IA local sin humo en Windows



IA local sin humo en Windows

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

Para quién: trabajas con documentos en Windows y quieres usar IA en tu propio PC sin comprar hardware a ciegas ni fiarte de promesas. Sabes abrir PowerShell y pegar un comando; no hace falta que programes.

Qué vas a conseguir

Al terminar tendrás un motor de IA funcionando en tu PC (Ollama, en su versión portable), sabrás trabajar con él desde herramientas cómodas, sabrás qué sale a internet y qué no, habrás medido la velocidad y la calidad con documentos de prueba y podrás retirarlo todo sin dejar restos.

Cada capítulo te guía paso a paso y te dice cómo comprobar que ha ido bien. Los recuadros «Lo probamos» cuentan qué pasó cuando lo hicimos nosotros, fallos incluidos, para que sepas qué esperar.

Cómo lo probamos

Lo probamos en un PC con Windows 11 y una gráfica NVIDIA de 16 GB. Estas son las herramientas y modelos que aparecen en el libro:

- Ollama (zip portable, con su huella SHA256 comprobada), OpenCode y Strata Terminal. Otras versiones pueden comportarse distinto: comprueba la tuya y repite las pruebas al actualizar (capítulo 09).
- Modelos en el PC: qwen3.5:4b, qwen3.5:9b, gemma4:12b, qwen3.6:35b y embeddinggemma:300m.
- Modelos de nube usados como referencia, 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 PC y no se trasladan al tuyo. Cuando damos un resultado, decimos cuántas veces se repitió.

Antes de empezar

- Usa una carpeta de pruebas y los documentos ficticios del paquete, nunca documentos reales, hasta terminar las pruebas.
- Descarga solo de fuentes oficiales y comprueba las huellas SHA256 (el capítulo 03 explica cómo).
- No pongas claves ni contraseñas en tus mensajes al modelo, en registros ni en capturas.
- Lo que no hayas medido en tu PC, apúntalo como «sin medir».

Contenido

01 · Qué significa local y qué sale a internet **6**

Las piezas de un asistente local . . . 6
Cuatro preguntas en lugar de «es privado» . . . 6
Paso a paso: comprueba qué sale de tu PC . . . 6
Los modelos «:cloud» no son locales . . . 7
Hasta dónde llega esta comprobación . . . 7
Prepara tu carpeta de pruebas . . . 8
Tu mapa de localidad . . . 8
Práctica . . . 8
Resumen . . . 8

02 · Hardware: RAM, VRAM y disco sin comprar a ciegas **9**

RAM y VRAM no son lo mismo . . . 9
Paso a paso: haz tu inventario . . . 9
Lo probamos: rendimiento con gráfica . . . 9
Lo probamos: solo con el procesador . . . 10
Lo probamos: el modelo que no cabe . . . 10
El disco . . . 11
Cuándo parar . . . 11
Antes de comprar . . . 11
Práctica . . . 11
Resumen . . . 12

03 · Herramientas: el motor (Ollama) y las interfaces cómodas (OpenCode, Strata) **13**

El motor: Ollama portable paso a paso . . . 13
Interfaz 1: OpenCode con un modelo local . . . 14
Interfaz 2: Strata Terminal para modelos grandes . . . 16
LM Studio, como alternativa con ventana . . . 18
Qué elegir . . . 18
Práctica . . . 18
Resumen . . . 19

04 · Modelos, cuantización y contexto **20**

Paso a paso: ficha de cada modelo . . . 20
Cuantización, en una frase . . . 20
El contexto: 4.096 por defecto . . . 20
El razonamiento que se come el contexto . . . 21
Salida estructurada: format: json . . . 22
Qué ajustar, en orden . . . 22

Práctica . . . 23
Resumen . . . 23

05 · Corpus de prueba y línea base 24

Qué trae el paquete . . . 24
Paso a paso: prepara tu copia de trabajo . . . 24
Dos líneas base, no una . . . 25
El corrector también es software . . . 25
Errores sembrados y qué control los caza . . . 26
Versiona el corpus . . . 26
Práctica . . . 26
Resumen . . . 26

06 · RAG con referencias: cuándo hace falta y cuándo no 27

Tres formas de dar contexto al modelo . . . 27
Paso a paso: empieza por lo sencillo . . . 27
La trampa del contexto por defecto . . . 28
Referencias: el control que sí importa . . . 28
Lo probamos: el corpus y las preguntas . . . 28
Lo probamos: resultados . . . 28
Cuándo sí esperamos que RAG aporte . . . 29
Práctica . . . 29
Resumen . . . 30

07 · Calidad en español: local frente a nube 31

La batería . . . 31
Paso a paso: las sumas se comprueban con código . . . 31
Paso a paso: documentos en imagen . . . 32
Lo probamos: primera tanda, sin forzar el formato . . . 32
Lo probamos: segunda tanda, con format: json . . . 33
Lo probamos: la receta con validación . . . 33
Español de trabajo . . . 34
Práctica . . . 34
Resumen . . . 34

08 · Medir el rendimiento sin engañarte 35

Los tres tiempos . . . 35
Qué devuelve la API de Ollama . . . 35
Paso a paso: mide . . . 35
La primera carga . . . 36
La caché de prompt . . . 36
No te fíes de un solo indicador . . . 36
Lo que hay abierto cambia todo . . . 37

Repeticiones . . . 37
Del tiempo de máquina al tiempo de trabajo . . . 37
Práctica . . . 37
Resumen . . . 38
09 · Red, copias y actualizaciones **39**
Paso 1: escribe tu política de red . . . 39
Paso 2: vigila las conexiones . . . 39
Paso 3: usa una carpeta de modelos propia . . . 40
Paso 4: copia tu trabajo, no los modelos . . . 40
Paso 5: restaura la copia de prueba . . . 40
Paso 6: fija las versiones . . . 41
Paso 7: actualiza sin perder lo validado . . . 41
Práctica . . . 41
Resumen . . . 42
10 · Diagnóstico, desinstalación y cuándo usar la nube **43**
Paso a paso: diagnostica leyendo el registro, no el mensaje . . . 43
Tabla de síntomas . . . 43
Si el modelo no carga . . . 43
Si va lento con la gráfica llena . . . 44
Paso a paso: desinstala y comprueba . . . 44
Cuándo usar la nube . . . 45
Diseño híbrido . . . 45
Práctica . . . 45
Resumen . . . 46

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 PC. Al acabar sabrás comprobarlo, habrás cortado la conexión que Ollama hace por defecto y tendrás tu carpeta de pruebas.

Las piezas de un asistente local

Un asistente documental que «funciona en local» tiene varias piezas, y cada una se comporta a su manera:

- **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 (un programa que usa el modelo para leer y escribir archivos por su cuenta) o Strata Terminal (capítulo 03).
- **Tus documentos y resultados**: las carpetas donde guardas documentos, 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 preguntas en lugar de «es privado»

Cambia «es privado» por cuatro afirmaciones que puedas comprobar:

1. **El modelo responde en tu PC.**
2. **Tus documentos no salen de una carpeta que controlas.**
3. **Sabes qué conexiones se abren**, y hacia dónde, durante una prueba concreta.
4. **Sabes qué deja el programa en el disco** y cómo retirarlo (capítulo 10).

La primera no demuestra las otras tres, como verás enseguida.

Paso a paso: comprueba qué sale de tu PC

Necesitas Ollama arrancado desde PowerShell con su registro guardado en un archivo (el capítulo 03 explica cómo). El registro es el archivo de texto donde Ollama apunta lo que hace.

1. Busca en el registro la línea de la nube. Cómo sabes que ha ido bien: sale una línea con `cloud disabled` seguida de `true` o `false`.

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

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

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

3. Busca otra vez la línea. Cómo sabes que ha ido bien: ahora dice `true`.
4. Mientras el modelo responde a una pregunta, ejecuta el vigilante de red del capítulo 09 y guarda el resultado. Cómo sabes que ha ido bien: termina con `0 conexiones externas`.
5. Revisa qué ha dejado Ollama en tu carpeta de usuario. Cómo sabes que ha ido bien: ves solo las dos claves `id_ed25519` y la caché de recomendaciones.

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

Descargar un modelo (`ollama pull`) necesita internet; responder con un modelo ya descargado, no. Descarga con red y, para trabajar, arranca Ollama con `OLLAMA_NO_CLOUD=1`.

Lo probamos: Ollama al arrancar

Arrancamos `ollama serve` (versión zip portable, sin instalador) y vigilamos 20 segundos las conexiones del proceso y de sus procesos hijos. Una ejecución por configuración.

- **Con la configuración por defecto**, nada más arrancar y sin pedirle nada, Ollama abrió una conexión cifrada a `34.36.133.15:443`. Su nombre inverso era de `googleusercontent.com`: es la dirección a la que resuelve ollama.com.
- El registro lo explicaba: la línea `Ollama cloud disabled: false` y una tarea llamada `model recommendations cache`, que descarga una lista de modelos «recomendados».
- La lista se guardó en `C:\Users\<usuario>\.ollama\cache\model-recommendations.json`. Junto a ella aparecieron dos claves, `id_ed25519` / `id_ed25519.pub`: la identidad de tu PC ante ollama.com, para iniciar sesión o subir modelos.
- **Con** `OLLAMA_NO_CLOUD=1`, el registro pasó a `Ollama cloud disabled: true` y en 20 segundos no hubo ninguna conexión externa.
- **Durante una respuesta**, con la nube cortada y qwen3.5:4b contestando, vigilamos 40 segundos: 0 conexiones externas.

El zip portable no creó servicio, ni tarea programada, ni arranque automático, ni reglas de firewall. El instalador normal de Ollama sí se instala como aplicación que arranca con Windows; no lo probamos, así que no sabemos qué deja.

Los modelos «:cloud» no son locales

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

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

Hasta dónde llega esta comprobación

- **Ventanas cortas.** Vigilamos 20 y 40 segundos, mirando las conexiones cada 300 milisegundos. Una conexión muy breve entre dos consultas puede escaparse. Si necesitas más garantía, vigila más tiempo y repite.
- **Solo el proceso de Ollama.** El vigilante mira Ollama y sus procesos hijos. La interfaz que uses encima (OpenCode, Strata u otra) es otro programa con sus propias conexiones. A esas las miramos solo a ratos: 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 vale para la versión de Ollama que probamos. Otra versión puede cambiar el comportamiento por defecto; 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 funciona desconectado. No sabes qué intenta hacer el programa cuando sí tiene red.

Prepara tu carpeta de pruebas

Trabaja en 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 borrarás `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 PC.

Tu 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 (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

1. Copia la carpeta `corpus` del paquete (los documentos ficticios con los que vas a probar) a `C:\IA-LAB\trabajo\corpus`. Cómo sabes que ha ido bien: ves `DOC-01-acta-sintetica.md` dentro.
2. Haz el paso a paso de arriba dos veces: con la configuración por defecto y con `OLLAMA_NO_CLOUD=1`. En la segunda, pregunta a un modelo ya descargado la Q-01 de `corpus/preguntas.md`, pegando el texto de DOC-01, mientras vigilas la red.
3. Apunta el resultado en la fila `RED-01` de `plantillas/matriz-pruebas-local.csv`, con fecha, versión de Ollama y duración de la vigilancia. Cómo sabes que ha ido bien: tienes dos vigilancias guardadas, sabes qué modelo usaste y que no termina en `:cloud`, y tu mapa de localidad deja como «sin medir» lo que no has observado.

Resumen

Con la configuración por defecto, Ollama se conectó a ollama.com nada más arrancar, sin pedirle 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 PC. La comprobación vale para esa versión, ese proceso y ese rato: la interfaz que pongas encima se comprueba aparte.