

COMENZAR

*El taller de agentes de IA
Del repositorio a la obra, paso a paso*

David Bories

A prueba del flujo — Una ontología procesual del dato, puesta en práctica
Comenzar — El taller de agentes de IA — Del repositorio a la obra, paso a paso

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.

Julio de 2026.

Versión 0.3.0-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>

contact@artisanat-de-la-donnee.fr

Índice

A quién se dirige este documento	3
0. La regla de arbitraje de las fuentes (para el agente)	5
1. Requisitos previos	7
2. La estructura de la carpeta	9
3. Lo que aporta el usuario	11
4. Poner en marcha el taller, agente por agente	13
5. Validar en cada punto de control	15
6. Recuperar e instalar	17
7. Hacer evolucionar	19
En caso de necesitar ayuda	21
Decisiones de traducción	21

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.

A quién se dirige este documento

Al **artesano del dato** (un oficio por crear) que acompaña a un emprendedor, o al emprendedor mismo si tiene las competencias. Describe cómo poner en marcha el **taller de agentes de IA** —para producir, a partir de un enunciado de actividad, el **manual de actividad** y la **obra de software** que es su versión activa—.

Este documento es deliberadamente **fechado y concreto**: nombra herramientas, comandos, una estructura de carpetas. Vive en el repositorio y se revisa con él. La doctrina, en cambio, está en el manual *Datos técnicos*, capítulo 12.

0. La regla de arbitraje de las fuentes (para el agente)

Antes de cualquier generación, el agente tiene el corpus por su jurisdicción, según cinco reglas de precedencia. **Uno.** Para el **fondo** —lo que es verdad del dato—, la trilogía hace fe; en caso de tensión aparente, el tomo prevalece sobre todo volumen de continuidad y sobre la síntesis. **Dos.** Para la **realización** —cómo hacer—, los volúmenes de continuidad hacen fe (*Datos técnicos*, el SGQA); su versión más reciente arbitra. **Tres.** La **síntesis** resume y no sustituye: no puede citarse contra un tomo. **Cuatro.** El **léxico canónico** zanja las definiciones; un término ausente del léxico se señala como fuera del corpus, nunca se improvisa. **Cinco.** Los **ejemplos fechados** (cifras, herramientas, sectores) no comprometen nada: ilustran la forma, nunca el fondo. Toda respuesta del agente debe poder designar su fuente y la regla que la hace prevalecer.

1. Requisitos previos

- **El taller digital** (*Datos técnicos*, capítulo 10): una máquina capaz de ejecutar un modelo de IA local.
 - **Un agente de programación** instalado: Vibe Code (Vibe CLI) — acceso supervisado al sistema de archivos, al shell y a herramientas; lee, ejecuta, escribe código—.
 - **git** instalado, para versionar la obra (copia de seguridad y memoria de la evolución).
 - **La base**: el repositorio Artisanat-de-la-donnee (URL por confirmar: <https://github.com/Dawebxx/Artisanat-de-la-donnee>), clonado o descargado en zip y copiado en la máquina local.
-

2. La estructura de la carpeta

Una vez recuperada la base, la carpeta de trabajo se presenta así:

```
artisanat-de-la-donnee/
├── README.md
├── commencer.md
├── corpus/
│   ├── tome_1_le_fleuve_et_le_canal.md
│   ├── tome_2_de_la_source_au_torrent.md
│   ├── tome_3_la_trace_ou_la_donnee.md
│   ├── artisan_de_la_donnee.md
│   ├── systeme_gestion_qualite_activite.md
│   ├── donnees_techniques.md
│   └── ... (el conjunto de los volúmenes)
├── atelier/
│   ├── plan-developpement-atelier-agents.md
│   ├── chartes-agents.md
│   └── secteurs/
│       ├── secteur-bijouterie.md
│       └── gabarits/
├── mon-activite/
│   ├── enonce.md
│   ├── manuel/
│   └── oeuvre/
```

← este documento

← la base de conocimiento (el fondo)

← los prompts de los Agentes IA

← documentos de sector (punto de partida)

← plantillas de artefactos vacías

← el espacio de trabajo del emprendedor

← POR RELLENAR

← producido por la capa de fondo

← producida por la capa de forma (PWA)

La carpeta corpus/ y la carpeta atelier/ son la base de conocimiento y las instrucciones; mon-activite/ es la única carpeta que uno rellena y que se va llenando.

3. Lo que aporta el usuario

El usuario **no** redacta prompts: los prompts son las **cartas** (atelier/chartes-agent). Solo aporta dos cosas:

1. **El enunciado de su actividad** —en mon-activite/enonce.md—: lo que hace, lo que vende, a quién, con qué. En lenguaje libre, sin preocuparse por la forma.
2. **La elección de un sector** —un documento de atelier/secteurs/ que *inicia* la reflexión (por ejemplo secteur-bijouterie.md)—. Da el eje; el enunciado da la realidad singular.

Todo lo demás —la cartografía, el manual, la arquitectura, el código— lo producen los Agentes IA.

4. Poner en marcha el taller, agente por agente

Abrir Vibe Code en la raíz de la carpeta, y darle corpus/ y atelier/ como base de conocimiento. Se sigue después el itinerario de equipamiento (*Datos técnicos*, capítulo 11) **en orden**. Para cada agente, la instrucción es la misma: «*Desempeña el papel definido por su carta; lee el artefacto de entrada; produce el artefacto de salida.*»

Orden	Agente IA	Se le da	Escribe en
1	Descubrimiento	enonce.md + sector	manuel/brief-activite.
2	Modelador de actividad	brief-activite + corpus	manuel/modele-activite
3	Trazabilidad	modele-activite	manuel/tracabilite.md
4	Flujos y datos	modele-activite	manuel/flux-et-donnees
—	<i>(se ensambla el manual: los cuatro archivos anteriores)</i>		
5	Arquitecto hexagonal	el manual	oeuvre/dossier-hexagon
6	Ingeniero de lógica de negocio	dossier-hexagonal	oeuvre/ (núcleo + pruebas)
7	Ingeniero de toma / fijación	modelo + expediente	oeuvre/ (captura, almacenamiento, correos)
8	Ingeniero de pilotaje	flux-et-donnees	oeuvre/ (cuadro de mando)
9	Ingeniero de circulación (<i>si hace falta</i>)	dossier-hexagonal	oeuvre/ (exportaciones, puerto de correo, API)

Orden	Agente IA	Se le da	Escribe en
10	Validador	la obra + el manual	oeuvre/rapport-validation documentación viva

El **Guardián del corpus** relee cada artefacto antes de pasar al siguiente (vocabulario, invariantes, salvaguarda de la facturación). El **Orquestador** —aquí, es usted— mantiene el orden y desencadena los puntos de control.

5. Validar en cada punto de control

Es ahí donde se juega la validación real, en la puesta en práctica. Antes de pasar de un agente al siguiente, el **referente** (el artesano o el emprendedor) lee el artefacto producido y lo confronta con la realidad de la actividad. Con Vibe Code, nada se escribe sin supervisión: se relee, se corrige, se vuelve a lanzar el agente si hace falta. El gesto se mantiene: *entregar una obra de la que el cliente pueda recuperar el control.*

6. Recuperar e instalar

Al final del itinerario:

- mon-activite/manuel/ contiene el **manual de actividad** (el fondo, duradero).
- mon-activite/oeuvre/ contiene la **obra**: un núcleo nativo, *local-first*, **instalable como PWA** en la computadora y el smartphone del emprendedor.

Se inicializa un repositorio git en mon-activite/: copia de seguridad y memoria de la evolución. Se instala la PWA. La actividad funciona.

7. Hacer evolucionar

El software no está congelado. Para un ajuste o una funcionalidad nueva, se vuelve a lanzar **solo el agente afectado** con el artefacto que hay que corregir, se valida, se versiona. Es el ciclo del itinerario: **inmersión → ajuste → expansión**.

Tres scripts en la raíz del repositorio equipan este ciclo, todos conectados al mismo puerto de generación (`lib-agent.sh`, que despacha hacia Vibe Code o Claude Code):

- **init.sh** — el itinerario completo: del enunciado al manual, luego a la obra, hasta la primera entrega.
- **evolve.sh** — una evolución (diseño, funcionalidad, corrección): en **rama beta**, validada en la puerta de pruebas, pasada a producción y **entregada** como versión.
- **debug.sh** — corregir un defecto **en la rama actual**, sin rama beta ni *release*. Sirve para el caso que las pruebas del núcleo no ven: un **cableado de interfaz** (un formulario que no desencadena nada). Su agente de depuración *reproduce* el defecto e inspecciona `adaptateurs/interface/app.js` (escuchadores de eventos, selectores) además del núcleo; no concluye «todo está en verde» sobre la sola fe de las pruebas. No se entrega ninguna versión; solo se escribe una constancia en `progress.md` si se pide.

En la **puerta de pruebas** común a estos scripts, cuando `node --test` falla, tres salidas: **d** deja que el agente corrija y vuelve al bucle; **c** prosigue el itinerario cuando uno mismo ha corregido («corregido manualmente»); **n** detiene.

En caso de necesitar ayuda

El emprendedor puede recorrer solo este itinerario si tiene las competencias, hacerse acompañar por el **artesano del dato**, o delegarle el conjunto. El artesano sabe gestionar los Agentes IA: es su oficio —el que el corpus llama a crear—.

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*.

- **artesano del dato (un oficio por crear)**: la precisión acompaña la primera aparición del término, porque el oficio todavía no existe; el final del documento lo recuerda.
- **itinerario** traduce el *parcours* del taller (el itinerario de equipamiento, el ciclo del itinerario); *camino* queda reservado a los tres caminos del tomo 3.
- **realización** traduce la *réalisation* de la segunda regla (cómo hacer); *puesta en práctica* traduce *mise en œuvre*.
- **obra** traduce *œuvre*, la obra de software que el taller entrega; **taller de agentes de IA**, *atelier d'agents IA*; **Agentes IA**, con mayúscula, es el nombre del equipo de agentes.
- **referente** es aquí la persona que valida en cada punto de control (el artesano o el emprendedor); **constancia** traduce la *trace* corriente que se deja en *progress.md*.
- **cartas** traduce las *chartes* de los agentes; **instrucciones**, las *consignes*.
- **inmersión** → **ajuste** → **expansión** traduce *imprégnation* → *ajustement* → *expansion*.
- Los nombres de carpetas, archivos, scripts y comandos (*mon-activite/*, *enonce.md*, *init.sh*, *node --test*) se conservan tal cual; solo se traducen los comentarios del árbol de carpetas.

Comenzar — El taller de agentes de IA

—

Del repositorio a la obra, paso a paso

—