. No describe automáticamente las versiones siguientes.

- Seis meses más tarde, el recorrido de invitación puede haber cambiado.
- Un feature flag puede activar otra variante, o puede haber aparecido un modo nuevo de conexión.

El `Contexto` conserva la pregunta, la respuesta funcional, la fecha de la comprobación, la versión examinada, las pruebas citadas y el límite señalado. Así recupero lo que sabíamos cuando razonamos.

Para conocer el comportamiento presente, le pido sin más al skill que repita la comprobación.

- Parte del producto real en vez de copiar la respuesta antigua.

Puedo entonces distinguir dos preguntas que a menudo se confunden:

- ¿por qué tomamos esa decisión en su momento?
- ¿qué hace el producto hoy?

El `Contexto` responde a la primera conservando el razonamiento. El skill responde a la segunda volviendo a la fuente de verdad.

Así que el producto no sustituye al `Contexto`, y el `Contexto` no pretende contener el producto.

- Uno aporta una realidad que comprobar;
- el otro guarda los conocimientos necesarios para darle sentido y hacerla evolucionar.

Esa circulación no se detiene en las herramientas exteriores. Cuando varios Contextos trabajan en aspectos distintos de un mismo producto, sus conocimientos también deben poder circular sin fundirse en una única carpeta.

PARA RECORDAR — EL ENFOQUE CODE-CENTRIC

Ninguna especificación se mantiene como descripción duradera del producto: lo que describe su comportamiento se reconstruye desde el código. Una respuesta solo vale por las pruebas que la acompañan y por los límites que señala. El código registra lo implementado; el Contexto guarda el porqué y permite volver a discutirlo.

El día en que tuve tres Contextos

Un técnico me cuenta que tuvo que volver a llamar a un cliente por una información que faltaba en un parte de intervención.

Anoto la frase. Luego me paro, porque no sé dónde ponerla.

- Tengo un Contexto sobre la calidad de los partes de intervención — el que me ha ocupado toda la segunda mitad de este libro.
- Tengo un segundo sobre la intervención misma: lo que contiene, las reglas que la rigen, lo que ya se ha zanjado al respecto.
- Y tengo un tercero, más amplio, abierto unas semanas antes: cómo conseguir que los operarios se hagan cargo de las intervenciones de primer nivel.

La frase del técnico concierne a los tres:

- documenta un parte deficiente,
- dice algo de la intervención,
- aclara la cuestión de los operarios.

Es el momento en que el problema cambia de naturaleza.

- Mientras tuve un solo Contexto, la pregunta era: ¿esto entra o queda fuera del tema?
- Con tres, la pregunta pasa a ser: ¿en cuál?

Las dos maneras de equivocarse

La primera tentación es reunirlo todo.

- Un único espacio grande, todo el conocimiento en el mismo sitio, se acabó la cuestión de dónde guardar las cosas.

Es exactamente de lo que huía. Un espacio único que lo contiene todo recrea el problema de los sistemas de gestión documental: un lugar donde

- la información está presente y la selección vuelve a empezar en cada consulta,
- los objetivos se mezclan,
- las fuentes se acumulan,
- y ninguna nota tiene ya un dueño claro.

Conté ese callejón sin salida al principio del libro; recrearlo con un vocabulario nuevo no lo mejoraría.

La segunda tentación es la contraria: que cada Contexto se quede en su casa. Cada uno es limpio, acotado, coherente — y ciego.

El resultado se ve pronto. Mi Contexto sobre los partes pasó tres sesiones consolidando una regla que el de la intervención ya contenía. No era un duplicado visible: era trabajo rehecho, sin que nadie pudiera notarlo.

Hace falta, pues, una tercera vía, y empieza por saber qué lleva cada Contexto.

Una cifra delante del nombre

Mis Contextos se llaman 01-saisie-compte-rendu, 02-intervention, 03-autonomie-operateurs, 04-exploitation-catch-all.

La cifra no es un rango, es una convención de nombres. Dice qué lleva el Contexto antes incluso de que lo abra.

- **02- una entidad del producto.** Una parte del software que funciona en producción y que tiene sus propias reglas: 02-facturation en un ERP, 02-intervention en el mío. Su contenido está establecido. Ese Contexto permanece activo.
- **01- una hipótesis sobre una entidad.** 01-saisie-compte-rendu: creo que el registro debería cambiar, todavía no sé cómo. Nada de lo que contiene está establecido, así que creo un Contexto 01- para separar bien lo que es estable (en producción) de una hipótesis.
- **03- un problema.** 03-autonomie-operateurs: conseguir que los operarios se hagan cargo de las intervenciones de primer nivel. No es una funcionalidad, atraviesa varias entidades y da lugar a nuevos 01-.
- **04- el cajón de sastre.** Una frase oída, un PDF, una intuición: material cuyo tema todavía no tiene ningún Contexto. Sin él, habría que crear uno por cada fragmento de idea — empecé así y me encontré con más de cincuenta Contextos sin consistencia en unas semanas.

Un 01- acaba siempre por salir del trabajo activo

El día de la puesta en producción, el nuevo registro deja de ser una hipótesis: pasa a formar parte de la intervención.

Lo que describe ahora el producto se propone a 02-intervention. El resto permanece en el Contexto exploratorio,

que después se archiva. Ya no hay dos sitios activos donde buscar cómo funciona una parte de intervención.

Una hipótesis que no llega a nada se archiva junto con lo que la detuvo. No es un fracaso que haya que borrar: es lo que evitará rehacer el mismo camino dentro de dos años.

El 04-, por su parte, se vacía en el otro sentido. El día en que surge un problema, creo el Contexto que le corresponde y luego vuelvo a ver qué había guardado el cajón de sastre al respecto. Lo que no sirve de nada se suprime.

Todo eso resuelve una sola pregunta — dónde pongo esto — separando lo que investigo de lo que está establecido.

- Sin esa separación, cada hipótesis abandonada deja sus huellas en la descripción del producto.
- Y ya no se atreve uno a suponer nada en un sitio donde todo debe ser cierto.

El error que tardé meses en ver

Empecé abriendo 03-tarification. Me parecía evidente: es un tema grande, afecta a varias entidades y perdura en el tiempo. Seis meses después contenía:

- la lista de precios,
- dos análisis de la competencia,
- una discusión sobre los descuentos,
- un ticket de soporte sobre una factura errónea,
- una nota sobre el IVA belga,
- tres capturas de pantalla de un competidor,
- ...

Nada de eso quedaba fuera del tema: todo hablaba efectivamente de tarificación.

Ese es exactamente el problema. Para rechazar un material hay que poder decir «esto no me ayuda». Pues bien, ese Contexto no buscaba nada — **no tenía pregunta**, solo una etiqueta.

03-autonomie-operateurs sí la tiene: cómo conseguir que los operarios se hagan cargo de las intervenciones de primer nivel.

Así que un 03- no es un tema grande. Es una pregunta. La tarificación, por su parte, no ha desaparecido: atraviesa mis Contextos — 02-facturation lleva sus reglas, un 01- puede depender de ella — sin gobernar ninguno.

Mi convención no tiene, pues, casilla para los dominios, ni para muchas otras cosas. Sirve para guardar lo que tiene un sitio evidente, y nada más. El resto espera en el cajón de sastre, y lo recupero describiendo lo que busco en vez de intentar recordar dónde lo guardé.

Ahora tengo dos, una solución que aprendí por ensayo y error. Un único cajón de sastre mezclaba lo que venía de mi trabajo de producto con lo que venía del ámbito de la explotación industrial — dos vocabularios, dos públicos, dos maneras de nombrar los mismos objetos. Se volvió ilegible más deprisa de lo que se vaciaba.

- 04-exploitation-catch-all recibe lo que pertenece al ámbito funcional de mis clientes: la explotación;

-
- 04-productops-catch-all recibe lo relacionado con mi trabajo y su organización: product ops.

Cada uno sigue siendo legible, así que cada uno se vacía.

La frase del técnico, por su parte, no necesitó el cajón de sastre. Fue a 02-intervention, porque ahí es donde vive lo que es cierto del producto con independencia de la pregunta que me hago en este momento.

Queda por saber cómo se enteran de ello mis tres Contextos.

PARA RECORDAR — CUATRO PREFIJOS, UNA SOLA PREGUNTA

La cifra delante del nombre dice qué lleva el Contexto: 02- una entidad que funciona, 01- una hipótesis sobre esa entidad, 03- un problema que atraviesa varias, 04- la zona de espera donde se deposita lo que todavía no se sabe colocar. Un 01- existe para proponer a su 02- lo que describe ahora el producto, y luego para archivar. Si no llega a nada, se archiva junto con lo que lo detuvo. Un dominio no es un Contexto: atraviesa varios sin gobernar ninguno. Un Contexto sin pregunta no sabrá nunca qué queda fuera del tema.

Leer en casa del vecino sin mudarse con él

Retomo 01-saisie-compte-rendu, el Contexto dedicado a la calidad de los partes de intervención. Para avanzar necesito conocer las reglas de la intervención misma: lo que un técnico tiene derecho a anotar, lo que es obligatorio y lo que ya se ha zanjado.

Esas reglas no viven en mi Contexto. Están fijadas en 02-intervention, el Contexto vecino que describe lo que existe realmente en el producto.

Podría copiarlas para tenerlas a mano. Seis meses más tarde tendría probablemente dos versiones de la misma regla y ninguna manera sencilla de saber cuál vale.

Empiezo entonces por una relación más ligera: mi Contexto declara que consulta 02-intervention.

Leer deja el conocimiento en casa de su dueño

Esa relación está escrita en el marco estable de mi Contexto, al lado de su misión y de sus directivas. Abre un derecho preciso: consultar lo que 02-intervention ya sabe.

El material se queda en casa de su dueño. Sigue evolucionando allí, y mi exploración se apoya en él sin pretender poseerlo.

Así puedo comprobar la regla actual sobre los campos obligatorios sin duplicarla en cada estudio que afecte a un parte de intervención. También puedo volver a sus fuentes y a las decisiones que la construyeron.

Leer no autoriza a modificar nada. Si una nota me parece incompleta o falsa, mi Contexto puede formular una propuesta, pero nunca corrige directamente a su vecino.

Esta primera relación responde a una necesidad corriente: usar un conocimiento exterior y dejarlo vivir en el lugar responsable de mantenerlo.

No responde a todos los casos.

Absorber vuelve el conocimiento autónomo

A veces una simple consulta no basta.

Puedo necesitar fijar la regla exacta en la que se apoya una decisión. Puedo querer reformularla para el alcance particular de mi exploración. También puedo necesitar que ese conocimiento siga disponible si el Contexto de origen se archiva, se traslada o se reorganiza.

En ese caso elijo absorberlo.

Absorber hace entrar un conocimiento en mi propio Contexto.

Puedo tomar una nota tal cual o reformularla en mi vocabulario. Según la necesidad, copio también las fuentes que permiten comprobarla, o conservo referencias a ellas.

La copia no se convierte por ello en una idea sin pasado.

Conserva el Contexto del que viene, la fecha de la absorción y lo que se ha reformulado.

La diferencia está, pues, menos en el gesto técnico que en la responsabilidad que resulta de él:

- cuando leo, el conocimiento sigue perteneciendo al vecino;
- cuando absorbo, una versión autónoma entra en mi propio Contexto y me vuelvo responsable de su uso.

Leer crea una dependencia. Absorber crea una responsabilidad nueva.

El movimiento puede volver en el otro sentido

Mi exploración no se limita a consumir lo que 02-intervention ya sabe. Produce a su vez observaciones, hipótesis y a veces un conocimiento duradero.

Al estudiar la calidad de los partes, puedo descubrir que una regla de registro pertenece en realidad a la intervención misma. Ese conocimiento no debería quedarse encerrado en un Contexto exploratorio que quizá acabe archivado.

No lo escribo directamente en 02-intervention. Se lo propongo. El sentido de la circulación acaba de cambiar:

```
01-saisie-compte-rendu — lee → 02-intervention
```

```
01-saisie-compte-rendu — propone → 02-intervention
```

```
      |
      └─ decide si absorbe o no
```

Quien produce un conocimiento no decide su lugar definitivo. El Contexto que lo recibe lo evalúa según su propia misión. Puede:

- tomar la nota tal cual;

- reformularla en su vocabulario;
- conservar una divergencia cuando las dos lecturas siguen siendo válidas;
- rechazar la absorción cuando lo que es útil a la exploración no describe la intervención misma.

El rechazo no es un fracaso de la circulación. Es lo que impide que un resultado local se convierta silenciosamente en una verdad general.

Cuando la exploración se vuelve una realidad del producto

Unas semanas más tarde, el nuevo registro de los partes se pone en producción. 01-saisie-compte-rendu ha cumplido su papel: lo que solo era una hipótesis se ha vuelto una realidad del producto.

No por ello traslado todo su contenido a 02-intervention.

Repaso lo que la exploración ha producido y separo dos conjuntos.

El primero contiene lo que describe o explica ya el producto:

- las notas que establecen las nuevas reglas de registro;
- el vocabulario fijado para nombrarlas;
- las decisiones fundamentales y las razones de esas elecciones;
- las contradicciones que siguen abiertas tras la puesta en producción;
- las fuentes o las referencias necesarias para comprobar ese conocimiento.

Esos elementos se proponen a *02-intervention*. El Contexto estable no los acumula a ciegas: enriquece sus notas existentes, reformula lo que debe reformularse y solo crea una nota nueva cuando una idea es realmente nueva.

En nuestro ejemplo, la regla que limita el cierre a cuatro datos pasa a *02-intervention* con la decisión que la explica. Ahora describe una realidad del producto.

El segundo conjunto cuenta la historia del trabajo sin describir el producto actual:

- las hipótesis abandonadas;
- las maquetas y las alternativas descartadas;
- los borradores intermedios;
- las tareas propias de la exploración;
- las discusiones que permiten entender el camino recorrido.

Esos elementos se quedan en *01-saisie-compte-rendu*. No se han vuelto inútiles: explican lo que se intentó, por qué se descartaron ciertos caminos y lo que no habrá que volver a empezar sin una razón nueva.

Una vez terminada esa selección, el Contexto exploratorio se archiva en su conjunto. Sigue siendo legible y puede retomarse más tarde, pero ya no pretende describir el trabajo activo.

Si la exploración no hubiera llegado a nada, también se habría archivado, con las razones de su abandono, sin imponer nada a *02-intervention*.

Así que la puesta en producción no fusiona dos Contextos. Desencadena una revisión: lo que describe ya el producto pasa al Contexto estable; lo que cuenta la exploración se queda en el Contexto archivado.

Esa selección protege a ambas partes:

- la exploración sigue siendo libre de equivocarse;
- el Contexto estable solo recibe lo que elige reconocer como suyo.

Las relaciones forman parte del marco

Leer, absorber y proponer no deben quedar como gestos implícitos.

El marco estable del Contexto indica:

- qué otros Contextos puede consultar;
- qué autoriza esa consulta;
- según qué reglas puede absorberse un conocimiento;
- cómo se conservan su origen y las posibles reformulaciones.

Esa gobernanza no requiere una infraestructura pesada.

Simplemente da un lugar explícito a las relaciones para que la procedencia, las responsabilidades y las fronteras sigan siendo visibles.

Así que la red no se construye conectándolo todo con todo. Se construye relación a relación, cada vez que un trabajo necesita de verdad a un vecino.

- Leer en casa del vecino evita duplicar lo que él sabe.
- Absorber permite volver autónomo lo que debe vivir en mi casa.

-
- Proponer permite, por último, que el vecino decida si quiere integrar lo que mi exploración ha producido.

Esa circulación parece limpia sobre el papel. Tiene sin embargo un precio: una referencia puede envejecer sin avisar, una copia puede perder su origen y dos Contextos pueden conservar lecturas distintas.

Es lo que va a examinar el capítulo siguiente.

PARA RECORDAR — TRES GESTOS, TRES RESPONSABILIDADES

Leer consulta un conocimiento que se queda en casa de su dueño; absorber crea una versión autónoma, acompañada de sus fuentes, en el Contexto que la recibe. Una exploración propone lo que ha aprendido, pero el dueño del Contexto de destino decide lo que entra en su casa. La puesta en producción no fusiona las carpetas: el Contexto estable absorbe lo que describe ya el producto, mientras que la exploración conserva el resto antes de ser archivada.

Lo que cuesta la circulación

Han pasado seis semanas.

Mi trabajo sobre los partes de intervención está listo para salir a revisión, y alguien detecta una incoherencia. La regla en la que me apoyo — la que dice qué campos son obligatorios al cierre — ya no es la del producto.

Ha cambiado. Mientras yo trabajaba, otra exploración llegó a su fin y el Contexto de la intervención absorbió sus resultados. Nadie me lo dijo. No volví a leerlo, porque lo había leído seis semanas antes y nada avisó de que había que regresar.

El capítulo anterior plantea una regla que parece sencilla: se referencia en el sentido de la lectura, se copia en el sentido de la absorción. La di por una solución definitiva.

No suprime el riesgo. Lo desplaza, y más vale saber dónde recae.

Una dependencia que no se ve

Lo que acaba de pasar tiene un nombre: una dependencia ciega.

El vínculo está vivo, así que lo que leo puede cambiar sin mí. No es ese el problema — a menudo es exactamente lo que quiero. Una regla que evoluciona en casa de su dueño es el sistema funcionando.

El problema es no saber que ha cambiado.

Lo que falta es fácil de enunciar: nada supervisa lo que leo. Mi Contexto sabe de quién depende; no sabe cuándo ha cambiado aquello de lo que depende. Es la mitad del mecanismo, y es precisamente la parte costosa.

Una copia me habría protegido de eso. El Contexto que recibe se vuelve autónomo: ya no depende ni de la existencia ni de la evolución de su fuente.

Pero pierde otra cosa a cambio. Si nada dice de dónde viene la copia, la procedencia desaparece — y un año después tengo una nota correcta de la que ya no sé en qué se apoya. Es precisamente la situación que toda la trazabilidad de este libro busca impedir.

Así que la conclusión no es elegir un bando. Es registrar lo que cada acción debe conservar:

- una **referencia** exige saber cuándo cambia su fuente;
- una **copia** exige decir de dónde viene — qué Contexto, en qué fecha, y qué se reformuló por el camino.

EVOLUCIÓN POSIBLE

Otra respuesta, todavía por llegar, será tecnológica: los Contextos podrán avisar a sus observadores cuando cambie su estado, sobre todo cuando el cambio afecte al alcance que supervisa otro Contexto. El Contexto es un tema vivo en el que quedan muchas posibilidades por explorar.

Los derechos que la lectura no da

Cuando leo mucho un Contexto, acaba pareciéndome un poco mío. Varias veces me han dado ganas de corregir directamente una nota en casa del vecino — estaba incompleta, yo sabía qué poner y me habría llevado treinta segundos.

La regla no cambia por ello: lo que tengo que decirle es una propuesta, y la decisión pertenece a quien recibe.

Eliminar sin romper la trazabilidad

La eliminación exige una regla aparte, y la aprendí vaciando una zona de espera.

Cuando una nota cambia de Contexto, las fuentes que la produjeron se quedan detrás. Pueden quedarse solas, sin ninguna nota que se vincule ya a ellas.

El reflejo es hacer limpieza. Solo es acertado en parte.

Una fuente ya leída y convertida en notas puede eliminarse: lo que llevaba vive en otra parte. Una fuente que nadie ha leído todavía no puede eliminarse.

- Lleva un material que nadie ha extraído, y tirarla equivale a perder lo que ni siquiera se ha abierto.

Así que el criterio no es ni la edad del archivo ni el espacio que ocupa.

- Hay que saber si la fuente ya se ha leído y comprobar qué la sigue citando.

Hay además una condición, y no se negocia. Antes de suprimir, comprobar que ninguna nota, ninguna entrada de

vocabulario, ningún borrador y ninguna línea del registro cita ya esa fuente. Sin esa comprobación, la regla rompe la trazabilidad que pretende proteger.

Una divergencia no es un error

Queda el caso en que dos Contextos no dicen lo mismo del mismo tema.

El reflejo es ver en ello una incoherencia y zanjar deprisa para restablecer el orden. A menudo, esa conclusión es errónea.

Una divergencia puede ser una diferencia de alcance: cada uno puede tener razón dentro de sus propios límites. Puede ser una hipótesis todavía abierta que uno investiga y el otro ignora. Pueden ser dos lecturas que se sostienen las dos, porque miran el tema desde dos sitios.

Merece entonces conservarse como tal, y le corresponde a la persona decidirlo. Lo que el sistema debe garantizar es que se sepa que existe.

6

Una divergencia olvidada siempre reaparece en el peor momento: cuando alguien se apoya en una de las dos versiones sin saber que la otra existe. Es lo que me pasó con la regla de cierre, con la diferencia de que no eran dos lecturas sino una sola que había envejecido.

Mantener la red fiable tiene un coste, y prefiero nombrarlo antes que descubrirlo en revisión. Enlazar dos Contextos no suprime el trabajo de vigilancia: lo hace posible, y lo desplaza hacia el humano.

Es, en el fondo, lo que este libro cuenta desde el principio.

PARA RECORDAR — EL PRECIO DE LA RED

Una referencia puede envejecer sin avisar, una copia puede perder su procedencia: cada gesto tiene una deuda, y hay que escribir cuál. Leer no autoriza a escribir; lo que se tiene que decir al vecino es una propuesta, nunca una corrección. Una fuente ya convertida en notas puede desaparecer; una fuente nunca leída no puede — y nada se elimina sin comprobar quién la cita todavía. Dos Contextos que se contradicen no están forzosamente equivocados: lo grave es ignorarlo.

Antes de cerrar

Cierre

Cierre

El libro se detiene aquí, pero el tema no se detiene con él.

El epílogo dice en qué punto está este trabajo, qué no ha hecho más que rozar y dónde aparecerá su continuación. Los parentescos sitúan el Contexto entre los enfoques que lo precedieron o que lo rodean, para que sepas qué toma prestado y qué no retoma.

Epílogo

Gracias por haber llegado hasta aquí.

He intentado hacer este libro lo más legible posible, y no lo he conseguido del todo. Quedan pasajes más áridos de lo que habría querido, sitios donde la mecánica se impone al relato. Has aguantado igualmente: es un regalo que no me tomo a la ligera.

Este libro solo ha rozado el tema

A pesar de su grosor, aborda el Contexto solo en la superficie. Varias partes siguen siendo ligeras en el plano conceptual, y lo sé al publicarlas. Las notas atómicas, por ejemplo: según cómo las implemente, obtengo resultados muy desiguales. Lo que funciona en un Contexto exige adaptaciones en cuanto se pasa a un LLM Wiki, donde el material no tiene ni el mismo volumen ni el mismo uso.

Esas zonas merecen reforzarse. No lo han sido aquí porque un libro tiene que terminar en algún momento, no porque el tema esté cerrado.

Un trabajo que seguirá siendo un proyecto paralelo

El Contexto seguirá avanzando, pero siempre como side project.

Es una actividad llevada al margen de mi oficio, y ya le he dedicado demasiadas tardes. Volveré a él — siempre vuelvo — pero ya toca bajar el ritmo y disfrutar de la vida.

Lo que me gustaría explorar después

La búsqueda de antecedentes sí la hice. Sacó a la luz conceptos muy cercanos al mío, a veces en campos alejados del producto y de la gestión del conocimiento, a menudo en otras lenguas: el japonés y el chino sobre todo, cuyos trabajos siguen siendo en gran medida invisibles desde el mundo francófono y anglófono. He escrito un artículo sobre esa búsqueda; aparecerá en el blog.

Lo que falta es el paso siguiente: analizar esos trabajos.

Los nombres están identificados, los artículos están ahí, y no los he leído. Son décadas de investigación hechas por cientos de personas sobre preguntas que yo abordé solo, por las noches. Estudiarlos a fondo reforzaría este marco mucho más que seguir inventándolo.

Lo que venga después no será un tomo II

Este libro ha servido para cristalizar el concepto. Era su papel: reunir lo que estaba disperso, hacerlo transmisible, darle una forma lo bastante estable como para poder discutirla.

Durante un tiempo, las novedades se publicarán en malorean.net, y no en un tomo II ni en una nueva versión de esta obra. Un artículo se corrige, se completa, se contradice. Un libro fija — y lo que se describe aquí todavía no está lo bastante maduro como para fijarse dos veces.

Si quieres seguir lo que viene, es ahí donde se escribirá.

Parentescos y vecindades

Contexto no apareció en un vacío intelectual.

La mayoría de los mecanismos que lo componen tienen parientes, algunos muy antiguos, en la gestión del conocimiento, los métodos de toma de notas, la ingeniería del software, los sistemas de ayuda a la decisión, las ciencias cognitivas o la memoria organizativa.

Algunas de esas influencias fueron conscientes desde el principio. Otras solo se descubrieron después de diseñar el método, durante una búsqueda de antecedentes hecha con el libro casi terminado.

Hay que distinguir, pues, dos cosas.

Una **influencia** es un método o una idea que yo conocía y de los que retomé o adapté conscientemente algunos principios.

Un **parentesco** o una **vecindad** designa un sistema anterior que resuelve un problema cercano, a veces con mecanismos sorprendentemente parecidos, sin haber influido directamente en Contexto.

Esa distinción es importante. Dos ideas pueden ser cercanas sin que una derive de la otra.

Zettelkasten: el conocimiento como red de unidades reutilizables

El parentesco con el Zettelkasten es probablemente el más evidente.

En el sistema de Niklas Luhmann, una nota no se guarda simplemente en una carpeta correspondiente al proyecto del momento. Entra en un sistema duradero de notas enlazadas entre sí y puede reutilizarse en nuevos razonamientos.

Contexto comparte con fuerza esa idea: un conocimiento producido durante un trabajo no debería desaparecer con su entregable.

Las notas atómicas del Contexto pertenecen claramente a esa familia.

La diferencia principal está en su estatuto. Una nota atómica debe conservar su procedencia y se inscribe en una arquitectura que distingue explícitamente fuentes, contradicciones, decisiones, estado actual, síntesis y entregables.

El Zettelkasten aporta, pues, un parentesco muy fuerte para la **construcción progresiva de un capital de conocimientos**, pero no para el conjunto del sistema.

PARA: organizar la información en función de la acción

PARA, de Tiago Forte, forma parte de las influencias conscientes de Contexto.

PARA organiza la información en cuatro categorías: Projects, Areas, Resources y Archives. Una de sus intuiciones

importantes es que una organización de la información debe servir a la acción y no a una clasificación abstracta del mundo.

Contexto conserva algo de esa intuición.

Un Contexto no existe porque exista un tema. Existe porque un humano persigue una misión.

La frontera no es, por tanto: «todo lo que sé sobre bases de datos».

Podría ser: «elegir una arquitectura de datos para el producto X».

La diferencia está en que los Contextos mismos se convierten en unidades de trabajo autónomas, con su propio estado, sus fuentes, sus conocimientos, sus decisiones y su ciclo de vida.

Domain-Driven Design: la frontera antes que el contenido

El propio término Contexto evoca de forma natural el **Bounded Context** del Domain-Driven Design.

El parentesco se apoya sobre todo en una idea: una información o un concepto solo tienen sentido dentro de cierta frontera.

La misma palabra puede significar cosas distintas en dos contextos distintos sin que haga falta fusionarlos.

Contexto traslada esa intuición.

Pero, en lugar de hacer depender la frontera de un modelo de negocio coherente, la hace depender de una **misión humana**.

Dos Contextos pueden, por tanto, conservar comprensiones distintas del mismo tema porque persiguen objetivos distintos.

Git: divergencia, autonomía y reintegración

Git es una influencia más directamente técnica.

Cuando una exploración cobra bastante importancia, puede ser preferible hacerla evolucionar en un Contexto aparte en lugar de perturbar el espacio estable.

Esa lógica se parece a una rama.

La exploración puede divergir, producir sus propias fuentes y conocimientos, y luego algunas conclusiones pueden reintegrarse en el espacio de origen.

Pero, como con Git, integrar no significa necesariamente fusionarlo todo.

Esa intuición influyó también en las relaciones entre Contextos: autonomía local, divergencia posible e integración selectiva.

Project Memory: conservar el porqué del proyecto

Los trabajos sobre la **memoria de proyecto** son un parentesco que no había identificado al diseñar Contexto.

El problema que plantean es casi exactamente el que aparece cuando un trabajo dura varios meses: las organizaciones conservan con bastante facilidad los documentos finales, pero mucho peor el contexto que explica cómo y por qué el proyecto llegó a ese resultado.

¿Por qué esta decisión?

¿Qué habíamos probado antes?

¿Qué restricciones existían en ese momento?

¿Qué debería entender una persona nueva antes de continuar?

Contexto responde a esas mismas preguntas con varias memorias distintas: el estado actual, la historia, las decisiones, la síntesis y el handoff.

La proximidad funcional es, pues, muy fuerte.

Experience Factory: el entregable no es el capital

La Experience Factory, desarrollada en torno a los trabajos de Victor Basili y ensayada sobre todo en la NASA, aporta un parentesco aún más importante.

Distingue la organización que realiza un proyecto de la que transforma la experiencia del proyecto en conocimiento reutilizable para los proyectos siguientes.

Esa idea coincide directamente con una distinción fundamental de Contexto:

el entregable no es el capital.

Un proyecto puede producir un resultado excelente y no dejar casi nada detrás.

A la inversa, su experiencia puede analizarse, transformarse y conservarse de modo que aumente la capacidad de resolver los problemas siguientes.

Contexto integra esa capitalización directamente en el espacio donde se hace el trabajo.

Knowledge-Centered Service: producir el conocimiento trabajando

Knowledge-Centered Service — KCS — es probablemente uno de los vecinos más interesantes de Contexto.

Su principio es que la documentación no debe ser una actividad aparte realizada después del trabajo.

Cuando se resuelve un problema, se busca y se reutiliza el conocimiento existente. Si es imperfecto, se mejora. Si no existe, se crea.

El simple hecho de trabajar se convierte así en un mecanismo de creación y de mantenimiento del conocimiento.

Es exactamente una de las ambiciones de Contexto:

cada uso del sistema debería poder aumentar su capital informativo.

KCS está pensado, eso sí, sobre todo para entornos de servicio y de soporte, mientras que Contexto busca ofrecer una estructura genérica para muchas misiones distintas.

HWMSE: humanos, máquinas y conocimientos en un mismo espacio

El parentesco más sorprendente descubierto durante la búsqueda viene probablemente de China.

El **Hall for Workshop of Meta-Synthetic Engineering**, o HWMSE, se desarrolla en torno a los trabajos de Qian Xuesen a partir de comienzos de los años noventa.

Está concebido para tratar problemas complejos combinando explícitamente:

- expertos humanos;
- sistemas máquina;
- sistemas de conocimientos.

Más interesante todavía, los trabajos sobre HWMSE no hablan solo de usar conocimiento existente. Hablan también de **creación de conocimiento** y de un sistema de conocimiento propio del problema estudiado que se enriquece continuamente mientras se usa.

La proximidad con *Contexto* es, por tanto, importante:

- una actividad orientada a un problema;
- una colaboración humano-máquina;
- recursos y modelos movilizados durante el trabajo;
- conocimiento producido por la actividad;
- un capital que se acumula.

HWMSE no tiene, sin embargo, la misma gramática operativa que *Contexto*. No se encuentra allí, con esta forma, la separación entre fuente, nota atómica, estado, síntesis, contradicción, handoff y entregable.

Pero, entre los sistemas estudiados, es probablemente uno de los vecinos arquitectónicos más cercanos.

ScholOnto y SWAN: una proposición no es su fuente

Otro parentesco fuerte aparece en los sistemas dedicados al discurso científico.

ScholOnto y, más tarde, SWAN distinguen los documentos científicos de las **proposiciones producidas a partir de esos documentos**.

Un recurso puede servir de origen a una afirmación.

Esa afirmación conserva su procedencia.

Puede ser sostenida, discutida o contradicha por otros elementos.

Es extremadamente parecido al principio que sustenta las notas atómicas del Contexto:

*la fuente sigue siendo una fuente; el conocimiento
construido a partir de ella es otro objeto.*

Esos sistemas muestran también que una proposición registrada en un sistema de conocimientos no debe considerarse automáticamente verdadera.

Tiene una procedencia y un estatuto dentro de un razonamiento.

Es un parentesco especialmente fuerte con la arquitectura epistémica de Contexto.

Sistemas federados: leer no es copiar

Por último, algunas ideas sobre las relaciones entre Contextos tienen parientes sorprendentemente antiguos en los sistemas de bases de datos federadas.

Esas arquitecturas deben permitir que varias bases autónomas cooperen sin renunciar a su autonomía.

Un sistema puede exponer cierta información.

Otro puede consultarla a distancia o decidir guardar una copia local.

Y en cuanto se crea una copia aparece el problema clásico: ¿qué ocurre cuando cambia el original?

Es exactamente lo que resume en Contexto la fórmula:

referencia = dependencia; copia = responsabilidad.

Los mecanismos informáticos existían, pues, desde hacía mucho.

Contexto los traslada a reglas de gobernanza del conocimiento entre espacios humano-IA.

Una síntesis más que una invención de la nada

Esas vecindades hacen difícil presentar Contexto como la invención de mecanismos enteramente nuevos.

- No es lo que muestra la historia.
- La procedencia existía.
- Las notas atómicas existían.
- La memoria de los proyectos existía.
- La capitalización entre proyectos existía.
- La creación de conocimiento durante el trabajo existía.
- Los entornos que combinan humanos, máquinas y sistemas de conocimientos existían.
- Las aserciones con fuente y discutibles existían.
- Las arquitecturas autónomas capaces de compartir o copiar información existían.

Lo que parece caracterizar más a Contexto es la manera en que esos mecanismos están **seleccionados, combinados y normalizados dentro de un mismo método**.

El método les da una gramática común:

- fuentes;
- notas atómicas;
- contradicciones;
- decisiones;
- estado;
- historia;
- síntesis;
- acciones;
- handoff;
- entregables;
- relaciones entre Contextos.

Y sobre todo, aplica ese ensamblaje a un problema muy actual: mantener una colaboración duradera entre un humano y una IA sin convertir cada conversación en un episodio aislado y amnésico.

Contexto tiene, por tanto, numerosos parientes.

Pero, como suele ocurrir, conocer a los propios parientes sirve sobre todo para entender mejor lo que uno tiene de propio.