IA local sin humo en Windows

```
$ ollama serve
→ conexión a ollama.com:443
$ $env:OLLAMA_NO_CLOUD=1
→ 0 conexiones en 20 s
```

VRAM · 16 GB



IA local sin humo en Windows

Un asistente documental en tu propio PC, con privacidad y rendimiento comprobados

Edición: 04/11/2026. **Audiencia:** profesionales que trabajan con documentos en Windows y quieren usar IA local sin comprar hardware a ciegas ni fiarse de promesas. **Estado de las prácticas:** prácticas ejecutadas y medidas en laboratorio el 04/11/2026 en el equipo descrito (RTX 5060 Ti 16 GB, 31 GB RAM).

Qué vas a conseguir

Montar un motor local (Ollama portable), trabajar con él desde interfaces cómodas (OpenCode y, para modelos grandes, Strata Terminal), saber qué sale a internet, medir velocidad y calidad con un corpus ficticio, y retirar Ollama sin dejar restos (la desinstalación verificada es la de Ollama; la de OpenCode y Strata no la comprobamos). Cada capítulo incluye bloques «Lo probamos» con lo que medimos, incluidos los fallos.

El laboratorio

- Windows 11 Pro, Intel Core Ultra 5 245K, 31,3 GB de RAM, NVIDIA RTX 5060 Ti con 16 GB de VRAM, NVMe de 1 TB.
- Ollama 0.40.14 (zip portable, SHA256 verificado), OpenCode 1.19.5, Strata Terminal 0.1.55.
- Modelos locales: qwen3.5:4b, qwen3.5:9b, gemma4:12b, qwen3.6:35b y embeddinggemma:300m.
- Referencia de nube con los mismos documentos ficticios: Claude Haiku 5.5, Claude Sonnet 5.5, GPT-6 luna, DeepSeek V4.1 Flash y DeepSeek V4.1 Flash 1017.

Las velocidades son de ese equipo y no se trasladan a otro. Cuando damos un resultado, indicamos cuántas veces se repitió.

Antes de empezar

- Usa una carpeta de laboratorio y el corpus ficticio del paquete, nunca documentos reales, hasta terminar las pruebas.
- Descarga solo de fuentes oficiales y verifica las huellas SHA256.
- No pongas claves ni credenciales en prompts, registros ni capturas.
- Lo que no hayas medido en tu equipo es «por medir».

Contenido

IA local sin humo en Windows . . . 2
Un asistente documental en tu propio PC, con privacidad y rendimiento comprobados . . . 2
Qué vas a conseguir . . . 2
El laboratorio . . . 2
Antes de empezar . . . 2
01 · Qué significa local y qué sale a internet 6
Las capas de un asistente local . . . 6
Cuatro afirmaciones distintas . . . 6
Lo probamos: Ollama al arrancar, con la configuración por defecto . . . 6
Los modelos «:cloud» no son locales . . . 7
Cómo comprobarlo en tu equipo . . . 7
Lo que esta prueba no demuestra . . . 7
La carpeta de laboratorio . . . 8
Mapa de localidad . . . 8
Práctica . . . 8
Resumen . . . 9
02 · Hardware: RAM, VRAM y disco sin comprar a ciegas 10
El equipo del laboratorio . . . 10
RAM y VRAM no son lo mismo . . . 10
Inventario en PowerShell . . . 10
Lo probamos: rendimiento con GPU . . . 11
Lo probamos: solo CPU . . . 11
Lo probamos: el modelo que no cabe . . . 12
El disco . . . 12
Señales de parada . . . 12
Decidir antes de comprar . . . 12
Práctica . . . 13
Resumen . . . 13
03 · Herramientas: el motor (Ollama) y las interfaces cómodas (OpenCode, Strata) 14
El motor: Ollama portable paso a paso . . . 14
Interfaz 1: OpenCode con un modelo local . . . 15
Interfaz 2: Strata Terminal para modelos grandes . . . 16
LM Studio, como alternativa con interfaz gráfica . . . 19
Qué elegir . . . 19
Práctica . . . 19
Resumen . . . 20

04 · Modelos, cuantización y contexto **21**

Identidad y procedencia . . . 21
Cuantización, en una frase . . . 21
El contexto: 4.096 por defecto . . . 21
El razonamiento que se come el contexto . . . 22
Salida estructurada: format: json . . . 23
Qué ajustar, en orden . . . 23
Práctica . . . 24
Resumen . . . 24

05 · Corpus de prueba y línea base **25**

Qué trae el paquete . . . 25
Lo probamos: quitar las pistas . . . 25
Dos líneas base, no una . . . 26
El corrector también es software . . . 26
Errores inducidos y qué control los caza . . . 27
Versiona el corpus . . . 27
Práctica . . . 27
Resumen . . . 27

06 · RAG con referencias: cuándo hace falta y cuándo no **28**

Tres formas de dar contexto al modelo . . . 28
Lo probamos: el corpus y las preguntas . . . 28
Lo probamos: resultados . . . 28
La trampa del contexto por defecto . . . 29
Conclusión honesta . . . 30
Búsqueda por palabras clave en PowerShell . . . 30
Referencias: el control que sí importa . . . 30
Práctica . . . 31
Resumen . . . 31

07 · Calidad en español: local frente a nube **32**

La batería . . . 32
Lo probamos: primera tanda, sin forzar el formato . . . 32
Lo probamos: segunda tanda, con format: json . . . 33
Las sumas se comprueban con código . . . 33
Lo probamos: OCR con modelos locales con visión . . . 33
Lo probamos: la receta con validación . . . 34
Español de trabajo . . . 35
Práctica . . . 35
Resumen . . . 35

08 · Medir el rendimiento sin engañarte **36**

Los tres tiempos . . . 36
Cómo medir con la API de Ollama . . . 36
Lo probamos: carga en frío y en caliente . . . 37
Lo probamos: la caché de prompt . . . 37
Lo probamos: no fiarse de un solo indicador . . . 37
Lo probamos: lo que hay abierto cambia todo . . . 38
Repeticiones . . . 38
Del tiempo de máquina al tiempo de trabajo . . . 38
Plantilla . . . 38
Práctica . . . 38
Resumen . . . 39

09 · Red, copias y actualizaciones 40

Política de red del laboratorio . . . 40
Observar las conexiones con PowerShell . . . 40
La carpeta de modelos propia . . . 41
Qué copiar y qué no . . . 41
Versiones fijadas . . . 42
Actualizar sin perder lo validado . . . 42
Registro de mantenimiento . . . 42
Práctica . . . 43
Resumen . . . 43

10 · Diagnóstico, desinstalación y cuándo usar la nube 44

Diagnóstico: lee el registro, no el mensaje . . . 44
Lo probamos: el 35B que no arranca . . . 44
Lo probamos: tres modelos cargados y un juego . . . 45
Tabla de síntomas: lo que vimos y su causa . . . 45
Desinstalación verificada . . . 45
Cuándo usar la nube . . . 46
Diseño híbrido . . . 46
Práctica . . . 47
Resumen . . . 47

01 · Qué significa local y qué sale a internet

«IA local» dice dónde se ejecuta el modelo, y nada más. No dice qué conexiones abre el programa al arrancar, qué descarga por su cuenta ni si el modelo que eliges se ejecuta de verdad en tu equipo. En este capítulo separamos esas preguntas y las respondemos con lo que medimos el 04/11/2026 en el equipo del laboratorio (Windows 11, RTX 5060 Ti 16 GB, 31 GB de RAM) con Ollama 0.40.14.

Las capas de un asistente local

Un asistente documental que «funciona en local» suele tener varias piezas, y cada una puede comportarse de forma distinta:

- **El motor**: el programa que carga el modelo y genera texto. En este libro, Ollama.
- **El modelo**: el archivo descargado (varios GB) que el motor ejecuta.
- **La interfaz**: lo que usas tú para hablar con el modelo. Puede ser la línea de comandos de Ollama, un agente como OpenCode o un lanzador como Strata Terminal (capítulo 03).
- **Tus documentos y salidas**: las carpetas donde guardas corpus, borradores y registros.
- **Servicios auxiliares**: descargas de modelos, recomendaciones, actualizaciones, inicio de sesión.

Que el motor sea local no convierte en locales a las demás piezas. Cada una se comprueba por separado.

Cuatro afirmaciones distintas

Sustituye «es privado» por afirmaciones que puedas comprobar:

1. **Inferencia local**: el modelo genera la respuesta en tu equipo.
2. **Documentos locales**: el corpus no sale de una carpeta que controlas.
3. **Red controlada**: durante una prueba acotada sabes qué conexiones se abrieron y hacia dónde.
4. **Salida limpia**: sabes qué deja el programa en el disco y cómo retirarlo (capítulo 10).

La primera no demuestra las otras tres. Lo vimos en el laboratorio.

Lo probamos: Ollama al arrancar, con la configuración por defecto

Arrancamos `ollama serve` (versión zip portable, sin instalador) y observamos durante 20 segundos las conexiones TCP del proceso y de sus procesos hijos con el script del capítulo 09. Una sola ejecución por configuración.

- **Con la configuración por defecto**, nada más arrancar Ollama abrió una conexión HTTPS a `34.36.133.15:443`. El DNS inverso devolvía un nombre de `googleusercontent.com`; es la dirección a la que resuelve ollama.com. No habíamos pedido nada al modelo.
- El registro del servidor lo explicaba: la línea `Ollama cloud disabled: false` y una tarea llamada `model recommendations cache` que descarga una lista de modelos «recomendados».
- Esa lista se guardó en `C:\Users\<usuario>\.ollama\cache\model-recommendations.json`. Junto a ella apareció una pareja de claves `id_ed25519` / `id_ed25519.pub`: la identidad del equipo ante ollama.com, que se usa para iniciar sesión o subir modelos.
- **Con** `OLLAMA_NO_CLOUD=1`, el registro cambió a `Ollama cloud disabled: true` y en los 20 segundos de observación no hubo ninguna conexión externa.
- **Durante una respuesta**, con la nube desactivada y qwen3.5:4b contestando, observamos 40 segundos: 0 conexiones externas.

Conclusión práctica: descarga los modelos con red y, para trabajar, arranca Ollama con `OLLAMA_NO_CLOUD=1`. Descargar (`ollama pull`) necesita internet; responder con un modelo ya descargado, no.

Los modelos «:cloud» no son locales

La lista de recomendaciones que Ollama descargó incluía modelos con el sufijo `:cloud`. Esos modelos se ejecutan en servidores de Ollama, no en tu tarjeta gráfica; uno de ellos venía marcado con `required_plan: pro`. El nombre aparece en el mismo catálogo y se usa con los mismos comandos, así que es fácil confundirlos.

Regla sencilla:

- Si el nombre del modelo termina en `:cloud`, tus textos salen de tu equipo.
- Si quieres que eso no pueda pasar por descuido, arranca con `OLLAMA_NO_CLOUD=1`.
- Antes de pegar un documento, mira qué modelo tienes seleccionado. En la interfaz, no en tu memoria.

Cómo comprobarlo en tu equipo

Necesitas Ollama arrancado desde PowerShell con su registro redirigido a un archivo (el capítulo 03 explica cómo). Después:

1. Busca en el registro la línea de la nube:

```
Select-String -Path C:\IA-LAB\ollama\serve.log -Pattern 'cloud disabled'
```

2. Si dice `false`, vuelve a arrancarlo con el script del capítulo 03 y la opción que desactiva la nube (el script para el servidor anterior y define la variable por ti):

```
C:\IA-LAB\ollama\iniciar.ps1 -SinNube
```

3. Comprueba que ahora dice `true`.
4. Mientras el modelo responde a una pregunta, ejecuta el script de observación de red del capítulo 09 y guarda el resultado.
5. Revisa qué hay en tu carpeta de usuario:

```
Get-ChildItem -Recurse -Force "$env:USERPROFILE\.ollama" | Select-Object FullName, Length, CreationTime
```

En nuestro equipo, tras usar el zip portable, solo había las dos claves y la caché de recomendaciones. El zip no creó servicio, ni tarea programada, ni entrada de inicio automático, ni reglas de firewall. El instalador normal de Ollama sí se instala como aplicación con arranque; no lo probamos, así que no podemos decirte qué deja.

Lo que esta prueba no demuestra

Medir es útil si sabes hasta dónde llega la medición:

- **Ventanas cortas.** Observamos 20 y 40 segundos, consultando las conexiones cada 300 milisegundos. Una conexión muy breve entre dos consultas podría no aparecer. Si necesitas más garantía, observa más tiempo y repite.
- **Solo el proceso de Ollama.** El script vigila Ollama y sus procesos hijos. La interfaz que uses encima (OpenCode, Strata u otra) es otro programa con sus propias conexiones. En el laboratorio miramos sus conexiones solo a ratos (unas cuantas consultas durante una prueba, no una vigilancia continua): el servidor de Strata solo escuchó en `127.0.0.1`; OpenCode abrió una conexión a un proveedor de nube por su `small_model` (capítulo 03).

- **Una versión concreta.** Esto es Ollama 0.40.14. El comportamiento por defecto puede cambiar en otra versión; repite la prueba al actualizar (capítulo 09).
- **Funcionar sin red no es lo mismo que no conectarse.** Si desconectas el cable y el modelo responde, sabes que ese caso funciona desconectado. No sabes qué intenta hacer el programa cuando sí tiene red.

La carpeta de laboratorio

Trabaja con una estructura separada y sin documentos reales hasta terminar las pruebas:

```
C:\IA-LAB\
  ollama\          motor (bin\), modelos (modelos\) y registros (serve.log)
  trabajo\
    corpus\        copia del corpus del paquete
    salidas\       respuestas guardadas
    evaluacion\    plantillas rellenas
```

Separar el motor de tu trabajo tiene una razón práctica: en el capítulo 10 borraremos `C:\IA-LAB\ollama` entero, y tus resultados no deben estar dentro. Evita carpetas sincronizadas con servicios en la nube mientras pruebas: la sincronización es otra conexión que sale de tu equipo.

Mapa de localidad

Rellena esta tabla para tu instalación. La primera fila es la que medimos; las demás, compruébalas tú:

Pieza	¿Dónde se ejecuta?	Conexiones observadas	Cómo se controla	Evidencia
Motor Ollama 0.40.14 (zip)	Local	Al arrancar, a ollama.com si la nube está activa	`OLLAMA_NO_CLOUD=1`	Registro + script de red
Modelo `nombre:etiqueta`	Local	0 durante una respuesta (nube desactivada)	No usar modelos `:cloud`	Script de red
Interfaz: servidor de Strata	Local	Ninguna vista durante el trabajo (muestreo puntual)	—	`Get-NetTCPConnection`
Interfaz: OpenCode	Local, pero…	Al `small_model` si es de nube; a npm en el primer arranque	`model` y `small_model` locales en el proyecto	`Get-NetTCPConnection`
Descargas de modelos	Red	Sí, por definición	Hacerlas antes de trabajar	—
Tus documentos	Tu carpeta	—	Carpeta fuera de sincronización	—

Práctica

Con el corpus del paquete (`corpus/DOC-01-acta-sintetica.md` y `corpus/preguntas.md`):

1. Copia la carpeta `corpus` del paquete a `C:\IA-LAB\trabajo\corpus`.
2. Arranca Ollama con la configuración por defecto (capítulo 03) y ejecuta el script de red durante 20 segundos. Anota destinos y la línea `cloud disabled`.
3. Para el servidor, arráncalo con `OLLAMA_NO_CLOUD=1` y repite la observación.
4. Con la nube desactivada, pregunta a un modelo ya descargado la Q-01 de `preguntas.md` pegando el texto de DOC-01. Observa la red mientras responde.
5. Registra el resultado en la fila `RED-01` de `plantillas/matriz-pruebas-local.csv`, con fecha, versión de Ollama y duración de la observación.

6. Lista lo que ha quedado en `%USERPROFILE%\.ollama`.

Criterios de aceptación:

- Sabes qué línea del registro indica si la nube está activa.
- Tienes dos observaciones de red (por defecto y sin nube) guardadas con fecha.
- Puedes decir qué modelo usaste y que su nombre no termina en `:cloud`.
- La tabla de localidad deja como «por medir» lo que no has observado.

Resumen

Ollama con la configuración por defecto se conectó a ollama.com nada más arrancar, sin que le pidiéramos nada; con `OLLAMA_NO_CLOUD=1` no hubo ninguna conexión externa ni al arrancar ni durante una respuesta. Los modelos con sufijo `:cloud` se ejecutan fuera de tu equipo. La medición vale para esa versión, ese proceso y esa ventana de tiempo: la interfaz que pongas encima se comprueba aparte.