

DATOS TÉCNICOS

Herramientas y métodos del canalero

Edición de junio de 2026

David Bories

A prueba del flujo — Una ontología procesual del dato, volumen de continuidad
Datos técnicos — Herramientas y métodos para un canalero

Concebido y dirigido por David Bories — Albi, Francia.

Redactado utilizando la inteligencia artificial generativa.

Traducción realizada mediante inteligencia artificial generativa, bajo la supervisión del autor.

Octubre de 2026.

Acorde con la segunda edición de la trilogía.

Primera versión: julio de 2026.

Versión 0.13.0-es, formato de bolsillo.

© David Bories, 2026. Difundido bajo licencia Creative Commons Atribución 4.0 Internacional (CC BY 4.0). Se autoriza todo uso, incluido el comercial, siempre que se cite al autor y la fuente.

<https://creativecommons.org/licenses/by/4.0/deed.es>

El texto íntegro de este volumen y de todo el corpus es de libre acceso en <https://artisanat-de-la-donnee.fr>

Índice

Preámbulo — Un manual de forma	1
Lo que este libro es, y lo que no es	1
Para quién	2
Cómo está organizado este manual	2
La regla de este manual en una frase	3
Capítulo 1 — El canal, el canalero, la obra	5
El río, el canal, el dato	5
La intención y los criterios	5
El óptimo observacional	6
Los tres tiempos, los tres gestos	6
La obra del canalero	7
La actitud de referente	7
Capítulo 2 — El método hexagonal: cavar lo perenne	9
El principio: aislar el núcleo	9
Por qué es la herramienta de la perennidad	10
La infraestructura obedece a los puertos	10
El gesto del canalero: modelizar antes de codificar	11
La organización concreta de un proyecto hexagonal	13
Lo que el hexágono no aísla por sí solo	13
Capítulo 3 — El núcleo sin dependencias: HTML, CSS, JavaScript	15
Por qué lo nativo en lugar de un framework	15
Los tres materiales, y lo que lleva cada uno	16
Aplicación web, aplicación instalable: el mismo núcleo	16
El almacenamiento local del núcleo	17
Varios bancos de trabajo, una sola obra	17
Conectar el mundo exterior — siempre por el hexágono	17
Una nota de honestidad	18
Capítulo 4 — La gamificación: volver la obra agradable y atractiva	19
El principio: transformar el dato consolidado en señal motivadora	19
Las mecánicas, y el gesto de oficio al que sirven	19

Los umbrales: el puente entre gamificación y dirección	20
La salvaguarda: gamificar sin falsear	21
La experiencia fluida, más allá de las mecánicas	21
Capítulo 5 — Fabricar el dato: toma y anotación	23
Cartografiar la materia: qué datos fabricar	23
Regular el plazo límite de uso	24
El armazón y la toma: el cauce y el agua	24
La lista de adyacencia: sostener el árbol	25
Las dos aristas: la sustentación y la referencia	25
Anidar o enumerar: el criterio es la búsqueda	26
El armazón se mantiene	26
Las herramientas de toma	27
Las herramientas de anotación	29
Regular la toma en el óptimo	29
Documentar los propios instrumentos	30
Capítulo 6 — Hacer circular: fijación y transporte	31
Las herramientas de fijación: dónde se sostiene el dato	31
El sistema de eventos: hacer circular el acto de inscripción	32
El transporte local: dirigir la propia actividad	33
El transporte remoto: compartir el dato	34
La multiplicación: un hecho compartido ya no es un objeto único	35
Reutilizar sin transmitir: el momento objetual, y cómo lo sostiene el fondo	35
El flujo entrante: tomar el correo	37
Capítulo 7 — Leer el dato: el cuadro de mando de dirección	39
El lector está situado, su ventana es finita	39
La selectividad: menos, pero lo justo	39
Los indicadores: elegir los adecuados	40
La frescura de la lectura: irrigar el cuadro de mando	40
Las herramientas de lectura, y sus sesgos	41
Las tres posturas del lector, y lo que dictan a la interfaz	42
Mantener la capacidad de leer	42
La ventana nombrada, en la firma	42
Capítulo 8 — La IA local: el operador semántico del canalero	45
Lo que la IA generativa sabe hacer para el emprendedor en solitario	45
Por qué local: soberanía, costo, dominio	46
Combinar la IA y las reglas deterministas	46
El método de automatización progresiva	47
La contrapartida, nombrada	47
Capítulo 9 — Mantener: mantenimiento, seguridad, rituales	49
Las herramientas de mantenimiento	49
La seguridad y la higiene digital	50
La reserva: el doble retirado del ciclo de uso	50
Los rituales del taller	51

La firma de la obra	52
Documentación viva: el mantenimiento por el cliente	52
Capítulo 10 — El taller digital: el material y los programas	55
El material	55
Los programas	56
El taller como lugar, físico o virtual	58
La articulación taller / cadena	58
Capítulo 11 — Itinerario de equipamiento: ensamblar las herramientas	59
El procedimiento, siempre el mismo: tres gestos	59
El itinerario tipo, etapa por etapa	59
Tres ensamblajes ilustrados	61
El hilo que lo sostiene todo	62
Capítulo 12 — El taller de agentes de IA: industrializar el itinerario	63
La base técnica: no rehacer lo probado	63
Los Agentes IA	64
Dos objetos que no hay que confundir: el taller de agentes de IA y la obra	64
El agente de programación: el banco de trabajo de los agentes	64
El repositorio: copia de seguridad, memoria y base distribuida	65
Los puertos soberanos: extender sin depender	65
La dirección del taller: lo determinista primero, el humano en los hitos	65
La aplicación práctica	66
Lo que, en este capítulo, no caduca	66
Anexo — La caja de herramientas del canalero	67
A. Las seis familias de herramientas, de un vistazo	67
B. Los métodos transversales	68
C. Lista de comprobación del flujo de observación encontrado	68
D. Rejilla de caducidad	68
E. Modelo de revisión del canal (orden del día)	69
F. Ficha de firma de obra	69
G. Recordatorio de material y programas	69
H. Las salvaguardas, en siete frases	70
I. Tabla de correspondencias — el vocabulario del corpus y los marcos existentes (forma fechada, julio de 2026)	70
Decisiones de traducción	71

Preámbulo — Un manual de forma

Nota sobre la traducción — Esta traducción re-fija el texto fuente sin desplazar su fondo. Algunos términos del corpus no tienen un equivalente evidente en español; las decisiones adoptadas, y la lectura que debe hacerse de ellas, se exponen en la sección «Decisiones de traducción», al final del volumen. Esta traducción no está congelada: evolucionará a medida que lleguen los comentarios de los lectores y se perfeccionen los medios de traducción. La versión francesa hace fe.

«El concepto es una lámpara; no dibuja la carta de los arrecifes.» Toda la trilogía *A prueba del flujo* se atuvo a esta regla: decir lo que es verdad del dato por mucho tiempo, y dejar a otros la tarea de determinar lo que hay que hacer con ello hoy. El presente libro hace la apuesta inversa, y la asume. Es un libro de **forma**.

Allí donde *El río y el canal* planteaba que hay que cavar canales en el flujo de lo real, allí donde *El artesano del dato* nombraba al canalero y sus seis familias de herramientas sin prescribir su uso concreto, este manual baja un escalón: dice **con qué** y **cómo** se cava, hoy, en 2026, un canal digital. Qué lenguajes, qué máquinas, qué arquitecturas, qué gestos. Es el banco de trabajo del canalero, abierto y surtido.

Lo que este libro es, y lo que no es

Este libro **es** un manual de uso. Describe herramientas materiales y de software disponibles ahora, métodos de fabricación probados, y la manera de ponerlos al servicio de un solo fin: ofrecer al cliente —el emprendedor en solitario, el empresario individual— una gestión de actividad fluida, legible, dominada, y la serenidad que viene con ella.

Este libro **no es** una doctrina. Todo lo que nombra está fechado. Los lenguajes citados vivirán y morirán; los modelos de IA locales de hoy serán sustituidos; las versiones cambiarán. No tiene importancia, con una condición: que el lector sepa distinguir permanentemente **el fondo** (lo que no caduca —la actitud de referente, la exactitud de la toma, la responsabilidad de la obra—) de **la forma** (las herramientas, los

lenguajes, las plataformas). El fondo se hereda de la trilogía; este manual no lo repite, se sirve de él. La forma es lo que añade, sabiendo que es revisable.

Cuando este manual escribe «JavaScript», «Docker» u «Ollama», hay que leer esas palabras como se leería, en un texto de 1950, «un automóvil puede alcanzar los 100 km/h»: verdadero en la fecha de escritura, accesorio para el razonamiento, destinado a ser superado. Aquello a lo que se compromete el manual es el *principio* al que sirve la herramienta —la separación del núcleo de negocio y de la infraestructura, el control local, la perennidad del soporte—, no la marca de la herramienta.

Para quién

Para el **canalero** —el artesano del dato (un oficio por crear)— que quiere equipar su taller digital. Para el aprendiz que aprende los gestos. Para el emprendedor en solitario curioso que quiere comprender lo que hace su herramienta, y por qué. El manual da por adquirida la trilogía: solo vuelve a ella en recordatorios apretados, allí donde es necesario para tener claro un gesto.

Cómo está organizado este manual

El libro sigue la estructura que da el *Cuaderno del taller*: las **seis familias de herramientas** del canalero, ordenadas según los **tres tiempos del dato**.

- **Fabricar el dato** (tiempo 1): toma y anotación. Cómo la actividad del cliente produce un dato exacto, captado en el óptimo.
- **Hacer circular el dato** (tiempo 2): fijación y transporte. Cómo el dato se sostiene, viaja en local para dirigir, y a lo lejos para compartirse.
- **Leer el dato** (tiempo 3): lectura. Cómo un cuadro de mando vuelve practicable la parte del lector.
- **Mantener**: mantenimiento. Cómo un canal sigue vivo.

Antes de estas familias, dos capítulos plantean los **métodos** que las atraviesan todas: la **arquitectura hexagonal**, que vuelve perenne la obra aislando el núcleo de negocio; y la **gamificación de la experiencia**, que vuelve la obra agradable y atractiva. Después de ellas, un capítulo reúne el **taller** del canalero —el material, el software, los rituales— y un último propone **itinerarios de equipamiento** progresivos.

Cada capítulo se sostiene solo. Se puede entrar por la herramienta que se necesita. Pero el orden tiene un sentido: solo se lee bien lo que se ha sabido fabricar y hacer circular.

La regla de este manual en una frase

Las herramientas cambian; el oficio —tomar con exactitud, volver legible, hacer autónomo— permanece. Este manual arma la mano para hoy; a la mano le corresponde saber que deberá volver a aprender mañana.

Capítulo 1 — El canal, el canalero, la obra

Este capítulo no enseña nada nuevo a quien ha leído la trilogía. Reúne, en una lengua apretada, los puntos de referencia de los que se sirve cada herramienta de este manual. Se puede volver a él de un vistazo durante la lectura.

El río, el canal, el dato

No hay, en lo real, datos que esperan a que se los recoja. Hay un **flujo de lo real** —lo que sucede, permanentemente, indiferente al humano, que nadie ha cavado y que nadie detiene—. Tomar es construir un **flujo de observación**: un canal que el humano desvía del río por su intención, según criterios que ha elegido. El canal es infinitamente más pobre que el río; solo contiene lo que su trazado ha admitido. Pero el canal, en cambio, el humano lo domina: puede seguirlo, remontarlo, reproducirlo, enlazarlo con otros, confiarlo a un vecino, cegarlo cuando ya no le sirve.

El **dato de observación** es el grano de agua que se toma de ese canal. Un dato no se recoge; se cava un canal, y el dato es lo que fluye de él. Otros datos nacen de otro modo: una factura emitida, un presupuesto firmado se instituyen por un acto, bajo una regla; un indicador es derivado, calculado a partir de otros datos.

Para el emprendedor en solitario, el río es su actividad que se desarrolla: trabajos, desplazamientos, presupuestos, llamadas, entregas, horas, kilómetros, existencias que suben y bajan. Todo eso fluye, y se pierde, si nada lo capta. El canalero cava, en esa actividad, los canales que traen de ella lo que la dirección necesita —y nada más—.

La intención y los criterios

Un canal nace de una **intención**: seguir un tema, responder a una pregunta, dirigir una decisión. De la intención se siguen los **criterios**, que responden a cinco preguntas:

- **¿Qué?** ¿Qué hechos brutos de la actividad se quiere tomar?
- **¿Con qué cadencia?** ¿Con qué frecuencia se toma?
- **¿Con qué fidelidad?** ¿Qué precisión, qué instrumento, qué margen de error aceptable?
- **¿Con qué corpus?** ¿Qué datos de contexto acompañan cada toma para volverla interpretable (unidades, referencias, fecha)?
- **¿Sobre qué perímetro?** ¿Qué porción de la actividad se observa, y cuál se deja de lado?

Estos criterios nunca son neutrales: son la firma de la intención. Ante todo canal, la primera pregunta no es «¿qué dice?» sino «¿qué intención lo hizo nacer?». Un canal nunca miente sobre lo real; solo muestra lo que sus criterios le permiten mostrar.

El óptimo observacional

Más no es mejor. Entre dos escollos simétricos —la **subtoma**, que pasa por debajo del umbral y pierde la señal, y la **sobreingeniería**, que toma mucho más allá de lo necesario y no gana nada salvo costo— existe una zona en la que el canal está cavado lo mejor posible. Su punto central es el **óptimo observacional**: el reglaje conjunto de la cadencia, del corpus y de la probidad que maximiza la diferencia favorable entre el beneficio (la calidad del resultado) y el costo (toma, transporte, almacenamiento, cálculo, tiempo).

No existe un óptimo universal. La pregunta que lo determina es siempre la misma: *¿puede la actividad observada cambiar significativamente entre dos tomas?* Si es así, la cadencia es demasiado lenta. Si no, es inútilmente rápida. El óptimo vuelve a plantearse en cada misión. Es la regla de oro del canalero digital: no sobreequipar nunca a un cliente, ni dejarlo ciego.

El umbral de Nyquist, en claro. Para que una serie de tomas restituya fielmente un movimiento, hay que tomar sensiblemente más deprisa de lo que cambia la cosa. Unas existencias que varían cada día no pueden seguirse con un inventario anual. Una cifra de negocio mensual no se dirige al minuto. Ajustar la cadencia a la velocidad de la actividad: he ahí el primer reglaje.

Los tres tiempos, los tres gestos

El dato tiene tres tiempos, planteado cada uno por un tomo. El canalero digital los dota de herramientas a los tres.

1. **La fabricación** (tomo 1) —el **acto de inscripción**: detener un valor en un flujo que no se detiene, y confiárselo a un soporte—. Tres componentes: *qué* (el objeto), *cuándo* (el instante, la cadencia), *cómo* (el instrumento, el criterio).

2. **La circulación** (tomo 2) —el **acto de intercambio**: poner la obra en movimiento—. El transporte no es el depósito de un dato en alguna parte; es una relación entre dos desplazamientos (el dato hacia el lector, el lector hacia el dato), cumplida solo cuando la parte del lector es practicable.
3. **La recepción** (tomo 3) —el **acto de lectura**: *percibir* → *distinguir* → *descifrar*—. Sin lectura, el dato permanece constituido, pero nada se actualiza. Una obra entregada que nadie puede leer es una obra inacabada.

La obra del canalero

La obra no es un dato aislado. Es un **canal en el óptimo** —cavado, dimensionado, mantenido, que restituye fielmente para el uso previsto lo que fue concebido para captar— y, al final de la cadena, un **dato consolidado**: la síntesis que responde a la pregunta de uso (un indicador, un estado, una tendencia, una alerta). Una obra se reconoce por cuatro rasgos:

- está **a la medida del uso** (ni sobrecalibrada, ni por debajo del umbral);
- es **duradera** dentro del límite de lo que su naturaleza autoriza;
- está **firmada** —su autor está nombrado, su fecha es conocida, sus criterios están enunciados—;
- es **entregable** —su parte de lector es practicable, su ventana de lector está nombrada—.

La actitud de referente

Es el eje en torno al cual gira todo, y la brújula de cada elección técnica de este manual: **ayudar, hacer autónomo, no desposeer**. Se siguen de ello tres compromisos concretos, que juzgarán todas las herramientas del libro.

- **Documentación viva**: cada herramienta va acompañada de una cartografía que el cliente comprende. Es redactor jefe de sus datos, no su inquilino.
- **Coconcepción**: el cliente identifica él mismo lo que merece la automatización y lo que debe seguir siendo manual; conserva la soberanía sobre su gesto de oficio.
- **Capacidad de leer mantenida**: la herramienta vuelve al responsable más capaz de leer sus datos, nunca lo dispensa de ello.

Criterio verificable. En todo momento, un cliente debe poder volver a hacerse con el control o cambiar de proveedor sin perder sus datos ni su comprensión. Toda herramienta de este manual que violara este criterio es una mala herramienta, por eficaz que sea.

La misión recurrente del canalero es la del herrero al que se vuelve a llamar para una pieza nueva, no la del software del que ya no se puede salir. Este principio, se verá, rige la arquitectura entera de lo que se fabrica.

Capítulo 2 — El método hexagonal: cavar lo perenne

De todas las herramientas del manual, esta no es un software: es un **método de trazado**. Responde a la restricción más dura del canalero digital —*las herramientas cambian, el oficio permanece*— y responde a ella mediante una disciplina de organización del código. La arquitectura hexagonal (llamada también *ports & adapters*) es la traducción técnica directa de la actitud de referente: garantiza que un cliente pueda cambiar de tecnología sin perder su lógica de negocio, y que la obra no caduque con la versión de una biblioteca.

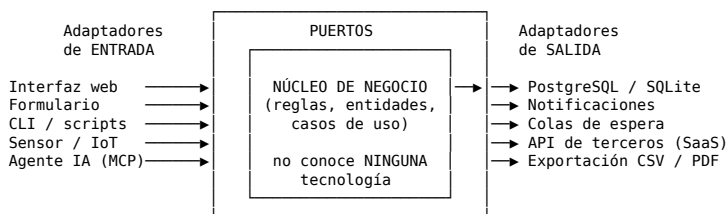
El principio: aislar el núcleo

Una aplicación de gestión mezcla dos cosas de naturalezas muy diferentes:

- **El núcleo de negocio** —los conocimientos y las reglas propios de la actividad del cliente—. *Una carga es óptima por encima del 85 % de utilización. Un presupuesto no convertido en pedido en treinta días se convierte en un recordatorio. El IVA cambia al superarse tal umbral.* Este núcleo no depende de ninguna tecnología. Es el gesto de oficio puesto en reglas.
- **La infraestructura** —todo aquello por lo que el núcleo toca el mundo exterior: la base de datos, la interfaz de usuario, las API de terceros, los sensores, los archivos, los correos electrónicos—. Esta capa está **sujeta a las restricciones externas de evolución**: envejece, se sustituye, se reconfigura.

El error corriente es tejer las dos juntas —escribir la regla de negocio *dentro* del código que habla con la base de datos o que compone la pantalla—. El día en que la base cambia, la regla se pierde con ella.

La arquitectura hexagonal **prohíbe esta mezcla**. Sitúa el núcleo de negocio en el centro, totalmente ignorante del mundo exterior, y hace comunicar ese centro con la infraestructura mediante **contratos** llamados *puertos*, de los que cada tecnología concreta no es más que una implementación *adaptador* intercambiable.



A la izquierda, los **adaptadores de entrada**: todo lo que solicita al núcleo (un clic en el cuadro de mando, un formulario móvil relleno en el terreno, un script programado, un sensor, un agente de IA). A la derecha, los **adaptadores de salida**: todo lo que el núcleo acciona para persistir o emitir (registrar en base, enviar un recordatorio, publicar en una cola, llamar a una API). El núcleo, en el centro, no sabe si la base es PostgreSQL o un archivo, ni si la entrada viene de un humano o de una máquina. Solo conoce sus *puertos*.

Por qué es la herramienta de la perennidad

Tres beneficios, que son otras tantas promesas cumplidas al cliente.

Se cambia la forma sin tocar el fondo. El cliente empieza con un almacenamiento local en archivo; seis meses más tarde su volumen crece, se pasa a una verdadera base de datos. Del lado del núcleo de negocio: *nada cambia*. Solo se ha sustituido un adaptador de salida por otro, detrás del mismo puerto. La regla de gestión que hacía el valor de la obra está intacta. Es exactamente la distinción fondo/forma de la trilogía, grabada en el código.

El oficio es legible y transmisible. La lógica de negocio, aislada, puede escribirse y releerse en lenguaje claro antes de todo código (véase más adelante). El cliente puede validarla —«sí, es así como calculo el rendimiento de un vehículo»— sin saber nada de programación. Es la coconcepción hecha posible por la arquitectura.

Se prueba el núcleo sin encender nada. Como el núcleo no depende de ninguna base ni de ninguna red, se lo puede verificar aisladamente, con adaptadores falsos. El canal se controla antes de conectarlo al río.

La infraestructura obedece a los puertos

El orden habitual del razonamiento es: *tengo* una base de datos, un servidor de correo, un modelo de IA —y escribo código para servirme de ellos—. La infraestructura viene primero; el código se pliega a ella. El hexágono **invierte** ese orden. El puerto —el contrato— viene primero; la infraestructura obedece. No se pregunta «¿qué sabe hacer mi infraestructura?» sino «¿qué necesita el núcleo?», y el puerto lo enuncia.

Toda tecnología concreta ya no es entonces más que un adaptador que cumple ese contrato, intercambiable —incluso en el tiempo—.

La consecuencia va más allá de un simple intercambio de adaptadores. **Un mismo puerto puede llevar varios adaptadores, y es lo que está disponible lo que decide cuál se monta.** Un puerto tiene su adaptador *degradado*, cuando no hay ninguna infraestructura real, y su adaptador *amplificado*, cuando la hay. Forma fechada, para concretar las ideas: un puerto «envío» —sin servidor de envío, un adaptador deposita la solicitud en una **cola de espera** que el artesano retoma a mano; con servidor, un adaptador envía de verdad—. Un puerto «consulta de un modelo» —sin motor, el adaptador responde *«puesto requerido»*; con motor, responde—. (En la obra de esta edición, se llama «el puesto» a esta infraestructura que cumple los puertos; es una forma, fechada y sustituible.) El núcleo, y la regla de negocio que llama al puerto, **ignoran cuál de los dos adaptadores está conectado.**

El punto decisivo está ahí: **la ausencia de infraestructura no es una avería, es un caso que el puerto prevé.** Cuando falta la infraestructura, el puerto no se rompe —se degrada *fielmente*—. Una solicitud que no puede atenderse ahora se pone en cola (un acto diferido, pero consignado) o devuelve *«falta esto»* (una consulta que dice lo que espera). El puerto cumple su contrato incluso con las manos vacías: no miente, no falla en silencio. Que la infraestructura obedezca al puerto quiere decir que incluso su ausencia es un estado previsto, nombrado, atravesable.

De ahí una sustitución **local y total**. El día en que llega una infraestructura real, se sustituye el adaptador —puerto por puerto— sin tocar el núcleo, ni la regla de negocio, ni la interfaz. La función que llama a *«enviar»* no cambia según que el envío espere en cola o salga de verdad. Es el *«se cambia la forma sin tocar el fondo»* ya visto, extendido: no solo se sustituye una infraestructura por otra, sino que se la ve **aparecer y desaparecer** sin que el núcleo lo advierta.

Esta inversión tiene un alcance que excede lo técnico: dice *quién manda*. El artesano, guardián del canal, establece mediante los puertos lo que el núcleo necesita; la infraestructura —una base, un servidor, un modelo, una cola— es forma fechada: alquilada, sustituible, a veces ausente, nunca dueña. Se posee el contrato; no se posee el mundo exterior, uno se conecta a él —y si no está, se lo espera en una cola, sin romper nada—.

Los puertos dictan la infraestructura —nunca a la inversa—. Es la máxima que hay que grabar sobre el hexágono: el puerto es el fondo (la necesidad duradera del oficio), el adaptador es la forma (la tecnología del día), y la perennidad de la obra depende de que la segunda nunca mande sobre el primero.

El gesto del canalero: modelizar antes de codificar

El procedimiento cabe en tres gestos, en este orden, siempre.

1. **Hablar el oficio del cliente.** Distinguir el **corazón del oficio** —allí donde la informática no dicta nada, allí donde reside el saber hacer del cliente— de la **matriz de apoyo**: las tareas repetitivas, administrativas, automatizables, que no lo movilizan (recordatorios, seguimiento, contabilidad, entrada de datos). Nunca se toca el corazón del oficio; se modeliza la matriz para liberar el corazón del oficio. A la matriz de apoyo, común y genérica, responde la **matriz de oficio**: el encadenamiento condicionado del trabajo propio —los estados de la pieza y sus condiciones, nunca el gesto mismo—. Su principio se expone en el volumen gemelo, el *Sistema de gestión* (sección III); *Datos técnicos* dice con qué se lo dota de herramientas.
2. **Escribir la lógica de negocio en claro.** Antes de abrir el editor de código, se redactan las **entidades** (los objetos del dominio y sus propiedades), las **reglas** (las restricciones) y los **casos de uso** (los encadenamientos). En lengua natural, en pseudocódigo, en esquema. Este documento se valida con el cliente y sobrevivirá a todos los lenguajes.
3. **Implementar en el lenguaje elegido.** El desarrollador transcribe la lógica validada. Si el lenguaje cambia algún día, se vuelve a transcribir; el documento de lógica, en cambio, no se mueve.

Un ejemplo: la lógica de optimización de una carga

He aquí a qué se parece la capa núcleo, escrita en claro, independiente de toda tecnología. Podría implementarse en cualquier lenguaje sin perder nada.

Entidades del dominio

- *Artículo*: identificador, nombre, peso unitario (kg), volumen unitario (m³), cantidad, frágil (sí/no), prioridad (1 alta → 3 baja), apilable (sí/no).
- *Vehículo*: identificador, capacidad de peso (kg), capacidad de volumen (m³), costo por kilómetro, disponibilidad.

Reglas de negocio fundamentales

- *RG1 — restricciones físicas.* El peso total y el volumen total cargados nunca superan las capacidades del vehículo; los dos se aplican simultáneamente.
- *RG2 — fragilidad.* Un artículo frágil aumenta en un 10 % el volumen que ocupa (seguridad).
- *RG3 — prioridades.* Se trata primero la prioridad alta; en caso de igualdad, se privilegia la densidad alta (peso/volumen); como último criterio, la cantidad grande.
- *RG4 — umbral de eficacia.* Una carga es óptima si su utilización alcanza al menos el 85 %, siendo la utilización el mínimo entre la tasa de llenado en peso y la de volumen. Por debajo, el sistema alerta.

Caso de uso — cálculo de eficacia (pseudocódigo)

```

PARA un conjunto de artículos y un vehículo:
  peso_total      = SUMA(artículo.peso_unitario × artículo.cantidad)
  volumen_nominal = SUMA(artículo.volumen_unitario × artículo.cantidad)
  volumen_ajustado = volumen_nominal
  PARA cada artículo frágil:
    volumen_ajustado += artículo.volumen_total × 0,10
  eficacia_peso   = peso_total / vehículo.capacidad_peso
  eficacia_volumen = volumen_ajustado / vehículo.capacidad_volumen
  eficacia = MIN(eficacia_peso, eficacia_volumen)
  limitación = SI eficacia_peso < eficacia_volumen ENTONCES "peso" SI NO "volumen"
DEVOLVER { eficacia, limitación }

```

Este texto es la obra, en el sentido del fondo. El código JavaScript u otro que lo implemente no es más que una encarnación fechada de él. Es esta inversión —la regla primero, la tecnología después— la que hace la arquitectura hexagonal, y la que hace lo perenne.

La organización concreta de un proyecto hexagonal

En el disco, la separación se lee en el árbol de carpetas. El núcleo (domain) nunca menciona una tecnología; los ports declaran los contratos; los adapters los implementan; infra aloja los servicios externalizables.

```

proyecto-cliente/
├── domain/           # NÚCLEO: ninguna dependencia tecnológica
│   ├── entities/    # objetos del oficio (Artículo, Vehículo, Presupuesto...)
│   ├── rules/       # reglas de gestión (RGL...RGn)
│   └── use-cases/    # encadenamientos (optimizarCarga, recordarPresupuesto...)
├── ports/           # CONTRATOS
│   ├── input/       # puertos de entrada (comandos, consultas)
│   └── output/      # puertos de salida (depósito, notificación, publicación)
├── adapters/        # IMPLEMENTACIONES intercambiables
│   ├── storage-sqlite/ # un adaptador del puerto "depósito"
│   ├── storage-postgres/ # otro, que se puede sustituir sin tocar el núcleo
│   ├── notify-email/
│   └── export-pdf/
├── apps/            # APLICACIONES cliente (véanse cap. 3 y 7)
│   ├── web/         # el cuadro de mando (HTML/CSS/JS)
│   └── infra/        # servicios de base (base, caché, secretos, logs)

```

Salvaguarda de sobriedad. El hexágono no es una invitación a multiplicar las capas. Para un emprendedor en solitario, el núcleo puede caber en unos pocos archivos y un solo adaptador de almacenamiento. El método no impone la pesadez; impone la *separación*. Se permanece en el óptimo: bastante estructura para que el cliente pueda cambiar de herramienta algún día, no más.

Lo que el hexágono no aísla por sí solo

La arquitectura protege el núcleo, pero no dispensa del juicio. Le corresponde al canalero decidir *lo que* entra en el núcleo (el saber de oficio duradero) y *lo que* permanece en la periferia (las elecciones fechadas). Mal trazada, la frontera deja escapar una regla de negocio hacia un adaptador, y la promesa de perennidad se rompe. El método es

una lámpara; el trazado justo sigue siendo un gesto de artesano, que hay que retomar en cada misión.

Capítulo 3 — El núcleo sin dependencias: HTML, CSS, JavaScript

Si la arquitectura hexagonal decide *dónde* va el código, este capítulo decide *con qué* se escribe el núcleo de la aplicación que toca al cliente: su interfaz y su lógica de presentación. La respuesta del canalero preocupado por la perennidad es nítida y, a primera vista, austera: **el trío nativo de la web —HTML, CSS, JavaScript— sin framework ni dependencias**. Se habla a menudo de *vanilla JS*. Esta elección no es un conservadurismo; es una estrategia de duración y de soberanía.

Por qué lo nativo en lugar de un framework

Los frameworks de interfaz (React, Angular, Vue y sus sucesores) prestan servicios reales en las grandes aplicaciones de equipo. Pero para la obra de un canalero entregada a un emprendedor en solitario, introducen tres fragilidades que la actitud de referente no puede aceptar.

La caducidad acelerada. Un framework impone sus versiones, sus rupturas, su cadena de herramientas de compilación. Una aplicación construida sobre él exige un mantenimiento permanente *solo para seguir en su sitio*. El lenguaje nativo del navegador, en cambio, no se rompe: una página HTML/CSS/JS bien escrita hoy se abrirá todavía dentro de diez años. Es la diferencia entre un canal que hay que volver a cavar cada temporada y un canal de mampostería.

El peso y la dependencia. Un framework añade cientos de kilobytes y un bosque de dependencias de terceros, cada una de las cuales es una puerta de entrada para un fallo de seguridad o un abandono. Lo nativo no añade nada: el navegador ya está ahí. La aplicación sigue siendo ligera, rápida, compatible con equipos modestos —lo que cuenta para un artesano con un teléfono de gama baja, en una furgoneta, sin red—.

La reversibilidad. Una obra en nativo es legible por cualquier desarrollador, sin conocer un framework particular. El cliente puede

cambiar de proveedor sin quedar prisionero de un ecosistema. Es el criterio verificable del capítulo 1, aplicado a la capa de interfaz.

Lo nativo no es la ausencia de método. Escribir sin framework no quiere decir escribir de cualquier manera. El núcleo aplicativo nativo se organiza en módulos claros (un módulo por pantalla, un módulo por servicio), y es precisamente la arquitectura hexagonal la que le da su columna vertebral. Lo nativo proporciona la materia; el hexágono proporciona el orden.

Los tres materiales, y lo que lleva cada uno

HTML — la estructura. El HTML describe *lo que es*: los campos de un formulario de entrada de datos en el terreno, las zonas de un cuadro de mando, la lista de los presupuestos. Bien escrito (etiquetas semánticas, rótulos explícitos, orden lógico), es legible por los humanos, por las máquinas y por las tecnologías de asistencia. La estructura es el esqueleto de la obra; sobrevive a todo revestimiento.

CSS — la forma y la sensación. El CSS describe *cómo aparece*: colores, espaciados, legibilidad, jerarquía visual, adaptación al tamaño de la pantalla (un mismo cuadro de mando debe leerse en una gran pantalla de taller y en un teléfono). Es también, se verá en el capítulo siguiente, el material de las animaciones que vuelven agradable la experiencia —la barra de progreso que se llena, el *confeti* discreto al alcanzar un objetivo—. El CSS lleva una gran parte de la fluidez percibida, sin una línea de JavaScript.

JavaScript — el comportamiento. El JS describe *lo que reacciona*: calcular, poner al día un indicador cuando entra un dato, cambiar un estado, desencadenar una alerta al superarse un umbral. Es el JS el que anima el adaptador de entrada (la entrada de datos) y el adaptador de lectura (la presentación), y el que llama al núcleo de negocio. Bien ordenado y modular, sigue siendo comprensible y comprobable.

Aplicación web, aplicación instalable: el mismo núcleo

El mismo núcleo HTML/CSS/JS puede presentarse bajo varias formas según la necesidad, sin reescribir el oficio:

- **SPA** (*single-page application*) —una aplicación web que se comporta como un programa, sin recarga de página—. Adecuada para el cuadro de mando de dirección consultado en la oficina.
- **PWA** (*progressive web app*) —la misma aplicación, vuelta **instalable** en el teléfono o el ordenador, capaz de **funcionar sin conexión** y de almacenar localmente—. Es a menudo la forma ideal para el emprendedor en solitario móvil: «instala» la herramienta en su

pantalla de inicio, introduce un movimiento de existencias en la furgoneta sin red, y la sincronización se hace al volver la conexión.

- **Escritorio** mediante un envoltorio (tipo Electron) o **móvil híbrido** mediante un envoltorio (tipo Cordova/Capacitor) —cuando el cliente quiere una verdadera aplicación instalada—. No son más que **adaptadores de entrada** suplementarios en torno al mismo núcleo.

Lo importante es el principio: *un solo núcleo de negocio, varios revestimientos de acceso*. Se elige el revestimiento en el óptimo de la necesidad, nunca por moda.

El almacenamiento local del núcleo

Una aplicación nativa dispone, en el navegador, de reservas locales que permiten funcionar sin servidor:

- reservas simples para pequeñas preferencias;
- una base local estructurada (de tipo *IndexedDB*) para conservar volúmenes de datos de negocio en el equipo, sin conexión, y sincronizarlos después.

Para un gran número de emprendedores en solitario, **lo local basta**: sus datos caben en su equipo, cifrados, guardados con copias de seguridad, y nunca salen de la empresa. El servidor remoto (capítulo 6) solo se moviliza cuando la puesta en común o la mutualización lo exigen. Es la sobriedad de la arquitectura *local-first*: tratar lo más cerca posible de la fuente, no hacer viajar más que lo que debe viajar.

Varios bancos de trabajo, una sola obra

De la convergencia de las inscripciones separadas. Un artesano puede llevar su obra en varios bancos de trabajo. Lo que los bancos de trabajo inscriben por separado no se contradice: se sucede —cada inscripción permanece como un acto congelado, una constatación fechada, llevada por el banco de trabajo que la produjo—. Su reunión nunca es, pues, una copia de estado, en la que uno borraría al otro: es una reproducción de actos, en un orden convenido, en la que cada acto permanece. La lectura corriente sigue el orden de la reproducción; el acto superado no se destruye —permanece en el diario, con su fecha y su banco de trabajo—. Y cuando dos bancos de trabajo han inscrito lo mismo durante su separación, la divergencia se le dice al artesano: la obra constata, no zanja en silencio.

Conectar el mundo exterior — siempre por el hexágono

El núcleo nativo nunca habla *directamente* con un servicio externo. Todo lo que viene de fuera —un sensor, una API pública, un servicio SaaS, un

agente de IA— entra por un **adaptador** conectado a un **puerto**. Es la regla que vuelve perenne el conjunto y el capítulo 2 planteó su principio. Algunos ejemplos de conexiones corrientes, cada una aislada detrás de su puerto:

- **API públicas de referencia** (por ejemplo, las API del Estado para verificar un número de empresa o normalizar una dirección): un adaptador de entrada enriquece el dato introducido.
- **Cobro** (un proveedor de pagos): un adaptador de salida, sustituible si el cliente cambia de proveedor.
- **Logística, cartografía, mensajería**: otros tantos adaptadores, nunca dependencias tejidas en el núcleo.
- **Sensores y objetos conectados**: un adaptador de entrada traduce la señal del sensor en dato del dominio (véase el capítulo 5).

El día en que uno de esos servicios cierra, sube sus precios o caduca, se sustituye el adaptador afectado. El núcleo —el gesto de oficio del cliente— no se mueve. Es la promesa de soberanía cumplida hasta en la menor conexión.

Una nota de honestidad

Lo totalmente nativo pide rigor: sin framework que imponga una estructura, le corresponde al artesano sostenerla. Es precisamente por eso por lo que este manual insiste tanto en la arquitectura hexagonal, en la modularidad y en la documentación viva. Lo nativo no es la elección de la facilidad; es la elección de la duración y del dominio. Para una obra firmada, que un cliente debe poder conservar y comprender durante años, es el arbitraje correcto. Como todo arbitraje, vuelve a plantearse: un proyecto de una envergadura muy distinta podría justificar otras elecciones —pero eso ya no sería del todo el taller, sería la cadena—.

Capítulo 4 — La gamificación: volver la obra agradable y atractiva

Una obra solo sirve recibida: si el emprendedor en solitario no lee su cuadro de mando, no introduce sus movimientos, abandona la herramienta al cabo de tres semanas, el canal mejor cavado queda sin uso, y su valor no se realiza. La recepción no es un suplemento; es la condición de la utilidad. La **gamificación** —la ludificación— es el método que trabaja esa recepción: vuelve el uso agradable, motivador, casi sin esfuerzo, de modo que el gesto de gestión, por lo común tedioso, se vuelva fluido e incluso satisfactorio.

La constatación que la justifica está documentada: una parte importante de los emprendedores en solitario abandonan su gestión por **falta de visibilidad sobre su rendimiento** y por **desinterés por tareas percibidas como tediosas**. La gamificación ataca exactamente esas dos causas: vuelve el rendimiento *visible* y el esfuerzo *recompensado*.

El principio: transformar el dato consolidado en señal motivadora

El tomo 1 nombra el dato consolidado: la síntesis de alto valor de uso extraída de un canal. La gamificación es el arte de **presentar ese dato consolidado en una forma que hable al lector situado** —que lo informe de un vistazo, lo sitúe con respecto a un objetivo, y lo anime—. No fabrica ningún dato nuevo; viste el dato verdadero para que se lea.

El modelo de referencia es el de las aplicaciones que volvieron lúdico el aprendizaje o el seguimiento de hábitos: barras de progreso, niveles, rachas, pequeñas recompensas. Trasladado a la gestión de actividad, da mecánicas sencillas y honestas.

Las mecánicas, y el gesto de oficio al que sirven

Mecánica	Ejemplo concreto	A qué sirve
Barra de progreso	«Ha alcanzado el 80 % de su objetivo mensual de cifra de negocio.»	Visualización instantánea del estado (dato consolidado legible de un vistazo)
Niveles e insignias	«Insignia <i>Facturación al día</i> desbloqueada tras 10 facturas emitidas dentro de los plazos.»	Motivación para completar las tareas repetitivas de la matriz de apoyo
Notificación dirigida	«El pedido n.º 123 lleva retraso de entrega: hay que tratarlo.»	Reducción de los olvidos; alerta en el momento adecuado
Animación de umbral	Un efecto visual discreto (<i>confeti</i>) al alcanzar un objetivo.	Refuerza el sentimiento de logro
Racha / regularidad	«7 días de entrada de existencias sin interrupción.»	Instala el hábito de tomar (mantenimiento del canal por el propio cliente)

Todas estas mecánicas se realizan en **HTML/CSS/JS nativo**, sin ninguna dependencia: una barra de progreso es un `<div>` cuyo ancho sigue un indicador; un *confeti* es una animación CSS; una notificación es un resaltado en pantalla o una notificación del sistema. La gamificación no hace más pesada la obra; la vuelve viva con los materiales ya presentes.

Los umbrales: el puente entre gamificación y dirección

La mecánica más útil para el emprendedor en solitario no es la insignia; es el **umbral**. Un umbral es un límite cifrado que, superado, desencadena una señal. El glosario del oficio distingue tres tipos, y todos se gamifican de forma natural.

- **Umbral crítico bajo** —por debajo, la actividad ya no es perenne («*menos de 5 pedidos este mes*»)—. Señal de alerta.
- **Umbral crítico alto** —por encima, se impone una reestructuración («*más de 20 pedidos: plantearse delegar o contratar*»)—. Señal de anticipación.
- **Umbral intermedio** —un objetivo que la empresa se fija para un beneficio directo («*15 pedidos: una semana más de vacaciones*»)—. Es el umbral-recompensa, el corazón de la gamificación motivadora.

El caso ejemplar es la **superación de un umbral fiscal** (IVA, límite de cifra de negocio). Para el emprendedor en solitario, esos umbrales son fuente de angustia porque se padecen: se descubren después de haberlos superado. Una barra de progreso hacia el límite, una alerta al 70 % del umbral, transforman la angustia en dirección: *el umbral ya no se padece, se anticipa*. Es, literalmente, devolver al responsable el asidero sobre su trayectoria —el objeto mismo de la obra—.

La salvaguarda: gamificar sin falsear

La gamificación es potente, y por tanto peligrosa. El tomo 1 planteó que la observación inserta en un bucle decisional **transforma lo que observa**: una puntuación que cambia el comportamiento que mide. Mal empleada, la gamificación puede empujar a malas decisiones —cerrar una venta mediocre para «desbloquear una insignia», inflar una cifra para hacer avanzar una barra—. Tres reglas contienen ese riesgo.

1. **Recompensar el gesto justo, nunca la cifra bruta.** Se gamifica *«facturar dentro de los plazos», «reclamar un impago», «tener las existencias al día»* —gestos virtuosos— y no *«hacer cifra a toda costa»*. La mecánica debe servir a la salud de la actividad, no halagarla.
2. **No disfrazar nunca un dato proyectado de dato medido.** Una barra que anticipa una tendencia debe señalarse como proyección. Un dato medido y un dato estimado no se muestran con la misma mirada. La probidad de lo que se muestra prevalece sobre el efecto.
3. **Mantener la capacidad de leer, no adormecerla.** La gamificación simplifica la lectura; no debe dispensar de comprender. Detrás de cada indicador ludificado, el cliente debe poder acceder al dato bruto y a su modo de cálculo. La animación es una puerta de entrada hacia el dato, nunca un telón ante él.

El buen espíritu. La gamificación no es un entretenimiento superpuesto a la gestión. Es una manera de volver **practicable la parte del lector** —exactamente la definición que el tomo 2 da del transporte cumplido—. Un emprendedor en solitario que no tenía ni el tiempo ni el gusto de leer sus cifras las lee, porque se las han vuelto legibles y atractivas. Es alfabetización de datos hecha accesible, no barniz.

La experiencia fluida, más allá de las mecánicas

La gamificación en sentido estricto (insignias, barras) se inscribe en una exigencia más amplia: **la experiencia de usuario**. Para el emprendedor en solitario, fluido quiere decir concreto:

- **Poca entrada de datos, bien situada.** Sustituir un cuaderno olvidado por una interfaz móvil en la que cada movimiento pone al día un indicador; el artesano ya no «rellena una tabla», dispone de un inventario al día. La entrada de datos se funde en el gesto de oficio en lugar de añadirse a él.
- **Autocompletado y prerrelleno.** Verificar un cliente, normalizar una dirección, retomar la información ya conocida: cuanto menos teclea el emprendedor en solitario, más permanece en su oficio.
- **Legibilidad inmediata.** Un cuadro de mando que responde de un vistazo a las tres preguntas que cuentan (*¿dónde estoy? ¿qué debo hacer? ¿qué se está desviando?*), sin anegarlo bajo el resto.

- **Serenidad por la alerta justa.** Ser avisado de lo que se desvía, y *solo* de lo que se desvía. Una notificación que grita todo el tiempo ya no dice nada; la sobriedad de la alerta hace su valor.

La experiencia fluida, agradable, dominada no es una comodidad accesoria: es el criterio de éxito de la obra. Un canal que no da ganas de leer es un canal que no se actualizará. La gamificación, sostenida con probidad, es la herramienta que conserva al cliente *en situación de observar su propia actividad* —y le devuelve la serenidad—.

Capítulo 5 — Fabricar el dato: toma y anotación

Primer tiempo del dato, primer gesto: **la inscripción**. Es aquí donde la actividad del cliente —el río que fluye— se convierte en dato. Dos familias de herramientas sirven a este tiempo: las **herramientas de toma**, que materializan el acto de inscripción, y las **herramientas de anotación**, que transforman una señal en dato calificado. Este capítulo las declina en su forma digital de 2026.

Cartografiar la materia: qué datos fabricar

Antes que toda herramienta, una criba. No todos los datos de una actividad valen lo mismo, y el glosario del oficio los clasifica en tres tipos —distinción que rige la seguridad, el almacenamiento y la puesta en común—.

- **Dato de negocio** —directamente ligado a la realización de la actividad: tiempo dedicado a un trabajo, materiales consumidos, pedidos en curso, kilómetros recorridos—. Es el corazón de la toma.
- **Dato anexo** —recogido indirectamente, sin impacto directo en la actividad pero potencialmente valorizable más adelante: posiciones GPS durante los desplazamientos, horas de uso de una máquina (para un futuro mantenimiento predictivo)—. Por tomar con discernimiento: útil, pero nunca al precio de la sobreingeniería.
- **Dato sensible** —sometido a reglamentación (RGPD) o crítico: datos de contacto de clientes, datos financieros—. Impone cuidados particulares de almacenamiento y de acceso (capítulo 9).

A esta primera criba se superpone un segundo eje, que ya no clasifica el dato por su naturaleza sino por su **dependencia**. Algunos datos se sostienen solos; otros solo existen por otro —un recordatorio por su presupuesto, un documento adjunto por la ficha que documenta—. El primero se llama **sustentante**, los segundos **sustentados** (véase, en las notas complementarias, *El dato sustentante y el dato sustentado*). Localizar esta relación en la cartografía evita una falta de modelo: no se da a un sustentado una existencia autónoma que no tiene.

La realización más simple es el **anidamiento**: el sustentado se aloja en el registro de su sustentante (un campo que contiene sus recordatorios, sus documentos adjuntos). Hereda de él mecánicamente el destino —se lee, se guarda en copia de seguridad, y **desciende en racimo** con su sustentante cuando este entra en caducidad (véase «Regular el plazo límite de uso», más adelante en este capítulo)—. Se podría sostener la misma relación mediante una tabla vinculada con supresión en cascada: el medio es sustituible, el hecho de estructura no lo es.

La toma se modeliza sobre el **proceso de negocio** del cliente, descompuesto en términos estables: *etapa* (fase amplia: presupuesto, fabricación, entrega), *tarea* (acción atómica: cortar, enviar una factura), *secuencia operativa* (grupo ordenado de tareas que produce un resultado identificable: «preparar un trabajo»). Se toma allí donde la dirección necesita ver, no en todas partes.

Regular el plazo límite de uso

El tomo 3 plantea el principio: la caducidad de estado no es un instante sino un umbral —el **plazo límite de uso**, el punto en el que la brecha con lo real excede lo que el uso tolera—. El fondo dice que tal umbral existe; la forma dice aquí cómo se regula. Tres casos cubren el taller. **El umbral intrínseco**: el objeto del dato fija él mismo el término (un presupuesto rechazado ve su uso cerrado de entrada —plazo nulo, paso inmediato a la caducidad—; una oferta fechada expira en su fecha). Se inscribe como campo del dato, fijado en el acto de inscripción. **El umbral de uso**: nada en el objeto fija el término, pero el uso lo conoce (un registro de existencias solo sirve al acto durante un día; un precio de suministro, una temporada). Se regula por tipo de dato, en el armazón —es un reglaje del canal, revisado en la revisión del canal—. **El valor por defecto de edad**: cuando no se especifica nada, se aplica un valor por defecto prudente, según la edad del dato y la cadencia de su canal. Al término, la obra ejecuta el paso de estado en el orden que el fondo impone: la parte sumable se consolida, el nuevo estado se escribe, y solo entonces el dato sale del uso corriente —la caducidad ordena, no destruye—. Y el racimo sigue a su sustentante: cuando este entra en caducidad, sus sustentados descienden con él.

El armazón y la toma: el cauce y el agua

A la criba por la naturaleza (de negocio, anexo, sensible) y al eje de la dependencia (sustentante, sustentado) se añade un tercer punto de referencia, que rige la modelización. No todo dato está al mismo nivel de la construcción: algunos **estructuran** —los nombres, los tipos, los formatos en los que los demás vienen a ordenarse («empresa», «dato de contacto», «tipo de dato de contacto») —; otros son lo que **se ordena** —el valor de un SIRET, un número marcado, un importe—. Los primeros

forman el **armazón**; los segundos son las **tomas** operativas (véase, en las notas complementarias, *El armazón: el canal cavado en las fuentes estables*). El armazón es el cauce; la toma es el agua. Ambos son datos —todo dato nace de una toma (vía observacional), de un acto que instituye o de una operación que deriva—, pero el armazón se toma en **fuentes estables**, lo convenido, mientras que el agua viene de lo real volátil de la actividad.

En la forma, esta diferencia no se marca mediante una convención de escritura, sino mediante un **campo**. Cada nodo del modelo lleva un rol: armazón o toma. El canalero lee así, de un vistazo, dónde está el cauce y dónde está el agua —y la herramienta puede tratar de manera diferente lo que se cava una vez (el armazón) y lo que fluye sin cesar (las tomas)—.

La lista de adyacencia: sostener el árbol

El armazón no es plano: un dato de contacto lleva teléfonos, un teléfono una designación y un número. Es un **árbol**, y la forma más simple de sostenerlo es la **lista de adyacencia**: cada nodo conoce a su padre, y el árbol se reconstituye siguiendo los vínculos. Un nodo, esquemáticamente:

- id — un identificador único con marca de tiempo (UUID v7, capítulo 6); la identidad del nodo;
- rol — armazón o toma (el cauce o el agua);
- nombre — el nombre que lleva el nodo;
- padre — el id del nodo del que depende, o nada si está en la raíz;
- valor — lo que se capta, para una toma (vacío para un nodo de armazón);
- references — una lista de remisiones (véase más abajo);
- y los **metadatos** que este capítulo ya ha nombrado: fecha, autor, unidad, contexto.

Esta lista plana es **navegable** (se desciende, se remonta, se busca un nodo por su id) y **extensible**: cavar un canal más es añadir nodos, sin tocar los demás.

Las dos aristas: la sustentación y la referencia

Dos vínculos unen los nodos, y no se realizan de la misma manera —es la contrapartida de forma de la nota *El dato sustentante y el dato sustentado* (notas complementarias), que distingue el **sustentado** de la **referencia**—.

- **El vínculo padre realiza la sustentación** —la composición, el destino compartido—. Un nodo depende de su padre; nace, vive y **desciende en racimo** con él. Su supresión sigue a la del padre: **cascada**. Es la arista del árbol. Un dato de contacto sin su empresa, una designación sin su teléfono ya no tienen objeto: se retiran con él.

- **El campo references realiza la referencia** —la remisión hacia un dato con **vida propia**—. Se almacena en él el id (UUID) del destino, acompañado de un **label** que dice la naturaleza del vínculo (`{ label: "responsable del cargo", destino: <UUID de una ficha de RR. HH.> }`). **Sin cascada**: suprimir el destino solo rompe la remisión; suprimir el nodo que remite no toca el destino, que subsiste en su propio árbol. La referencia vincula sin anudar la existencia.

Sostener estas dos aristas distintas no es un refinamiento: es lo que rige, en la supresión, **lo que muere en racimo y lo que sobrevive solo**. Confundirlas —poner un vínculo de referencia en cascada, o a la inversa— corrompe la semántica de las dos muertes.

Anidar o enumerar: el criterio es la búsqueda

Este capítulo ya ha dado, para el sustentado, una realización más simple aún que la lista: el **anidamiento** —alojar el sustentado en el registro de su sustentante (un campo que contiene sus recordatorios, sus documentos adjuntos)—. El medio sigue siendo sustituible, pero la elección no es indiferente, y un criterio la zanja: **la búsqueda**.

Mientras un sustentado **solo se alcance a través de su sustentante** —las líneas de un presupuesto, un emisor congelado en ese presupuesto—, el anidamiento conviene: nunca se lo busca solo, y el racimo se lee en bloque. Pero en cuanto un dato debe **buscarse o recorrerse por sí mismo** —encontrar todos los datos de contacto de un tipo en toda la base, enumerar todas las citas de una semana—, el anidamiento lo vuelve costoso de alcanzar: habría que abrir cada sustentante para rebuscar en su contenido. La **lista de adyacencia** lo sostiene entonces mejor: cada nodo es una entrada aparte, indexable, consultable directamente, sin atravesar a sus vecinos.

La regla es, pues: **anidamiento para el sustentado que solo se lee en racimo; lista de adyacencia en cuanto el dato debe buscarse solo**. Elegir uno por otro no cambia el hecho de estructura —el sustentado sigue siendo el sustentado—, pero puede lastrar el funcionamiento.

El armazón se mantiene

Los nodos rol: armazón —los tipos, los formatos— no se inventan: se toman en lo convenido (las convenciones sociotécnicas que establecen cómo se designa una dirección, un número, una forma jurídica). Son datos de **fuerza estable mantenida** (véase la nota *La envoltura: la calidad como mantenimiento de lo convenido*, en las notas complementarias): duraderos, pero no eternos. En la forma, esto tiene una consecuencia: el armazón se **versiona** y se **revisa**, deliberadamente, cuando la convención se desplaza —el día en que una dirección se convierte en un triplete de coordenadas, en que el número cede ante un identificador—. Tener el

armazón afinado con lo convenido es un gesto de mantenimiento, de la misma naturaleza que la revisión del canal (capítulo 9), pero que versa sobre el cauce más que sobre el agua.

Salvaguarda sobre la forma. Lista de adyacencia, UUID v7, anidamiento, campo references: son medios fechados de 2026. El fondo —el armazón como canal, el sustentado y la referencia como dos vínculos de destinos diferentes, lo convenido como fuente estable mantenida— está planteado en las tres notas complementarias; estas páginas no hacen más que realizarlo. Otro soporte, otra base, realizarían los mismos hechos de otro modo.

En el óptimo, siempre. El modelo **admite** la profundidad; no la **prescribe**. Para una actividad pequeña, se cava un árbol **poco profundo pero completo en canales**: se ejercen las **dos aristas** —un sustentado en racimo, una referencia hacia otra función— sobre un armazón representativo, de modo que el potencial se vea, sin anegar el taller bajo tomas que no le sirven. Cavarlo todo sería construir en mampostería un canal donde bastaba una zanja; son la subtoma y la sobreingeniería lo que el óptimo observacional tiene a raya (capítulo 1).

Las herramientas de toma

Son los instrumentos que detienen un valor en el flujo. En versión digital, se ordenan en cuatro grupos.

1. Los formularios de recogida

La herramienta más corriente y la más subestimada. Un buen formulario de entrada de datos —una pantalla móvil rellena en el terreno, en la entrega, al final de la jornada— es un instrumento de toma, con su cadencia y su fidelidad. Los principios:

- **Fundir la entrada de datos en el gesto de oficio.** El artesano no debe «rellenar una tabla» además de su trabajo; registra un movimiento en el momento en que lo hace. Menos campos, mejor situados, en el instante adecuado.
- **Captar el corpus con el dato.** Cada toma lleva consigo su fecha, su autor, su unidad, su contexto —sin lo cual será ininterpretable más adelante—.
- **Prerrellenar y restringir.** Listas de opciones en lugar de texto libre, valores por defecto, controles de coherencia en la entrada: se impide el error aguas arriba en lugar de corregirlo aguas abajo.

2. Los scripts de extracción

Cuando el dato ya existe en otro sistema (un programa de contabilidad, una hoja de cálculo, un buzón de correo, una base existente), no se vuelve a introducir: se **extrae** mediante un script. Herramientas típicas de 2026: scripts en JavaScript/Node.js o en shell, lectura de archivos (CSV, JSON), consultas sobre una base de datos existente. El script de extracción es un adaptador de entrada: traduce el dato de un formato ajeno hacia el dominio del cliente.

Caso real. El director de un centro dispone de un gran programa de gestión pero no obtiene de él los indicadores que necesita. El canalero no instala nada nuevo: escribe **consultas sobre la base existente** que extraen exactamente la vista deseada, que el programa se limita luego a mostrar. Estos indicadores son datos derivados, calculados a partir de los del programa: fabricar un dato, a veces, es saber interrogar lo que ya está ahí.

3. Los sensores y objetos conectados (IoT)

Para las magnitudes físicas de la actividad —temperatura de un local, presencia, nivel de existencias mediante etiquetas (RFID), posición de un vehículo— unos **sensores** toman de forma continua, con alta cadencia. Pertenecen típicamente al *dato anexo* de alto potencial. El canalero conoce su **calibración** (la medida se desvía con el tiempo) y sus **sesgos** (lo que el sensor no ve). La señal bruta del sensor entra por un adaptador de entrada que la traduce en dato del dominio; nunca se infiltra en el núcleo de negocio.

4. La IA generativa como instrumento de toma semántica

He aquí el instrumento más nuevo y más potente. Un gran modelo de lenguaje (LLM) actúa como un **operador semántico**: sabe leer información *no estructurada* —un correo de cliente, una foto de una hoja de pedido, un mensaje de voz transcrito, un documento— y **extraer, calificar, estructurar** el dato. Allí donde antes hacía falta un humano para leer y copiar, o diccionarios y reglas deterministas pesados de sostener, el LLM extrae directamente los elementos deseados.

Usos típicos para el emprendedor en solitario: extracción de la información de una factura recibida (importe, fecha, proveedor); clasificación automática de los correos entrantes; calificación de una opinión de cliente; transformación de una nota manuscrita fotografiada en datos introducidos. La salida del modelo es **estructurada** (en JSON, en tabla) para entrar en el sistema. Es un dato derivado: su procedencia nombra el modelo y el documento de origen.

Este instrumento se maneja según reglas estrictas, detalladas en el capítulo 8: modelo **local** preferentemente (soberanía), **validación**

humana en todo lo que es crítico, y **combinación** con reglas deterministas allí donde la fiabilidad prevalece. El LLM no sustituye el juicio; prepara la materia que el juicio validará.

Las herramientas de anotación

Tomar no basta: hay que **calificar**. Una rejilla de anotación transforma una señal en dato imponiéndole categorías —que el flujo de lo real, por sí mismo, no contiene—. Es un acto decisivo y nunca neutral.

- **Rejillas y clasificaciones.** El canalero construye (o toma prestadas) las categorías: tipos de clientes, estados de pedido (preparación, entrega, pagado), naturalezas de gasto, niveles de prioridad. Sabe lo que su rejilla **incluye y excluye**. Recurre a una rejilla existente cuando está probada y es comparable; forja una nueva cuando el uso del cliente exige distinciones que no existen en ninguna otra parte.
- **Protocolos de anotación.** Unas reglas estables dicen *cómo* calificar, para que un mismo hecho reciba siempre la misma etiqueta —condición para que la serie sea comparable en el tiempo—.
- **Anotación asistida por IA.** El LLM puede proponer una clasificación (análisis de sentimiento de una opinión, categoría de un gasto); un humano valida. Se combinan la rapidez semántica de la máquina y la responsabilidad del gesto.

El taller **conserva sus rejillas** en su biblioteca, con su historia y las obras que han permitido. Una rejilla documentada es un saber hacer transmisible; una rejilla perdida es un canal que ya no se sabrá releer.

Regular la toma en el óptimo

Todo el arte está en el reglaje, retomado en cada misión. Para cada canal que se cava, el canalero se plantea la rejilla de los criterios del capítulo 1:

- **Qué:** qué hecho de negocio, qué dato anexo útil, qué dato sensible que proteger.
- **Cadencia:** ajustada a la velocidad de cambio de la actividad. Unas existencias de movimiento diario se toman en cada movimiento; una satisfacción de cliente se mide a un ritmo mucho más lento.
- **Fidelidad:** precisión del instrumento, calibración, margen de error aceptable para el uso.
- **Corpus:** el contexto que se lleva consigo (fecha, unidad, autor, condiciones).
- **Perímetro:** lo que se observa y lo que se deja deliberadamente fuera de campo.

Y la pregunta que zanja: *¿puede la actividad cambiar significativamente entre dos tomas?* Si es así, se densifica; si no, se aligera. Ni ciego, ni anegado. Es este reglaje el que distingue la obra del canalero de la

recolección indiscriminada —y el que ahorra al cliente el costo y el ruido de un dato inútil—.

Documentar los propios instrumentos

Un taller serio lleva un **inventario documentado** de sus herramientas de toma: para cada sensor, formulario, script o prompt de extracción, sus características, su calibración, sus sesgos conocidos, sus límites. Este inventario es la memoria de la fabricación. Permite, más adelante, juzgar lo que vale un dato —porque juzgar un dato es, ante todo, saber con qué instrumento se tomó—.

Capítulo 6 — Hacer circular: fijación y transporte

Un dato inscrito y nunca transmitido permanece en el único lugar de su inscripción; hacerlo circular extiende su disponibilidad y sus usos posibles. El segundo tiempo del dato es el de la circulación, y descansa en dos familias de herramientas: las **herramientas de fijación**, que conservan el dato en un soporte, y las **herramientas de transporte**, que lo ponen en movimiento. Para el emprendedor en solitario, la circulación tiene dos destinos bien distintos: **local**, para dirigir su propia actividad, y **remoto**, para compartir. Este capítulo trata los dos.

Las herramientas de fijación: dónde se sostiene el dato

El soporte nunca es inocente: tiene su vida útil, sus restricciones de relectura, sus dependencias técnicas. Mal elegido, expone la obra al **dato muerto** —el caso en el que el soporte sobrevive pero ya nadie puede interpretar lo que lleva, por falta de máquina o de formato legible—. El canalero elige su soporte según la duración de uso esperada, las restricciones de puesta en común, y el riesgo de caducidad del propio soporte.

Los formatos: preferir lo abierto y lo duradero

La primera regla de fijación es **no encerrarse en un formato propietario** cuya perennidad no se domina. Se privilegian los formatos abiertos, legibles durante mucho tiempo y por todos:

- **JSON** y **CSV** para los datos estructurados de intercambio y de archivo: universales, legibles por una máquina y a simple vista, reimportables en todas partes.
- **Markdown** para la documentación (y, mediante *pandoc*, la producción de PDF): texto plano que sobrevivirá a todos los procesadores de texto.
- **SQLite** como base de datos en un simple archivo: una base completa, sin servidor, transportable, abierta, legible dentro de diez años. Es a

menudo el soporte ideal para un emprendedor en solitario —robusto y sobrio—.

- **PostgreSQL** cuando el volumen, la concurrencia de acceso o la puesta en común lo exigen: una base de servidor probada y abierta.

El canalero **documenta el formato elegido** para que las generaciones siguientes —o simplemente el cliente, más adelante— puedan todavía leer lo que se escribió. Un formato documentado es una clave de lectura conservada.

La inmutabilidad: conservar la historia

Para los datos vinculantes (facturación, trazabilidad, seguimiento legal), una buena práctica de fijación consiste en **no sobrescribir nunca**, sino en **añadir**. En lugar de modificar un presupuesto en su sitio, se registra una nueva versión y se archiva la antigua con su marca de tiempo. Inspirada en los registros (*ledgers*) inmutables, esta disciplina conserva un **registro** de todo lo que ha sucedido, lo que es valioso para la conformidad fiscal y para reproducir el historial. Técnicamente, se identifica cada registro mediante un identificador único con marca de tiempo (de tipo *UUID v7*) y se puede sellar cada versión mediante una suma de control (un *hash* de su contenido), de modo que toda alteración se detecte.

En el óptimo, siempre. La inmutabilidad tiene un costo (volumen, complejidad). Se la reserva a lo que la merece —los datos vinculantes, trazables, duraderos— y se conserva un almacenamiento simple para lo demás. Inútil construir en mampostería un canal donde bastaba una zanja.

El sistema de eventos: hacer circular el acto de inscripción

En una obra viva, cada gesto —crear un presupuesto, reclamar a un cliente, cerrar un plazo— es un **acto de inscripción**: un hecho de la actividad, fechado y fijado —la mayoría de las veces instituido por el acto mismo (un presupuesto emitido, un plazo cerrado) más que medido—. Para que una obra sea algo más que la suma de sus funciones, esas inscripciones deben **circular** entre los módulos sin que ninguno dependa de otro. Es el papel de un sistema de eventos.

El **evento** es la forma de software del acto de inscripción. Se le da un nombre —función.entidad.acción—, una fecha, una fuente, y los datos del hecho. Es, en el sentido en que la trilogía lo entendía, un **datum**: «*lo que es dado —para una ocasión ulterior—*». Un módulo emite el evento sin saber quién lo escucha; otros se suscriben a él para desencadenar, a partir de él, una acción nueva —manual, automatizada o delegada—. Una misma inscripción puede abrir varias ocasiones.

Dos órganos lo llevan. El **bus** transporta la inscripción, aquí y ahora, de los emisores hacia los suscriptores: es el transporte local, **desacoplado**, que permite injertar una función sin tocar las demás. El **diario** conserva la sucesión de las inscripciones: es el flujo de observación vuelto duradero, inmutable, añadido al final (*append-only*) —la **fuentes fiable** de la actividad—. El estado corriente de una ficha no es más que una lectura cómoda; es el diario el que hace fe. En él se reproduce, se consolida, se audita.

Cada acto se abre y se cierra con una inscripción. Antes de actuar, se inscribe la intención (*init*) con sus parámetros: si el acto falla gravemente, se remonta a su origen. Cumplido el acto, se inscribe su cierre (*finish*); fallido el acto de forma controlada, su fallo (*fail*) —que un módulo puede escuchar para alertar en el acto—. El bucle del acto queda él mismo consignado.

Salvaguarda sobre la forma. Todo esto está fechado: un bus en memoria, un diario local, para una obra de un solo puesto. El fondo sigue siendo la ontología —el acto de inscripción, el datum, el flujo de observación—. Mañana, el bus podrá estar distribuido y el diario archivado en otro lugar; la inscripción, en cambio, sigue siendo lo que es: un hecho dado para una ocasión ulterior.

El archivo y las obligaciones

Algunos datos deben conservarse por obligación legal (las facturas durante varios años). El **archivo** es la conservación de larga duración de datos inactivos pero útiles u obligatorios. Se distingue del almacenamiento corriente: se lo puede comprimir, cifrar, guardar aparte, incluso programar su **supresión automática** al término legal (por ejemplo, un borrado después del periodo de conservación previsto, en aras de la conformidad con el RGPD). El canalero inscribe esas reglas desde la concepción.

El transporte local: dirigir la propia actividad

La circulación más cotidiana para el emprendedor en solitario no va a ninguna parte lejana: une sus propias tomas con su propio cuadro de mando. Es el transporte **interno**, que transforma una actividad dispersa en una dirección legible.

Arquitectónicamente, es el movimiento *del núcleo de negocio hacia la lectura*: el dato fabricado (capítulo 5) se consolida —media, total, estado, tendencia— y luego se encamina hacia la pantalla que lo volverá visible (capítulo 7). En *local-first*, todo esto sucede **en el equipo del cliente**, sin servidor remoto: el dato bruto no sale de la empresa, la dirección es instantánea, y funciona sin conexión.

El mecanismo típico: cada movimiento introducido pone al día un indicador

de estado; el cuadro de mando lee el indicador consolidado, nunca la masa bruta. El emprendedor en solitario no «consulta un diario de todas sus operaciones» —ve *dónde está*—. Es la consolidación (tomo 1) al servicio de la ventana del lector (tomo 3).

Cuando varios equipos o varias aplicaciones del mismo cliente deben coordinarse en una misma máquina o en una misma red local, entran en juego herramientas de circulación interna: una base compartida, una caché rápida (de tipo Redis) para los estados que se leen con frecuencia, una cola de mensajes (de tipo RabbitMQ) para hacer pasar eventos de un servicio a otro sin acoplarlos, tareas programadas (*cron*) para las consolidaciones periódicas. Para un emprendedor en solitario, la mayoría de las veces, **nada de esto es necesario**: un archivo de base y un poco de JavaScript bastan. Solo se movilizan esas herramientas cuando se supera realmente un umbral de complejidad —en el óptimo, una vez más—.

El transporte remoto: compartir el dato

El segundo uso de la circulación va hacia el exterior: transmitir a un tercero (el gestor contable, un cliente, un socio), o hacer entrar el propio dato en un intercambio más amplio. El tomo 2 da la **tipología mínima** del transporte, y cada modo pide sus herramientas.

- **Desplazar** — llevar el dato de un lugar a otro (una copia de seguridad, una migración). Herramientas: exportación de archivos, copia cifrada.
- **Transmitir** — hacerlo pasar a un destinatario designado (enviar el resumen anual al gestor contable). Herramientas: exportación **CSV/PDF**, mensaje, depósito seguro.
- **Hacer accesible** — ponerlo a disposición de quien venga a consultarlo. Herramientas: una **API** (interfaz programable) con autenticación.

El transporte solo se cumple si la parte del lector es practicable

Es la regla de oro, heredada del tomo 2: transmitir no es depositar. Entregar al gestor contable una exportación en bruto que tendrá que volver a tratar no es transmitir —es depositar—. El canalero **consolida y pone en el formato que espera el destinatario** para que este pueda cumplir su parte sin esfuerzo. Para el emprendedor en solitario que comparte, eso quiere decir: una exportación limpia, fechada, en el formato que el tercero sabe abrir, acompañada de lo necesario para interpretarla.

La API y el protocolo MCP: abrir el propio dato a los agentes de IA

Una necesidad nueva de 2026: volver el propio dato consultable por **agentes de IA**. Cada vez más clientes quieren que una IA pueda leer su agenda, sus pedidos, su actividad, para asistirlos. La respuesta técnica es desarrollar una **API cliente** —puntos de acceso que exponen, de forma controlada y autenticada, los datos deseados— y, para que dialogue con los modelos de IA según un estándar, hacerla conforme al protocolo **MCP** (*Model Context Protocol*), que normaliza la interconexión entre una fuente de datos y un agente.

Del lado de la arquitectura, esta API no es más que un **adaptador** más, de entrada y de salida del hexágono: se añade sin tocar el núcleo de negocio. Es la ilustración perfecta de la perennidad por la arquitectura: un programa de veinte años, cuyos datos estaban prisioneros de su interfaz, encuentra una segunda vida exponiendo una API MCP —sin reescribir su lógica—.

Las alteraciones del transporte, por prevenir

El transporte no es neutral: puede perder unidades, metadatos, contexto; lleva tiempo, durante el cual el dato envejece. El canalero conoce esas alteraciones y las previene: incluye el corpus en la exportación, pone marcas de tiempo, señala la frescura, verifica que un dato fresco en la fuente no llegue vencido. Un taller documenta, para cada tipo de obra, el **modo de transporte probado** y sus condiciones de recepción.

La multiplicación: un hecho compartido ya no es un objeto único

Particularidad del dato: hacerlo viajar puede **multiplicar sus portadores** (una copia en otro lugar sin dejar de existir aquí). Eso cambia la manera de razonar sobre la fiabilidad, la propiedad, la seguridad. Un hecho difundido en múltiples lugares no se protege como se guarda un ejemplar único. Antes de compartir, el canalero —y el cliente— se preguntan: *una vez fuera, este dato no volverá; ¿quién podrá hacer qué con él?* La circulación remota implica un compromiso; se decide con conciencia, no por defecto.

Reutilizar sin transmitir: el momento objetual, y cómo lo sostiene el fondo

La tipología del transporte deja un caso de lado: aquel en el que el canalero **reutiliza**, dentro de la obra, un dato que ya posee —sin enviarlo a nadie—. El presupuesto que retoma la ficha de un cliente es el ejemplo de ello: el consolidado pasa de un soporte a otro —de la ficha al presupuesto—, pero permanece **en el mismo destinatario**, el artesano. Es una navegación

espacial (tomo 2), pero *interna y sin destinatario tercero*: no se crea ningún portador suplementario hacia el exterior, el dato no se cede, ni siquiera se comparte. Se reutiliza; no se transmite.

Este gesto merece nombrarse, porque toca la ontología. Reutilizar un consolidado como un punto de referencia estable que se designa, que se copia, que se vincula, *es tratarlo, el tiempo de un uso, como un objeto*. Los **cuatro presupuestos tácitos** que el tomo 1 atribuye a la ontología objetual se verifican en él localmente: el consolidado parece **preexistir** al presupuesto, se lo tiene por **estable**, se le reconoce un **valor de uso**, se lo **posee** y se lo manipula. Es el **momento objetual** de la obra.

¿Hay que alarmarse por ello? No —y de eso se trata—. El tomo 1 ya lo dijo: el *datum* es en él «lo que puede ser **recogido** en la ocasión siguiente», un **residuo de proceso**. Recoger un consolidado no traiciona lo procesual: es su mecanismo mismo. El dato consolidado es, *por definición* (tomo 1), «reutilizable en otros contextos». El momento objetual no es, pues, una vuelta a la ontología dominante: es una **forma fechada** que el **fondo** procesual prevé, autoriza —y **acota**—.

Tres límites, que la obra vuelve visibles allí donde la ontología objetual corriente los olvida:

- **La caducidad.** El consolidado reutilizado es un residuo desprendido de su flujo de observación. El nombre del cliente, congelado en el presupuesto, ya no sigue a la ficha viva. Cuando la ficha desaparece, la obra lo **señala** —un *vínculo roto*— en lugar de fingir que el punto de referencia sigue valiendo. La caducidad se muestra, nunca se oculta; el nombre congelado, por su parte, sigue siendo exacto para ese presupuesto.
- **La inapropiabilidad del flujo.** Se posee el **soporte** y el **ejemplar** —el presupuesto, su consolidación—, nunca el flujo del que proceden (cuarta inversión, tomo 1); esta no rivalidad es ontológica y no prejuzga el régimen jurídico de la ficha ni del presupuesto. Suprimir el contacto no rompe el presupuesto: nunca se ha poseído más que el residuo.
- **La recontextualización.** El *datum* es **relacional** (tomo 1): el identificador retomado solo significa «este cliente» *en este presupuesto*, como una relación nueva —no como una sustancia transportada intacta—.

La regla de práctica cabe en una frase: *reutilizar el residuo sin olvidar nunca que es residuo*. Lo que separa la obra del enfoque dominante —los *data lakes*, «mis datos» tenidos por un bien perenne— no es que se prohíba manipular el consolidado como un objeto; es que nunca pierde de vista que el consolidado puede caducar para el uso que hace de él, que no posee el flujo, que recontextualiza en lugar de transportar sentido ya hecho. El momento objetual es legítimo *mientras el fondo lo sostenga*.

Salvaguarda sobre la forma. Todo esto está fechado: un identificador, un nombre congelado, un marcador de vínculo roto. El fondo permanece: se

reutiliza un residuo, no se posee un flujo; manipular el dato como un objeto está permitido el tiempo de una ocasión, a condición de verificar que sigue valiendo para el uso que se hace de él. Es aquí, en el punto de reutilización, donde la ontología objetual opera en la obra —y es aquí, exactamente, donde el fondo procesual debe sostenerla—.

El flujo entrante: tomar el correo

Hasta aquí, lo que la obra inscribía venía del flujo de la actividad del taller: un presupuesto emitido, un recordatorio, una factura. Pero llega otro flujo, de fuera: el correo. Un mensaje recibido no es un objeto guardado en un buzón; es, también él, un flujo de lo real —unas palabras dirigidas a alguien, fechadas, que no se repetirán de forma idéntica—. La obra no hace de él una excepción: lo trata como todo flujo, mediante una **toma**.

Esta toma tiene una exigencia propia, que la forma fechada vuelve visible. El correo no vive en la obra: vive en una fuente, fuera, que la obra no posee. Tomar, aquí, es **leer sin alterar la fuente**. El puesto recoge los mensajes en solo lectura: no marca nada, no desplaza nada, no consume nada en el servidor; hace una copia de ellos y deja intacto el flujo. Es la **inapropiabilidad del flujo** prolongada en la entrada: uno no se adueña del correo, recoge de él un ejemplar. La fuente sigue siendo la de otro; la obra solo conserva de ella un depósito, fechado, en su propio soporte —lo que no dice nada, por sí solo, de lo que tiene derecho a hacer con él—.

Una vez tomado, el mensaje se **inscribe**. Entra en los dos tiempos del dato como cualquier observación: un primer tiempo, el mensaje tal como se recibió —el remitente, el asunto, el cuerpo, intactos—; luego un segundo tiempo, en el que se recoge en la ocasión siguiente —leído, clasificado, respondido—. La obra lleva el historial de ese itinerario: ¿este mensaje es nuevo, leído, tratado? El buzón de fuera lo ignora; la obra, en cambio, lo sabe, porque no ha inscrito el correo, sino *su lectura por el artesano*.

Queda la lectura misma. Presentado al **lector situado**, el mensaje pide el acto de lectura: percibir que está ahí, distinguirlo de los demás, descifrarlo. En este último gesto, el desciframiento, es posible un apoyo —y es aquí donde hay que ser preciso sobre lo que el apoyo es, y no es—. Una consulta puede proponer una **calificación** (¿de qué orden es este mensaje?) y un **borrador** de respuesta. Esas propuestas no son el mensaje: son datos **verosímiles**, no datos observados. El mensaje recibido es observado; lo que se infiere de él es verosímil, y debe seguir siéndolo —mostrado como tal, *por validar*—. El artesano relee, ajusta, decide. El borrador solo tiene efecto por su acto; nada sale sin que lo haya hecho suyo. El apoyo descifra una primera vez; no lee en lugar del lector, y no es él quien responde de lo que sale.

Todo esto pertenece al fondo, y el paso por el correo electrónico no es más que su forma. Que el flujo entrante se recoja mediante tal protocolo hoy, mediante tal otro mañana, no cambia nada de lo que se dice: un flujo

de lo real llega de fuera; se lo toma sin alterarlo; se lo inscribe con su historial; un lector situado lo descifra, asistido pero no sustituido. La vía está fechada. El gesto, en cambio, es el de siempre.

Capítulo 7 — Leer el dato: el cuadro de mando de dirección

Tercer tiempo, tercer gesto: **la lectura**. Es ahí donde el dato se vuelve útil, o permanece sin leer. El tomo 3 lo estableció: *sin lectura, nada se actualiza*. La herramienta por excelencia de este tiempo, para el emprendedor en solitario, es el **cuadro de mando de dirección**. Este capítulo dice cómo concebirlo para que se lea realmente —es decir, para que respete la **ventana del lector**—.

El lector está situado, su ventana es finita

Ninguna lectura es de ninguna parte. El emprendedor en solitario lee su actividad desde una situación precisa: poco tiempo, una atención ya solicitada, un corpus de competencias dado, a menudo una pantalla pequeña entre dos tareas. Su **ventana de lector** es estrecha. El canal mejor cavado, transportado con la mayor fidelidad, no sirve de nada si lo que entrega supera esa ventana. *Un dato que no se tiene la capacidad de examinar no se lee, se crea lo que se crea*.

La concepción de un cuadro de mando es, pues, ante todo, un trabajo sobre la ventana: no mostrar más que lo que puede leerse, en el tiempo y con la atención de que dispone el cliente. Es la contrapartida exacta, del lado de la recepción, del óptimo observacional del lado de la fabricación: ni ciego, ni anegado.

La selectividad: menos, pero lo justo

La lectura es **selectiva** por naturaleza: no se lee todo, se elige. Un buen cuadro de mando elige *en lugar del lector* lo que merece su escasa atención. Responde, de un vistazo, a tres preguntas y solo a ellas:

1. **¿Dónde estoy?** El estado corriente, consolidado en señales simples: cifra de negocio del mes frente al objetivo, tesorería, pedidos en curso, margen.

2. **¿Qué debo hacer?** Las acciones que piden una decisión: un presupuesto que reclamar, una factura con retraso, unas existencias por debajo del umbral.
3. **¿Qué se está desviando?** Las alertas: un umbral al que uno se acerca, una tendencia que se invierte, una anomalía.

Todo lo demás —la masa de las tomas en bruto— sigue siendo accesible *en profundidad*, pero no se muestra en la superficie. El cuadro de mando entrega el **dato consolidado**, no el diario. La regla de Lavoisier del canalero: *consolidar la actividad en señales simples, en lugar de entregar una masa en bruto indescifrable*.

Los indicadores: elegir los adecuados

Un **indicador** es un valor calculado a partir de los datos, destinado a la dirección y a la decisión. El canalero distingue sus tipos y los elige con el cliente, nunca en su lugar:

- **Indicadores de estado** — miden el instante: número de pedidos con retraso, nivel de existencias, tesorería.
- **Indicadores de tendencia** — miden el movimiento: evolución de la cifra de negocio, tasa de conversión presupuesto → pedido.
- **Indicadores de umbral** — sitúan respecto de un límite: progresión hacia el límite de cifra de negocio, hacia un objetivo (véase la gamificación, capítulo 4).

Para las actividades en las que la dirección financiera es demasiado volátil o demasiado tardía, uno puede apoyarse en **indicadores físicos** más que monetarios —horas de trabajo, volumen producido, kilómetros, tasa de utilización de un vehículo, energía consumida—, que describen la actividad de manera estable e inmediata, mientras la traducción financiera se hace a posteriori. Este enfoque por el flujo físico da una dirección robusta, poco sensible a los imprevistos, y a menudo más elocuente para un artesano que los asientos contables.

Caso real. Una empresa de transporte pierde dinero en entregas mal llenadas. El indicador adecuado no es global: es el **rendimiento unitario de cada vehículo según su tipo de carga** (tasa de utilización, costo/beneficio). Bien elegido, el indicador transforma una intuición vaga en una decisión argumentada: qué medios conservar, cuáles reducir. Elegir el indicador ya es la mitad de la obra.

La frescura de la lectura: irrigar el cuadro de mando

Una lectura se hace en un instante dado; el flujo, en cambio, no se detiene. Un cuadro de mando consultado y luego dejado tal cual pierde pronto su **frescura**: muestra un estado pasado del flujo de observación mientras la actividad, por su parte, ha continuado. El lector situado no debería ni

preguntarse si lo que ve está al día, ni recargar la página a mano para asegurarse. *Una lectura que va con retraso respecto del flujo ya no lee del todo el presente: lee un estado pasado.*

El principio, del lado de la recepción, es simple: la superficie de lectura debe estar **irrigada** por el flujo. Lo que se inscribe aparece, allí donde el lector mira. El cuadro de mando no es una instantánea que se refresca de memoria; es una superficie que se conserva fresca al ritmo del flujo de observación. Es la contrapartida exacta, en el tercer tiempo, del óptimo observacional del primero: ni congelado, ni parpadeante.

Concretamente —y de forma fechada—, cada acto se inscribe como un evento; la superficie escucha el flujo y se redibuja cuando crece. Tres salvaguardas sostienen esta irrigación dentro de la **ventana del lector**. Primero, *no refrescar más que lo que está a la vista*: la ventana es finita, inútil recalcular una vista que no se mira. Después, *no sobrescribir nunca un gesto en curso*: el lector es también un actor; si su mano está en un campo, la irrigación espera —no retoma lo que está inscribiendo—. Por último, *agrupar las ráfagas*: un refresco por ciclo, no uno por evento, sin lo cual el parpadeo desbordaría la ventana con la misma seguridad que una masa en bruto.

Dos complementos terminan el dispositivo: un «**refrescar**» al alcance de la mano, para el lector que quiere forzar la puesta al día, y un refresco al **cambiar de vista**, ya que el tiempo ha podido pasar. Así el cuadro de mando permanece en el óptimo: ni con retraso respecto del flujo —caducidad silenciosa de la lectura—, ni desbordante.

Salvaguarda sobre la forma. Todo esto está fechado: un bus en memoria, un redibujado local. El fondo permanece: *la lectura debe seguir siendo fresca respecto del flujo*. Los medios de la irrigación podrán cambiar; la exigencia de frescura, en cambio, no caduca.

Las herramientas de lectura, y sus sesgos

Las herramientas que dan a ver —gráficos, visualizaciones, consultas, cruces, estadística descriptiva— **transforman lo que leen**. Una visualización elige sus ejes, sus escalas, sus agregaciones; puede revelar una tendencia real o sugerir una falsa. El canalero conoce esos sesgos como conoce los de sus sensores, e informa de ellos al cliente. Concretamente, para el emprendedor en solitario, uno se atiene a representaciones honestas y legibles:

- **cifras clave** destacadas (lo estrictamente necesario);
- **curvas** simples para las tendencias, a una escala que no engañe;
- **barras de progreso** hacia los objetivos y los umbrales;
- **listas de acciones** ordenadas por urgencia;
- un **código de colores** sobrio (por ejemplo: verde/naranja/rojo) para el estado y la alerta.

Todo esto se realiza en **HTML/CSS/JS nativo** (capítulo 3): no hace falta una biblioteca pesada para dibujar una barra, una cifra, una curva simple. La herramienta de lectura sigue siendo ligera, rápida, abrible en todas partes, y reversible.

Las tres posturas del lector, y lo que dictan a la interfaz

El tomo 3 describe tres maneras de habitar el sobrecaudal. Cada una se traduce en elecciones de interfaz.

- **Encuadrar** — restringir o desplazar la propia ventana para leer de verdad lo que importa. *En la interfaz*: dejar que el cliente elija sus indicadores, ocultar lo demás, ir a lo esencial. El cuadro de mando es un encuadre instrumentado.
- **Delegar** — confiar la lectura a un dispositivo (un automatismo, un agente de IA) cuya **situación se hereda**. *En la interfaz*: permitir la delegación (un resumen automático, una alerta inteligente) señalando *quién* ha leído y con qué límites. Se delega la lectura, no la responsabilidad.
- **Aceptar** — consentir con lucidez en no leerlo todo. *En la interfaz*: la sobriedad. No culpabilizar al cliente por no verlo todo; algunos datos de frescura breve no estaban destinados a ser leídos, y eso no es un fracaso.

La peor postura, no escrita, es la cuarta: *creer que se ha leído lo que solo se ha visto pasar*. Un cuadro de mando que da la ilusión del dominio sin darlo realmente es nocivo. La probidad de la herramienta consiste en volver legible *lo que es*, y en dar acceso, bajo la superficie, al dato y a su modo de cálculo.

Mantener la capacidad de leer

Salvaguarda repetida porque es central: la herramienta de lectura vuelve al dirigente **más** capaz de leer sus datos, nunca lo dispensa de ello. Cada indicador mostrado puede «abrirse»: de dónde viene, sobre qué periodo, calculado cómo, con qué margen. Delegar en la interfaz la fatiga del cálculo es un buen arbitraje; delegar la comprensión, uno malo. El cuadro de mando es una lámpara que el cliente aprende a sostener, no un oráculo que piensa en su lugar.

La ventana nombrada, en la firma

Toda obra de lectura se entrega nombrando la **ventana de lector** a la que apunta: a quién está destinado este cuadro de mando, en qué condiciones es legible, qué hay que saber para interpretarlo. Este campo figura en la ficha de firma (capítulo 9). Concebir para un lector situado, y decirlo,

forma parte de la entrega —es lo que distingue una obra acabada de un dato arrojado a la pantalla—.

Capítulo 8 — La IA local: el operador semántico del canalero

La IA generativa es, para el canalero de 2026, la herramienta más potente y la más delicada. Atraviesa los tres tiempos: extrae y califica (fabricación), ayuda a transmitir y a resumir (circulación), descifra para el cliente lo que este le delega (recepción). El tomo 1 la situó con precisión: el gran modelo de lenguaje condensa en una entidad los tres papeles históricamente separados —la **biblioteca** donde se ordena la información, el **bibliotecario** que la encuentra, el **narrador** que la restituye—. Es un **salto sistémico**, con sus beneficios y su contrapartida. Este capítulo dice cómo emplearla como artesano: localmente, con mesura, y sin desposeer al cliente.

Lo que la IA generativa sabe hacer para el emprendedor en solitario

Como **operador semántico**, un LLM trata informaciones no estructuradas sin que haya que mantener diccionarios ni reglas complejas. Las tareas de gestión que automatiza, en todo o en parte:

- **Calificar y clasificar** — ordenar correos electrónicos, categorizar gastos, analizar una opinión de cliente.
- **Extraer y estructurar** — sacar de una factura recibida o de una hoja de pedido los campos útiles, con una salida **JSON** utilizable.
- **Generar** — redactar un borrador de presupuesto, de recordatorio, de respuesta tipo, un acta.
- **Sintetizar** — resumir un expediente, una jornada de actividad, una serie de intercambios.

El principio rector, en todas partes: **la IA propone, el humano decide**. Para las tareas repetitivas, estructuradas y de bajo riesgo, la automatización puede ser completa. Para todo lo que es crítico, sensible o jurídico —la validación de un presupuesto, de una factura, de un contrato, de una nómina— **la decisión y la validación siguen siendo humanas**, sin excepción.

Por qué local: soberanía, costo, dominio

El canalero privilegia la **IA local** —un modelo que funciona en la máquina del cliente o del taller, y no en una nube remota—. Tres razones, que son los tres pilares del oficio:

- **Soberanía y confidencialidad.** El dato —a menudo sensible— **no tiene por qué salir de la empresa**. Es la arquitectura *edge / local-first* aplicada a la IA: tratar lo más cerca posible de la fuente. La superficie de ataque disminuye; la conformidad resulta simplificada.
- **Dominio de los costos.** Ni suscripción por consulta, ni factura que crece con el uso. El costo es el del material, amortizado.
- **Perennidad e independencia.** No se depende de la política de un proveedor remoto que puede cambiar sus precios, sus condiciones, o cerrar.

Las herramientas de 2026 vuelven esto accesible: **modelos abiertos y ligeros** (de la familia de Llama, Phi, Gemma y otros), capaces de funcionar en una computadora portátil correcta o en un pequeño servidor; y **entornos de ejecución locales** (de tipo *Ollama*) que los instalan y los sirven en unas pocas órdenes. Se elige el modelo según la necesidad justa: un modelo pequeño para clasificar correos, un modelo algo más capaz para la extracción o la redacción. Inútil apuntar al más grande: en el óptimo, una vez más.

Cuándo la nube sigue siendo legítima. La IA local es la regla, no un dogma. Una tarea pesada y puntual, o una necesidad que el material local no cubre, puede justificar un recurso medido a un servicio remoto —a condición de decírselo al cliente, de no hacer transitar por él ningún dato sensible sin su consentimiento informado, y de aislarlo tras un adaptador reemplazable—. El arbitraje se plantea, como siempre.

Combinar la IA y las reglas deterministas

La IA no siempre es la herramienta adecuada. Para las tareas **estructuradas, deterministas, en las que prima la fiabilidad** —un cálculo de IVA, la aplicación de una regla de gestión, una validación de formato—, los métodos clásicos (reglas explícitas, automatización de tareas repetitivas de tipo RPA, diccionarios) siguen siendo superiores: previsibles, verificables, sin alucinación. El canalero **combina**: la IA para lo semántico y lo difuso, la regla determinista para lo exacto y lo crítico. A menudo, la salida de un LLM está *controlada* por una regla determinista antes de ser aceptada —indicadores, umbrales, controles de coherencia detectan la anomalía—.

El método de automatización progresiva

Equipar con IA a un emprendedor en solitario no se hace de una vez. El método es **progresivo y colaborativo**: empezar en pequeño, validar, extender. Seis etapas.

1. **Auditoría inicial.** Mediante una entrevista guiada, recoger las tareas diarias, semanales, mensuales. Clasificarlas según cuatro criterios —tiempo consumido, penosidad, riesgo de error, impacto en la calidad— y **priorizarlas** en una matriz. Entregable: una lista priorizada de las tareas candidatas, validada por el empresario. Se apunta primero a lo que consume tiempo, es penoso y es fuente de errores.
2. **Cartografía de las tareas automatizables.** Cruzar la lista con la viabilidad (datos disponibles, material, competencias) y proponer, para cada tarea prioritaria, la solución (modelo adaptado, herramientas complementarias, método de validación). Entregable: una tabla por tarea —solución, material requerido, beneficios, riesgos—.
3. **Prototipado y prueba.** Sobre **una o dos** tareas simples y de impacto (clasificar los correos, generar borradores de respuesta): instalar las herramientas, configurar, **formar al empresario**, y luego probar de dos a cuatro semanas midiendo ganancias de tiempo, errores, percepción. Entregable: un informe de prueba.
4. **Despliegue progresivo.** Por oleadas —dos tareas nuevas cada uno o dos meses—. Objetivos frecuentes: gestión de los correos, seguimiento de clientes y proveedores, existencias y tarifas, creación de documentos, vigilancia. Insistir en la **validación manual** y en la comprensión de los límites. Entregable: un cuadro de mando de seguimiento.
5. **Optimización.** Afinar los prompts, ajustar las reglas, automatizar tareas más complejas (análisis de las ventas, informes mensuales), hacer evolucionar el material si hace falta. **Fomentar la autonomía**: el cliente sabe modificar reglas simples, la documentación es clara.
6. **Mantenimiento.** Poner al día los modelos, proteger la instalación, hacer copias de seguridad, hacer revisiones periódicas. Entregable: unas modalidades de seguimiento claras.

Las frases literales que guían la relación con el cliente resumen su espíritu: «*La IA está ahí para ayudarle, no para sustituirle.*» — «*Empiece en pequeño, piense en grande.*» — «*Usted decide qué se automatiza y cómo.*»

La contrapartida, nombrada

El canallero dice la verdad, incluido lo que un vendedor callaría. Delegar el tratamiento de un dato en una IA es **alejarse un poco de él**. El tomo 1 lo planteó: la fusión de los tres papeles (biblioteca, bibliotecario, narrador)

hace desaparecer lo que su separación hacía posible —la trazabilidad (saber de dónde viene una información), el retorno a la fuente, la crítica, el contrapoder de un papel sobre otro—. Delegar la propia lectura en un modelo es **heredar su situación**: su corpus congelado, sus ángulos muertos, su frescura. *Se cambia el anegamiento por una dependencia, y todo consiste en saber lo que vale aquello de lo que uno se vuelve dependiente.*

La salvaguarda está inscrita en la concepción: **la herramienta mantiene la capacidad de leer del dirigente, nunca lo dispensa de ella**. Una IA que resume debe dejar acceder a lo que ha resumido. Una IA que clasifica debe dejar revisar su clasificación. La probidad de la delegación consiste en conocer la situación que se hereda —y en no confundir nunca la lectura delegada con una lectura sin situación, desde lo alto—. Es la diferencia entre un emprendedor en solitario equipado y un emprendedor en solitario desposeído.

Capítulo 9 — Mantener: mantenimiento, seguridad, rituales

Un flujo de observación se concibe, se despliega y luego se abandona a sí mismo con demasiada frecuencia. Esta negligencia degrada las obras: un sensor se desvía, un formato caduca, un canal se llena de datos que ya nadie lee. La sexta familia de herramientas —el **mantenimiento**— conserva vivo el canal en la duración. Se acompaña de una exigencia permanente, la **seguridad**, y se estructura mediante **rituales** regulares. Este capítulo cierra el bucle de las seis familias.

Las herramientas de mantenimiento

Son los dispositivos que permiten sostener un canal a lo largo del tiempo. El canadero las **integra desde la concepción**, nunca a posteriori.

- **Alertas de deriva de calibración.** Un instrumento de toma (sensor, script, prompt de extracción) se desvía. Se vigila la brecha aguas arriba/aguas abajo y se señala cuando la medida se aparta.
- **Indicadores de frescura.** Cada dato lleva su edad. El sistema señala cuando un dato ha superado su ventana de validez para el uso que se hace de él, a la vista de la velocidad de su real.
- **Nueva toma periódica.** Para los datos sujetos a caducidad, se planifica la renovación (tareas programadas, *cron*): la frescura no se recupera, se rehace.
- **Controles de coherencia aguas arriba/aguas abajo.** Se verifica que lo que sale del canal sigue siendo coherente con lo que entra en él, para detectar una ruptura silenciosa.

Para el emprendedor en solitario, estas herramientas siguen siendo **sobrias**: una alerta discreta, un indicador de frescura en el cuadro de mando, una tarea nocturna de consolidación. El mantenimiento no es una fábrica; es el cuidado regular del carpintero que revisa su techumbre.

La seguridad y la higiene digital

La seguridad no es una opción, sobre todo para el **dato sensible** (datos de contacto de los clientes, datos financieros, datos sometidos al RGPD). La **higiene digital** es el conjunto de las prácticas que conservan los datos organizados, protegidos y conformes. Los pilares, en la lógica *local-first*:

- **Cifrado.** Los datos sensibles se cifran en reposo (en el disco, en el archivo —por ejemplo en AES-256—) y en tránsito (intercambios protegidos). Una copia de seguridad no cifrada es una fuga a la espera.
- **Copia de seguridad 3-2-1.** Tres copias de los datos, en dos soportes diferentes, una de ellas fuera del sitio. Es la regla que protege de la **desaparición** —la segunda muerte del dato (tomo 1), distinta de la caducidad: aquí el soporte se degrada, el formato se vuelve ilegible, o se pierde el acceso—. Un disco externo cifrado y una copia remota cifrada bastan a menudo.
- **Accesos restringidos.** Cada dato solo es accesible para quien lo necesita; la autenticación protege el acceso (por ejemplo OAuth2 para los servicios conectados).
- **Trazabilidad.** Dejar constancia de quién ha hecho qué, cuándo —lo que la fijación inmutable (capítulo 6) facilita—.
- **Minimización y plazo de conservación regulado.** No tomar ni guardar más que lo que sirve; programar la supresión al término legal. Menos dato conservado, menos superficie de riesgo —y es también sobriedad—.

La soberanía como seguridad. La mejor salvaguarda sigue siendo estructural: si el dato en bruto no sale de la empresa (tratamiento local, transferencias controladas), la superficie de ataque se reduce en la fuente. La arquitectura sobria no es solo económica; es segura.

La reserva: el doble retirado del ciclo de uso

Un dato vive en el ciclo de uso de la obra: sirve en él, se consolida en él, caduca en él. Mientras está expuesto en él, puede perderse —una falsa maniobra, un soporte defectuoso—. Para prevenir la pérdida, se le da un doble sustraído a ese ciclo: una copia **congelada**, escrita una vez y nunca reescrita, que no se lee en el uso corriente y que solo se moviliza para restablecer lo que ha desaparecido. Es la **reserva**.

La reserva no está ni viva ni caducada: está retirada del ciclo de uso —no del flujo—. No existe ningún fuera-del-flujo (es la lección de fondo del tomo 2): el contenido de la reserva, como toda toma, sigue apartándose de lo real, que ha continuado sin él; lo que queda suspendido es su exposición al uso y a la falsa maniobra, no su caducidad. Se la distingue de la copia de seguridad 3-2-1, que multiplica las copias en lugares distintos para prevenir el siniestro material. La reserva, por su parte, apunta a la

continuidad en el tiempo: guarda, a intervalos, el estado completo de la obra, y hace rotar sus **soportes** —se conservan las últimas generaciones (por ejemplo la corriente y las dos anteriores), a varias temporalidades (el día, la semana, el mes); más allá, la más antigua sale—. Una misma instantánea puede servir a varias temporalidades a la vez: solo se duplica una vez por periodo.

El gesto del canalero consiste aquí en **regular la cadencia y el número de generaciones** según lo que puede perder sin daño, y luego en dejar que la reserva se sostenga sola. Conservada por el puerto de almacenamiento soberano, obedece a las mismas reglas que lo demás: sin puesto, espera; cuando el puesto vuelve, se deposita. La reserva nunca sirve a la actividad presente —es su virtud: lo que no sirve no corre el riesgo de ser alterado por el uso—.

Los rituales del taller

Un taller vive por sus rituales —momentos regulares, articulados, en los que se operan los arbitrajes y los cuidados—. Sin ellos, la obra se desvía; con ellos, el canal sigue vivo. Dos rituales estructuran el mantenimiento.

La revisión del canal

Con una periodicidad fija (mensual, trimestral según los casos), se revisa cada canal mantenido. Orden del día tipo:

1. Volumen y calidad de las tomas desde la última revisión.
2. Calibración de los instrumentos: ¿deriva constatada?
3. Pertinencia de los criterios con respecto al uso actual.
4. Corpus de referencia: ¿sigue siendo suficiente?
5. Bucles eventuales: ¿modifica el canal lo que observa?
6. Señales de caducidad: ¿qué refrescar?
7. Carga de mantenimiento: ¿sostenible?
8. Decisiones y arbitrajes: ¿quién responde?
9. **Recepción efectiva:** ¿leen el canal aquellos a quienes está destinado, o su dato permanece sin leer?

La revisión de la recepción

Contrapartida de la anterior, del lado del lector. ¿Se **recibieron** las obras entregadas? ¿Era practicable su parte de lector? ¿Llegaron dentro de su ventana de validez, o vencidas durante el trayecto? Cuando una obra no se lee, se diagnostica el camino:

- **incapacidad** del destinatario — camino remediable: se vuelve al cliente más autónomo (formación, simplificación de la interfaz);
- **desinterés** — camino que pide una decisión de atención, o una nueva concepción de la alerta;

- **caducidad rápida** — camino que no pide ninguna respuesta: *no todos los datos no leídos son fracasos*. Un dato de un ejercicio cerrado ve cerrarse su ventana de dirección: es su caducidad normal para ese uso, y sigue siendo válido para el balance y el archivo.

Esta distinción evita el error más costoso: tratar como un fracaso un dato que no estaba destinado a ser leído, o dejar pasar como normal una verdadera no lectura.

La firma de la obra

En cada entrega —un canal construido, un dato consolidado, un diagnóstico—, el canalero **firma**. La firma no es una vanidad: es un acto de responsabilidad (el artesano responde del trabajo entregado) y de transparencia (el usuario puede juzgar). Se coloca al principio de toda entrega, según el modelo probado por la comunidad del oficio.

FIRMA DE OBRA

- Artesano: [nombre]
- Fecha de entrega: [fecha]
- Tipo de obra: [canal mantenido / dato consolidado / diagnóstico]
- Pregunta de uso: [la pregunta a la que responde la obra, en una frase]
- Criterios principales:
 - Perímetro: [...]
 - Cadencia: [...]
 - Fidelidad buscada: [...]
 - Corpus movilizado: [...]
- Óptimo observacional: [justificación del arbitraje adoptado]
- Transporte / fidelidad esperada en la recepción:
 - [lo que el destinatario debe poder hacer para que su parte sea practicable; alteraciones posibles durante el trayecto]
- Ventana de lector buscada: [para quién es legible, en qué condiciones]
- Límites conocidos: [...]
- Validez estimada: [...]
- Condiciones de mantenimiento / nueva toma: [...]

Dos campos merecen atención, porque anclan la firma en la circulación y la recepción. **Transporte / fidelidad esperada** recuerda que firmar una obra es volver practicable la parte del lector —no solo depositar el dato—. **Ventana de lector buscada** recuerda que una obra está hecha para alguien situado, y que decirlo forma parte de la entrega. Sin estos dos campos, se firmaría un dato como si no tuviera que ser recibido.

Documentación viva: el mantenimiento por el cliente

El mejor mantenimiento es el que el cliente puede hacer por sí mismo. La actitud de referente lo exige: **documentación viva** —cada herramienta va acompañada de una cartografía que el cliente comprende, de modo que sea redactor jefe de sus datos, no su inquilino—. Una obra entregada sin su documentación es una obra que vuelve dependiente al cliente; una obra documentada es una obra que él puede mantener, hacer evolucionar, o confiar a otro. Es el **criterio verificable** del oficio, aplicado hasta la

última etapa: en todo momento, el cliente retoma el control sin perder nada.

Capítulo 10 — El taller digital: el material y los programas

Un artesano sin taller no es un artesano: es un teórico de la artesanía. El taller es el lugar donde el gesto se hace, donde se sostiene la herramienta, donde la obra se muestra. Este capítulo describe el banco de trabajo del canalero digital: el **material** (hardware) y los **programas** (software) de 2026. Todo está fechado, y por tanto es revisable; lo que no lo está es el principio que guía cada compra —*según la necesidad justa, ni ciego ni con sobreingeniería*—.

El material

El oficio de canalero es de **baja intensidad de capital**: no pide ni máquina rara ni gran desembolso. Unos pocos miles de euros bastan para equipar un taller completo. Es una fuerza del oficio —uno se instala sin endeudarse—.

El puesto de trabajo

- **Una computadora portátil** correcta y móvil: es la herramienta central, que sirve para desarrollar, para desplazarse a casa del cliente, para hacer funcionar los modelos de IA ligeros. Un procesador reciente y una memoria RAM generosa cuentan más que una potencia gráfica extrema. La movilidad permite la **proximidad** —ir a casa del cliente, tomar sobre el terreno, formar presencialmente—.
- **Una o dos pantallas**: para sostener el gesto justo, leer el código y la obra lado a lado.
- **Un puesto ergonómico** (escritorio, sillón): el gesto de lectura en pantalla es largo; el cuerpo del artesano también es una herramienta.

El servidor

Según las necesidades del cliente y del taller, un **servidor** local adopta varias formas, a veces acumuladas en una misma máquina pequeña:

- **Servidor web** — para alojar las aplicaciones de dirección, en local en casa del cliente o en una pequeña máquina de taller.
- **Servidor de base de datos** — cuando el volumen o la puesta en común superan lo que un simple archivo local puede sostener.
- **Servidor LLM local** — una máquina que sirve los modelos de IA internamente, para que el dato no salga de la empresa. Un servidor modesto basta para modelos ligeros; una tarjeta gráfica correcta acelera los modelos más capaces.

La **arquitectura edge / local-first** se declina en un continuo: desde el simple terminal del cliente (donde todo cabe en su equipo) hasta un pequeño servidor de taller compartido entre varios clientes de un mismo territorio. Se dimensiona lo más cerca posible de la necesidad real.

El almacenamiento y la copia de seguridad

- **Un servidor NAS** o un almacenamiento en red: para centralizar las obras del taller y hacer copias de seguridad de ellas.
- **Uno o varios discos externos cifrados**: para la regla **3-2-1** (tres copias, dos soportes, una fuera del sitio). Es la garantía contra la desaparición del dato.

Los periféricos

Una impresora (a menudo profesional, para los entregables y la documentación), un escáner si se digitalizan documentos existentes. Modestos, pero útiles para la circulación y el archivo.

Presupuesto tipo de instalación. La experiencia de las empresas piloto sitúa el equipamiento inicial completo (portátil, pantalla, disco externo, pequeño servidor/NAS, periféricos, puesto ergonómico) en una horquilla de unos pocos miles de euros —un perfil financiero robusto y poco arriesgado, a menudo aligerado por las ayudas a la creación de empresas—.

Los programas

El software del canalero es, en la medida de lo posible, **abierto, gratuito y perenne** —coherente con la soberanía que vende a sus clientes—.

El entorno de desarrollo

- **Un sistema operativo libre**: una distribución Linux (por ejemplo Debian) ofrece estabilidad, gratuidad y dominio. Equipa tanto el puesto de trabajo como los servidores.
- **Un editor de código (IDE)**: la herramienta donde se escribe la obra. Se elige uno que admita el trabajo con múltiples archivos de

la arquitectura hexagonal, el control de versiones, y la asistencia por IA si se desea.

- **Un gestor de versiones (Git)** y un alojamiento de repositorios: para guardar la historia del código, volver atrás, y conservar constancia de las evoluciones. Git es al taller lo que el registro inmutable es al dato: la memoria fiable del trabajo.

La contenedorización

- **Docker y Docker Compose:** para empaquetar una aplicación y sus servicios (base, caché, cola) en **contenedores** reproducibles. El beneficio es decisivo para el canalero: lo que funciona en su puesto de trabajo funciona de forma idéntica en casa del cliente, y los **entornos** se separan limpiamente (Desarrollo, Prueba, *Staging*, Producción). Se entrega un ensamblaje que se instala y se reinstala sin sorpresas.

El lenguaje y los servicios

- **JavaScript / Node.js** como lenguaje principal: un solo lenguaje desde el núcleo de negocio hasta los scripts de extracción y los servicios, lo que simplifica el mantenimiento para un taller de una persona.
- **Bases de datos abiertas:** *SQLite* (un archivo, sin servidor, ideal para la sobriedad) o *PostgreSQL* (servidor, para el volumen y la puesta en común).
- **Servicios internos** si hace falta, y solo si hace falta: una caché rápida (Redis), una cola de mensajes (RabbitMQ), un servicio de autenticación (OAuth2), una agregación de *logs*. Estos bloques se conectan como **adaptadores**; solo se añaden cuando se supera realmente un umbral de complejidad.

La IA local

- **Un entorno de ejecución de modelos locales** (de tipo *Ollama*) y algunos **modelos abiertos ligeros** (familias Llama, Phi, Gemma...), elegidos según la necesidad justa de cada tarea (clasificar, extraer, redactar, resumir). Véase el capítulo 8 para su empleo.

La documentación y la producción de entregables

- **Markdown** para escribir la documentación viva y los entregables textuales; **pandoc** para producir con ellos PDF limpios. Texto plano, duradero, versionable —que no depende de ningún procesador de texto propietario—.

El taller como lugar, físico o virtual

Más allá de las máquinas, el taller es un **lugar** dotado de funciones que las herramientas solas no dan:

- **un umbral** — un espacio identificado donde se entra para *hacer* el dato (fuera, se consume);
- **un espacio de trabajo** donde dejar los propios instrumentos y las obras en curso;
- **una biblioteca** — la memoria del taller: libros de referencia, rejillas de anotación conservadas, archivos de los flujos pasados, documentación de las obras entregadas;
- **un muro de la obra expuesta** — para honrar el trabajo, enseñar con el ejemplo, dar puntos de referencia.

Cuando el taller es **virtual** (espacio de trabajo en línea, repositorio compartido), debe reproducir esas **funciones**, no solo ordenar archivos: un umbral de acceso identificado, zonas personales, un espacio de revisión, archivos organizados, una galería de las obras. Sin eso, ya no es un taller: es una carpeta compartida.

La articulación taller / cadena

En toda organización, el taller coexiste con la **cadena** —el tratamiento masivo, a bajo costo, de los flujos estandarizados—. Los dos son necesarios y no sirven a las mismas necesidades. La cadena trata los volúmenes corrientes; el taller hace **a mano** los flujos que lo merecen: los que guían decisiones en las que hay mucho en juego, los que deben durar, aquellos cuya trazabilidad cuenta, los que hay que volver legibles para un destinatario preciso. El paso de uno a otro se decide explícitamente. El canalero sabe cuándo sacar el banco de trabajo, y cuándo dejar hacer a la cadena. Es una vez más un arbitraje en el óptimo —el último, y el más estratégico—.

Capítulo 11 — Itinerario de equipamiento: ensamblar las herramientas

Los capítulos anteriores describieron las herramientas una a una. Este las ensambla. Porque una herramienta solo vale en un gesto, y un gesto solo en un itinerario. Se muestra aquí cómo los métodos (hexagonal, gamificación, núcleo nativo, IA local) y las seis familias de herramientas se encadenan al servicio de un emprendedor en solitario —desde el primer contacto hasta la obra mantenida—. Todo es aquí ilustración: otro cliente pedirá otro ensamblaje.

El procedimiento, siempre el mismo: tres gestos

Sea cual sea el cliente, el canalero sigue tres gestos, en orden.

1. **Modelizar antes de programar.** Hablar el oficio del cliente, distinguir su **corazón del oficio** (intocable) de su **matriz de apoyo** (automatizable), y modelizar la matriz para liberar el corazón del oficio.
2. **Tomar según la necesidad justa.** Situar el óptimo entre subtoma y sobreingeniería, y volver a plantearlo en cada misión.
3. **Entregar una obra legible y reversible.** Documentada, comprendida por el cliente, firmada, y sobre la que él puede retomar el control en todo momento.

El desarrollo procede por etapas, nunca de una vez: **inmersión** (instalar la base vital), **ajuste** (el cliente se familiariza e identifica sus verdaderas necesidades), **expansión** (se añaden módulos si la necesidad lo justifica).

El itinerario tipo, etapa por etapa

Etapa 0 — La observación

Una fase de **auditoría** gratuita o ligera: conversar con el dirigente sobre su funcionamiento, sus expectativas, sus problemas, sus oportunidades.

Cartografiar su proceso de negocio (etapas, tareas, secuencias). Localizar las tareas que consumen tiempo, las penosas, las fuentes de errores. Entregable: un panorama de la situación y un plan de acción priorizado. *No se propone nada antes de haber escuchado.*

Eta­pa 1 — La esquematización (método hexagonal)

Esquematizar el sistema de gestión de los datos según la **arquitectura hexagonal**: aislar el núcleo de negocio, nombrar los puertos, identificar los adaptadores (almacenamiento, interfaz, servicios de terceros, sensores). El cliente obtiene una visión clara de lo que se va a construir, y puede evaluar sus plazos, su presupuesto, sus ganancias. Es aquí donde se decide la perennidad.

Eta­pa 2 — La lógica de negocio (en lenguaje claro, y luego en código)

Redactar las **entidades, reglas y casos de uso** en lenguaje claro (capítulo 2), hacerlas validar por el cliente, y luego implementarlas —en **JavaScript nativo** (capítulo 3)—. El núcleo puede probarse fuera de toda tecnología.

Eta­pa 3 — La toma y la fijación

Instalar las **herramientas de toma** según la necesidad justa (formularios móviles, scripts de extracción, sensores, extracción semántica por IA local) y las **rejillas de anotación** (capítulo 5). Elegir el **soporte de fijación** abierto y duradero (capítulo 6): a menudo un almacenamiento **local-first** en el equipo del cliente, con cifrado.

Eta­pa 4 — La dirección (lectura + gamificación)

Construir el **cuadro de mando** (capítulo 7): tres preguntas —*¿dónde estoy?, ¿qué debo hacer?, ¿qué se está desviando?*—, indicadores elegidos con el cliente, **umbrales** (fiscales, de negocio, objetivos). Revestirlo de una **gamificación** honesta (capítulo 4) para que se lea y sea agradable: barras de progreso, alertas exactas, anticipación de los umbrales. El cliente pasa de una actividad padecida a una trayectoria dirigida.

Eta­pa 5 — La circulación remota (si hace falta)

Si el cliente debe compartir (gestor contable, socio, agente de IA), añadir los **adaptadores de transporte**: exportaciones CSV/PDF consolidadas, o una **API (según el estándar MCP)** para los agentes de IA —sin tocar el núcleo—. La parte del lector destinatario debe ser practicable.

Etapa 6 — El mantenimiento y la autonomía

Instalar el **mantenimiento** (alertas de frescura, copia de seguridad 3-2-1, nuevas tomas), instaurar los **rituales** (revisión del canal, revisión de la recepción), **firmar** la obra, y —sobre todo— entregar la **documentación viva** que vuelve autónomo al cliente. La misión recurrente no es una dependencia mantenida: es el herrero al que se vuelve a llamar para una pieza nueva.

Tres ensamblajes ilustrados

A. El artesano de la construcción que debe pasar a la facturación certificada

Necesidad: un pintor, un albañil acostumbrados a una hoja de cálculo deben pasar a una solución de facturación conforme, sin saber elegir. *Ensamblaje:* poco desarrollo, mucho acompañamiento. Se audita, se presentan las opciones, se instala y se configura, se **forma** —idealmente en **formato compartido** (un pequeño grupo comparte la necesidad, intercambia, progresa en conjunto)—. La obra no es un programa: es una autonomía. Herramientas movilizadas: auditoría, rejilla de elección, formación.

B. El emprendedor en solitario móvil que dirige sus existencias y su tesorería

Necesidad: sustituir un cuaderno olvidado por una dirección al día. *Ensamblaje:* núcleo hexagonal en JS nativo; **PWA instalable** en el teléfono, que funciona sin conexión; almacenamiento **local-first** cifrado; entrada de datos fundida en el gesto (cada movimiento pone al día un indicador); cuadro de mando gamificado con anticipación del umbral de IVA; copia de seguridad 3-2-1; exportación PDF para el gestor contable. El artesano ya no «rellena una tabla»: dispone de un inventario y de una tesorería legibles de un vistazo.

C. La empresa que quiere indicadores a medida en un programa grande

Necesidad: un dirigente posee un programa costoso pero no saca de él sus indicadores. *Ensamblaje:* ninguna sustitución. El canalero escribe **consultas** sobre la base existente (indicadores derivados de sus datos), que el programa muestra. Eventualmente, una lógica de negocio de ayuda a la decisión (por ejemplo el rendimiento unitario de cada vehículo de una flota), redactada en lenguaje claro y luego implementada. Herramientas movilizadas: extracción, lógica de negocio hexagonal, lectura. Fabricar un dato, a veces, es saber consultar lo que ya está ahí.

El hilo que lo sostiene todo

A través de estos ensamblajes, un mismo hilo: **dar la capacidad, y luego la elección**. El canalero no vende un producto que encierra; transmite una capacidad que libera. Cada herramienta de este manual —desde el más humilde formulario hasta el modelo de IA local— se juzga con el mismo criterio: *¿vuelve al cliente más dueño de su actividad, o lo aleja de ella?* Mientras la respuesta sea la primera, la obra es justa.

El artesano solo no pesa nada en la economía del dato; la malla, sí. Cada emprendedor en solitario equipado, capacitado para leer y compartir su dato en su territorio, es un nudo de la malla. La potencia viene de su federación —pero empieza, siempre, por un canal bien cavado, en el óptimo, firmado, legible, y del que un cliente puede retomar el control—. Es el objeto de este manual; lo demás es asunto de práctica, y de la mano que aprende.

La federación no exige centro —solo exige un nudo más, cuyo oficio es coordinar—. El director de una construcción no manda sobre los calendarios de los gremios: sostiene la lectura más completa, establece las dependencias, y propaga mediante el intercambio lo que cada uno acepta en su propia obra. Coordinar mediante el intercambio cuesta menos gestos que la malla desnuda —y produce lo que la malla desnuda no produce: el orden coherente—. Es un oficio, no una autoridad; el día en que se convierte en una autoridad, la malla ha vuelto a ser una pirámide.

Capítulo 12 — El taller de agentes de IA: industrializar el itinerario

El capítulo anterior describió el itinerario de equipamiento tal como un canalero lo conduce a mano: observar, esquematizar, modelizar, tomar, dirigir, hacer circular, mantener. Este capítulo describe cómo **automatizar** ese mismo itinerario, sin cambiar nada de sus gestos, confiándolo a un equipo de modelos de IA especializados que se llamarán los **Agentes IA**. El canalero sabe gestionarlos: es su banco de trabajo, aumentado.

Como todo este manual, este capítulo es **forma**. Las herramientas que nombra —un agente de programación, un motor de contenedores, un servicio de alojamiento de repositorios— están fechadas. El fondo al que sirve —el itinerario, los tres gestos, la actitud de referente— no caduca.

La base técnica: no rehacer lo probado

Una parte de la obra no cambia nunca de una actividad a otra: la fontanería —servir archivos, funcionar sin conexión, ponerse al día, guardar el estado—. El canalero no la reescribe cada vez; la tiene por **base probada** y solo genera el dominio. Es la parte determinista del itinerario: cuanto menos genera el taller, menos se equivoca.

Sobre esta base, una regla simple: **una dependencia externa es una deuda y un riesgo**. Cuando una primitiva integrada y probada basta, se la prefiere —el cifrado mediante la interfaz nativa del navegador en lugar de una biblioteca de terceros cargada desde otro lugar, un servidor de unas pocas líneas en lugar de un paquete que instalar—. En lo que es sensible, esta preferencia no es negociable: una dependencia puede estar comprometida, y la obra debe poder funcionar sin instalar nada, sin red, y seguir siendo reversible. Un solo sistema de módulos, enlaces coherentes, pruebas que pasan antes de toda entrega: eso es lo que distingue una obra de la que el cliente retoma el control de un ensamblaje que se deshace.

El detalle aplicable —qué interfaces, qué formato, qué verificaciones— pertenece a la forma y vive fuera de este manual, en las restricciones

técnicas que leen los agentes. El principio, en cambio, no caduca.

Los Agentes IA

Un Agente IA es un modelo de IA sujeto a una **carta**: un papel, un artefacto que lee, un artefacto que produce, unas restricciones, y lo que debe ser verdad antes de que pase el relevo. A cada etapa del itinerario corresponde un agente. Trabajan en cadena: no un documento que se alarga, sino una **serie de documentos distintos**, versionados y legibles por un humano, cada uno de ellos la orden de trabajo del siguiente.

La singularidad del procedimiento cabe en una frase: **los documentos de aguas arriba no son simples especificaciones técnicas —son los capítulos del manual de calidad de la actividad—**. El manual de la actividad no se produce al lado del programa; es la parte de aguas arriba de la cadena. De ahí una frontera, operante aquí:

- por encima de la línea, la descripción de la actividad (actores, productos, procesos, trazabilidad, flujos, datos): es el manual, el **fondo**, el activo duradero;
- por debajo, el expediente hexagonal, el código, las pruebas: **forma fechada**, regenerable a partir del fondo.

El activo no es el programa. Es la especificación de la actividad.

Dos objetos que no hay que confundir: el taller de agentes de IA y la obra

El taller de agentes de IA es el taller de producción. Reúne, en la máquina local, el motor de modelos, los agentes, un entorno aislado de ejecución y de prueba, un repositorio. Se lo puede aislar en un contenedor (Docker): es la herramienta de producción, operada por el artesano.

La obra es el programa producido. Obedece a los capítulos 2 y 3: un núcleo nativo hexagonal, local-first, instalable, reversible, que el cliente puede retomar. **No exige el taller de agentes de IA para funcionar**. No se entrega un taller a un cliente: se le entrega un mueble del que tiene los planos. Confundir los dos —pedir al empresario que opere a diario la maquinaria que sirvió para fabricar— sería traicionar el gesto de reversibilidad.

El agente de programación: el banco de trabajo de los agentes

Los Agentes IA necesitan un entorno que les dé, bajo supervisión, acceso al sistema de archivos, al shell y a las herramientas: es el papel de un **agente de programación**. En 2026, se emplea para ello una herramienta como

Vibe Code: lee los archivos, ejecuta órdenes, escribe código, todo bajo el control del operador. Nada se escribe sin supervisión —lo que hace del hito de validación humana no una opción, sino la mecánica misma de la herramienta—. Esta herramienta es forma: será sustituida. El principio al que sirve —agentes que actúan sobre el banco de trabajo bajo la mirada del referente— permanece.

El repositorio: copia de seguridad, memoria y base distribuida

El trabajo se versiona en un repositorio **git**, por dos razones ya conocidas del capítulo 10: la **copia de seguridad** (la obra no depende de un solo disco) y la **memoria** (la evolución se relee, se compara, se deshace). A ello se añade una tercera función, propia del taller de agentes de IA: el repositorio sirve de **base distribuida**. El conjunto necesario para el arranque —el corpus como base de conocimiento, las cartas, las plantillas, el procedimiento de puesta en marcha— cabe en un repositorio que se clona o se descarga, y que se copia en la máquina local. El artesano dispone así de una base completa, lista para usar, que hace evolucionar para su territorio.

Los puertos soberanos: extender sin depender

El modelo local basta para la mayoría de las tareas, y es soberano por construcción. Allí donde no basta, el taller —como la obra— hace trabajar **servicios soberanos de alto rendimiento**, conectados por el hexágono (capítulos 3 y 6, «conectar el mundo exterior — siempre por el hexágono»). Se presentan tres usos:

- un **modelo de frontera soberano** para las tareas semánticas precisas fuera del alcance del modelo local;
- la **previsión** mediante modelos fundacionales para series temporales (demanda, existencias, carga);
- las **fuentes de datos interprofesionales**, para la vigilancia colectiva.

Regla absoluta, heredada de la soberanía distribuida: estos puertos **no son vitales**. El núcleo funciona cuando el vínculo exterior se rompe; el servicio soberano es un enriquecimiento, nunca una dependencia. Un adaptador, pues, y siempre sustituible.

La dirección del taller: lo determinista primero, el humano en los hitos

Un modelo local es más débil que un modelo de frontera para producir código fiable. El capítulo 8 dio la respuesta: **combinar la IA y las reglas deterministas**. El taller la aplica a la escala del programa entero. Es

mayoritariamente determinista —plantillas, andamiaje, generación acotada solo de lo que difiere de una base preconstruida y probada— y solo moviliza el modelo allí donde sobresale: describir el oficio, cartografiar, redactar. No es una autonomía total; es un itinerario asistido con puntos de control, en el que el referente valida antes de cada relevo.

La aplicación práctica

El detalle concreto —estructura de la carpeta, documentos que aportar, orden de lanzamiento de los agentes, órdenes— pertenece a una **guía de inicio** (*get-started*) viva, mantenida en el repositorio y revisada con él. Este manual no la reproduce: un modo de empleo que nombra órdenes y una dirección caducaría en pocas temporadas, y un libro de fondo no tiene por qué envejecer al ritmo de una herramienta. El manual dice el principio; el repositorio dice el gesto del día.

Lo que, en este capítulo, no caduca

El agente de programación, el motor de contenedores, el servicio de alojamiento de repositorios, los servicios soberanos nombrados: todo eso pasará. Lo que permanece es que un itinerario de equipamiento puede confiarse a agentes sujetos a cartas; que el manual de calidad de la actividad es su parte duradera y el programa su parte regenerable; que no se entrega el taller sino la obra; y que ningún servicio exterior debe convertirse nunca en vital. El canalero que sostiene esto sostendrá cualquier herramienta que el mañana le presente.

Anexo — La caja de herramientas del canalero

Este anexo reúne, como referencia rápida, las herramientas de interrogación y los puntos de referencia del manual. Una herramienta no es una regla: la regla dispensa de pensar, la herramienta abre una pregunta. Ninguna de estas fichas emite un veredicto automático —sirven para *no olvidar nada esencial*, nunca para sustituir el juicio situado—.

A. Las seis familias de herramientas, de un vistazo

Tiempo del dato	Familia	Herramientas digitales 2026
Fabricación	Toma	Formularios móviles, scripts de extracción, sensores/IoT, IA local (extracción semántica)
Fabricación	Anotación	Rejillas y clasificaciones, protocolos, anotación asistida por IA
Circulación	Fijación	Formatos abiertos (JSON, CSV, SQLite), inmutabilidad (UUID v7, hash), archivo cifrado
Circulación	Transporte	Local: consolidación, base, caché, colas. Remoto: exportación CSV/PDF, API (MCP)
Recepción	Lectura	Cuadro de mando nativo (HTML/CSS/JS), indicadores, visualizaciones sobrias, gamificación
Mantenimiento	Mantenimiento	Alertas de frescura/calibración, nueva toma, copia de seguridad 3-2-1, controles de coherencia

B. Los métodos transversales

- **Arquitectura hexagonal** — aislar el núcleo de negocio; puertos y adaptadores (*ports & adapters*); cambiar la forma sin tocar el fondo; modelizar antes de programar.
- **Núcleo sin dependencias** — HTML/CSS/JS nativo; SPA/PWA; local-first; perennidad, ligereza, reversibilidad.
- **Gamificación UX** — volver el dato consolidado legible y atractivo; umbrales; barras, insignias, alertas exactas; salvaguarda: recompensar el gesto justo, nunca la cifra en bruto.
- **IA local** — operador semántico; soberanía, costo, dominio; la IA propone, el humano decide; combinar con las reglas deterministas.

C. Lista de comprobación del flujo de observación encontrado

Para planteárselas uno mismo ante un conjunto de datos, un indicador, una cifra.

Fabricación — 1. ¿De qué acto nació este dato: toma sobre un flujo de lo real, acto que instituye bajo una regla, o cálculo a partir de otros datos? 2. ¿Quién lo cavó, con qué intención? 3. ¿Qué criterios —cadencia, perímetro, instrumento—? 4. ¿Qué unidades, normas, referencias implícitas? 5. ¿Qué es lo que no se ha tomado y podría ser pertinente? 6. ¿Cuál es la fecha de la inscripción, y su edad funcional? 7. ¿Está el flujo en un bucle —modifica lo que observa—? 8. ¿Qué sesgos engendra la clase de instrumento?

Circulación — 9. ¿Es trazable la cadena aguas arriba hasta la fuente? ¿Por cuántas manos, copias, nuevos tratamientos ha pasado —y qué ha podido sufrir en ellos—?

Recepción — 10. Para mi uso, ¿es adecuado este flujo? 11. ¿Tengo el medio y el tiempo de leerlo, con respecto a mi ventana? Si no, ¿es incapacidad, desinterés o caducidad rápida?

D. Rejilla de caducidad

Para cada dato y el uso que se hace de él, situar su temporalidad (desencadenante de pregunta, no veredicto): la caducidad se juzga para un uso. Un dato perecedero no es un mal dato: es un dato con una ventana de validez corta para ese uso, que puede seguir siendo válido para otro (balance, historial).

Dato	Fragmento de lo real	Velocidad de cambio	Edad funcional	Camino probable de no lectura
Dato demográfico	Población	Lenta	Varios años	—

Dato	Fragmento de lo real	Velocidad de cambio	Edad funcional	Camino probable de no lectura
Indicador económico	Mercado	Rápida	Días a semanas	Caducidad rápida
Medición ambiental	Aire, agua	Variable	Horas a meses	Según el uso
Puntuación de comportamiento	Persona	Rápida	Días	Caducidad rápida

E. Modelo de revisión del canal (orden del día)

1. Volumen y calidad de las tomas.
2. Calibración de los instrumentos: ¿deriva?
3. Pertinencia de los criterios con respecto al uso.
4. Corpus de referencia: ¿suficiente?
5. Bucles: ¿efectos observados?
6. Señales de caducidad: ¿qué refrescar?
7. Carga de mantenimiento: ¿sostenible?
8. Decisiones y arbitrajes: ¿quién responde?
9. Recepción efectiva: ¿leído, o no leído (incapacidad / desinterés / caducidad rápida)?

F. Ficha de firma de obra

Para colocar al principio de toda entrega (detalle en el capítulo 9).

FIRMA DE OBRA

- Artesano: [nombre]
- Fecha de entrega: [fecha]
- Tipo de obra: [canal mantenido / dato consolidado / diagnóstico]
- Pregunta de uso: [la pregunta a la que responde la obra, en una frase]
- Criterios principales:
 - Perímetro: [...]
 - Cadencia: [...]
 - Fidelidad buscada: [...]
 - Corpus movilizado: [...]
- Óptimo observacional: [justificación del arbitraje adoptado]
- Transporte / fidelidad esperada en la recepción:
 - [lo que el destinatario debe poder hacer para que su parte sea practicable; alteraciones posibles durante el trayecto]
- Ventana de lector buscada: [para quién es legible, en qué condiciones]
- Límites conocidos: [...]
- Validez estimada: [...]
- Condiciones de mantenimiento / nueva toma: [...]

G. Recordatorio de material y programas

Material — computadora portátil, pantalla(s), puesto ergonómico; servidor (web / base / LLM local); NAS y discos externos cifrados (copia de seguridad 3-2-1); impresora/escáner.

Programas — SO libre (Linux Debian); IDE + Git; Docker y Docker Compose (entornos Dev/Test/Staging/Prod); Node.js/JavaScript; SQLite o PostgreSQL; si hace falta, Redis, RabbitMQ, OAuth2; Ollama + modelos abiertos (Llama/Phi/Gemma); Markdown + pandoc.

H. Las salvaguardas, en siete frases

- 1. El óptimo vuelve a plantearse en cada misión: ni ciego, ni anegado.
- 2. El núcleo de negocio no depende de ninguna tecnología; la forma es sustituible, el fondo no lo es.
- 3. Tratar lo más cerca posible de la fuente; no hacer viajar más que lo que debe viajar.
- 4. La IA propone, el humano decide; la validación sigue siendo humana en todo lo que es crítico.
- 5. Gamificar el gesto justo, nunca la cifra en bruto; no maquillar una proyección como medida.
- 6. La herramienta mantiene la capacidad de leer del cliente, nunca lo dispensa de ella.
- 7. En todo momento, el cliente puede retomar el control sin perder nada.

I. Tabla de correspondencias — el vocabulario del corpus y los marcos existentes (forma fechada, julio de 2026)

El fondo no debe nada a los estándares del momento; el profesional, en cambio, se los encontrará. Esta tabla, revisable y fechada, da las equivalencias de trabajo —sin identidad de sentido—.

Corpus	Marco existente	Lo que coincide / lo que difiere
Cadena de trazabilidad; acto de inscripción atribuido	Procedencia / linaje (<i>lineage</i> , W3C PROV)	Misma preocupación por remontar la cadena; PROV modeliza entidades y actividades, el corpus parte del acto y de su sujeto.
Fijación inmutable; paso de estado escrito antes de la retirada	<i>Event sourcing; logs append-only</i>	Misma primacía del evento inscrito sobre el estado corriente; el corpus añade la caducidad de estado regulada por el uso.
Parte del lector practicable; corpus de referencia adjunto	Principios FAIR (<i>findable, accessible, interoperable, reusable</i>)	FAIR equipa la circulación; el corpus recuerda que la recepción no se decreta —un dato FAIR puede seguir mudo para tal lector, o simplemente sin leer—.
Frescura; plazo límite de uso	TTL, <i>data freshness / staleness</i>	Misma mecánica de umbral; el corpus funda el umbral en el uso del lector, no en la técnica.

Corpus	Marco existente	Lo que coincide / lo que difiere
Lo convenido; fuentes estables mantenidas	Referenciales, vocabularios controlados, <i>master data</i>	Misma función de acuerdo compartido; el corpus insiste: mantenido, luego perecedero.

Decisiones de traducción

Estas notas dicen cómo leer las palabras españolas que traducen los términos del corpus; las fija para toda la traducción española la *Carta de traducción del corpus*.

- **toma / tomar** (*prélèvement, prélever*): el gesto de tomar del flujo, como se dice una toma de agua; *tomar* no se emplea en sus usos corrientes. **Subtoma** y **nueva toma** (*sous-prélèvement, re-prélèvement*) lo siguen. **Registro** traduce el *relevé*, el valor inscrito; **entrada de datos**, la *saisie*, el gesto de introducir un dato.
- **canal; canadero; banco de trabajo** (*canal; canaillerie; établi*): el canal cavado en el río; el canadero es el artesano del dato (un oficio por crear) que lo cava y lo mantiene. El **taller** hace a mano lo que la **cadena** trata en masa.
- **mantener; mantenimiento** (*entretenir; entretien, maintenance*): el cuidado que conserva vivo un canal, un flujo o lo convenido, y la familia de herramientas que lo asegura.
- **caducidad; caducar; vencido; caducado** (*péremption; se périmer; périmée; péremptée*): una relación, no un deterioro; un dato caduca para un uso. La **caducidad de estado** y el **plazo límite de uso** (*durée limite d'utilisation*) nombran el umbral más allá del cual la brecha con lo real excede lo que el uso tolera; la **ventana de validez** es el tiempo anterior a él.
- **sustentante / sustentado** (*portante / portée*): un dato por el que otros solamente existen, y esos datos; el sustentado **desciende en racimo** (*descend en grappe*) con su sustentante. El vínculo de sustentación (el campo padre) es distinto de la **referencia** (el campo references).
- **armazón; lo convenido; fuente estable mantenida** (*armature; le convenu; source stable entretenue*): el cauce del canal, tomado de las convenciones compartidas, frente al agua de las tomas operativas. **Salvaguarda sobre la forma** (*Garde-fou de forme*) es la advertencia de que los medios descritos están fechados.
- **la reserva** (*la réserve*): el doble congelado retirado del ciclo de uso, distinto de la copia de seguridad 3-2-1.
- **momento objetual** (*moment objectal*): el tiempo durante el cual un dato consolidado se maneja como un objeto; **ontología objetual**, **residuo de proceso** e **inapropiabilidad del flujo** siguen el tomo 1. El consolidado traduce *la consolidée*.
- **restituir; verosímil** (*restituer; vraisemblable*): el narrador

restituye lo que el acervo encierra; un **dato verosímil** es inferido o estimado, frente a un dato observado.

- **huella; constancia** (*trace*): *huella* queda reservada a la marca que lo real deja de sí mismo; *constancia* traduce el sentido corriente («dejar constancia de quién ha hecho qué»); **trazabilidad** traduce *traçabilité*.
- **camino; itinerario; vía** (*chemin; parcours; voie*): *camino* se reserva para los tres caminos de la no lectura; **itinerario** traduce *parcours* (el **itinerario de equipamiento**); **trayecto**, el *chemin* del transporte. Las tres etapas del desarrollo son **inmersión, ajuste, expansión**.
- **núcleo de negocio; corazón del oficio; gesto de oficio** (*cœur métier; cœur de métier; geste métier*): el núcleo de negocio es el centro del hexágono; el corazón del oficio es el saber hacer del cliente, que la informática no toca. **Matriz de apoyo** y **matriz de oficio** traducen *matrice de soutien* y *matrice de métier*; **regla de negocio** y **lógica de negocio**, *règle métier* y *logique métier*. El **puesto** traduce *le poste*, el servidor soberano del dato («puesto requerido»; «de un solo puesto»); el **puesto de trabajo** conserva el sentido corriente. El **diario** traduce el *journal* de los actos (los *logs* de los marcos técnicos conservan su nombre); **reproducir, rejouer**.
- **emprendedor en solitario** traduce *solopreneur*; **empresario individual**, *entrepreneur individuel*. **Presupuesto** traduce *devis*; **recordatorio, relance**; **cuadro de mando de dirección**, *tableau de bord de pilotage*; **dirección**, el *pilotage* de la actividad.
- **malla; nudo** (*maillage; maille*): la red de los artesanos equipados de un territorio, y cada uno de ellos.
- **Agentes IA** (con mayúscula) es el nombre del equipo de agentes; **taller de agentes de IA** traduce *atelier d'agents IA*; las **cartas** son sus *chartes*; **agente de programación**, el *agent de codage*; **guía de inicio** (*get-started*), la guía viva del repositorio.
- **Código y ejemplos**. El pseudocódigo, los nombres de campos (rol, nombre, padre, valor, destino; references y label se conservan como en francés), los comentarios de la arborescencia y las etiquetas del esquema hexagonal se dan en español, como se escribirían en un taller de lengua española; los nombres de eventos siguen la forma función.entidad.acción. Las reglas de gestión *RG1-RG4* conservan su sigla, que es la misma en español. Los identificadores franceses sin equivalente (el número SIRET) se conservan. Allí donde el texto francés dice que la lógica de negocio se escribe «en francés», la traducción dice «en lengua natural».
- **Citas**. Heráclito se da en la forma española recibida («Nadie se baña dos veces en el mismo río»). Las fórmulas de la trilogía siguen su traducción española («Ninguna lectura es de ninguna parte», «Sin lectura, nada se actualiza», «No todos los datos no leídos son fracasos»).
- **Títulos**. *A prueba del flujo* traduce *À l'épreuve du flux*; este volumen es *Datos técnicos*. El *Cuaderno del taller* es el *Cuaderno del taller del artesano del dato*. Las herramientas del anexo conservan los títulos

fijados en *El canal y los talleres* (*Lista de comprobación del flujo de observación encontrado, Rejilla de caducidad, Modelo de revisión del canal, Ficha de firma de obra*).

Toda teoría gana con la práctica

Para el canalero que quiere surtir su banco de trabajo, para el aprendiz que aprende los gestos, para el emprendedor en solitario que quiere comprender lo que hace su herramienta —y por qué—.

Nadie se baña dos veces en el mismo río. Pero se puede aprender a cavar canales en él. Tras la trilogía *A prueba del flujo*, que decía el fondo, *Datos técnicos* da la forma: el **manual técnico del canalero**.

Herramientas y métodos para cavar, hoy, un canal digital —construir lo duradero, fabricar el dato, hacerlo circular, volverlo legible y mantenerlo—. Al servicio de un solo objetivo: ofrecer al emprendedor en solitario una gestión fluida, legible, dominada, y la serenidad que la acompaña. No un programa del que ya no se puede salir, sino una capacidad que libera.

Edición de junio de 2026

Las herramientas cambian; el oficio permanece