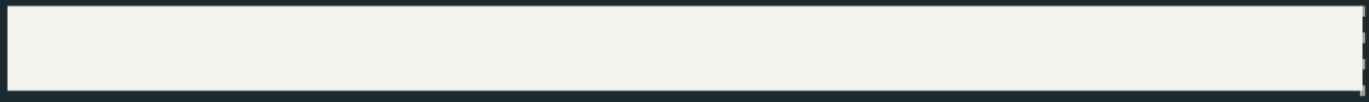
Pregunta a tus PDF sin nube

ENCARGO INGENUO 85/120



BRIEF OPERATIVO 120/120



Pregunta a tus PDF sin subirlos a la nube

IA local que responde con cita de archivo y página, comprueba sus propias citas y no saca nada de tu PC

Edición: 05/11/2026. **Para quién:** trabajas en una gestoría, un despacho, una consultoría o una administración, manejas PDF confidenciales (contratos, expedientes, escrituras, informes, normativa interna) y quieres hacerles preguntas y resúmenes con IA sin enviarlos a ningún servicio externo. Usas Windows y sabes instalar programas; no hace falta que programes: cada orden viene completa para copiar y pegar en PowerShell, y el script del paquete se usa sin tocarlo.

Alcance

Las cifras de este libro se midieron el 04/11/2026 en un solo equipo: Windows 11 Pro, Intel Core Ultra 5 245K, 31 GB de RAM y una NVIDIA RTX 5060 Ti de 16 GB, con Ollama 0.40.14 y documentos ficticios hechos para las pruebas. Las velocidades son de ese equipo; los métodos, los fallos y las comprobaciones te sirven en el tuyo. Lo que viene de documentación oficial se cita con su fuente, consultada el 05/11/2026; las versiones de los programas cambian, así que si una orden no responde igual, mira la documentación enlazada. Es el único aviso de este tipo que vas a leer.

Cómo usar este libro

Haz el capítulo 01 entero, de una sentada: en menos de una hora tendrás el sistema funcionando con tres PDF de ejemplo cuyas respuestas conoces, y sabrás distinguir una cita comprobada de una inventada. Antes de meter documentos de clientes, lee el capítulo 07. Lo demás, según lo que te pase:

Lo que te pasa o quieres	Capítulo
Algunas páginas salen vacías, el texto sale desordenado, tienes escaneados	02
Quieres resumir un documento entero o hacer preguntas sobre todo él	03
Tienes muchos documentos y preguntas concretas	04
Preguntas con otras palabras y no encuentra la página	05
Necesitas saber cuánto puedes fiarte antes de usarlo en serio	06
Te preguntan dónde están los datos y qué sale de tu equipo	07

Qué necesitas

- Windows 10 22H2 u 11 y unos 15 GB libres en disco.
- Mejor con una gráfica NVIDIA de 12 GB o más; con menos, o sin gráfica, funciona con un modelo más pequeño y más despacio.
- Conexión a internet solo para instalar y descargar el modelo. Después, nada.
- El paquete del libro: `scripts\` (el script `preguntar_pdf.py` y `comprobar-red.ps1`), `plantillas\` (baterías de preguntas, la instrucción con cita y la lista de privacidad) y `ejemplos\` (tres PDF ficticios).

Contenido

Pregunta a tus PDF sin subirlos a la nube . . . 2
IA local que responde con cita de archivo y página, comprueba sus propias citas y no saca nada de tu PC . . . 2
Alcance . . . 2
Cómo usar este libro . . . 2
Qué necesitas . . . 2
01 · En 60 minutos: tus PDF responden con cita de página 5
Lo que vas a montar . . . 5
Paso 1 · Instala Ollama y descarga el modelo (15 min, casi todo esperando) . . . 5
Paso 2 · Córtale a Ollama la salida a internet (5 min) . . . 6
Paso 3 · Instala Python (5 min) . . . 6
Paso 4 · Instala pdftotext y el paquete (10 min) . . . 6
Paso 5 · Convierte, indexa y pregunta (15 min) . . . 6
Paso 6 · Las tres preguntas que te dicen si puedes fiarte (10 min) . . . 7
Lo que hay debajo: por qué no te hace falta un «RAG» complicado . . . 7
Cómo compruebas que ha salido bien . . . 8
02 · Sacar bien el texto del PDF (y qué hacer con los escaneados) 9
Lo que te puede pasar . . . 9
Por qué pasa . . . 9
Pasos: PDF con texto . . . 9
Pasos: PDF escaneados . . . 10
Cómo compruebas que el texto está bien . . . 11
03 · ¿Cabe entero? Cuándo basta con meterlo todo en el contexto 12
Lo que te puede pasar . . . 12
Por qué pasa . . . 12
Pasos . . . 12
Lo que cuesta . . . 13
Cuándo elegir este modo . . . 13
Resúmenes con referencias . . . 14
Cómo compruebas que ha salido bien . . . 14
04 · Palabras clave: el buscador que te basta casi siempre 15
Lo que quieres conseguir . . . 15
Cómo funciona por dentro . . . 15
Lo que medimos . . . 15
Pasos para afinarlo con tus documentos . . . 16
Cuándo se queda corta . . . 16
Cómo compruebas que ha salido bien . . . 17

05 · Embeddings (RAG): cuando las palabras no coinciden **18**

Qué te pasa . . . 18

Por qué los embeddings lo resuelven (a veces) . . . 18

Lo que medimos . . . 18

Pasos . . . 18

Cómo decides entre los dos modos . . . 19

Cómo compruebas que ha salido bien . . . 19

06 · Comprobar que no se inventa **21**

Qué verifica el script y qué no . . . 21

La batería: diez preguntas que delatan los fallos . . . 22

Lee a mano cada «REVISAR» . . . 23

Repite la batería cuando cambies algo . . . 23

Las órdenes escondidas en un documento . . . 23

Cómo compruebas que ha salido bien . . . 23

07 · Privacidad y RGPD: lo que cambia (y lo que no) al trabajar en local **24**

Lo que quieres conseguir . . . 24

Lo que dice la norma . . . 24

Lo que puedes comprobar en tu equipo . . . 25

Dónde quedan los datos y cómo se borran . . . 26

Cómo compruebas que ha salido bien . . . 26

Fuentes . . . 27

01 · En 60 minutos: tus PDF responden con cita de página

Al terminar este capítulo tendrás en tu PC un sistema que lee una carpeta de PDF, responde a tus preguntas y te dice de qué archivo y de qué página sacó cada dato, con la frase exacta. El script, además, comprueba por su cuenta que esa frase existe en esa página. Nada de lo que hagas aquí sale de tu equipo, y lo vas a poder demostrar.

Necesitas Windows 10 22H2 u 11, unos 15 GB libres en disco y, si es posible, una tarjeta gráfica NVIDIA con 12 GB de memoria o más. Sin gráfica también funciona, más despacio (te lo cuantifico en el paso 3).

Lo que vas a montar

Cuatro piezas, todas gratuitas y todas locales:

Pieza	Para qué sirve
Ollama	Ejecuta el modelo de lenguaje en tu PC y lo ofrece en `127.0.0.1:11434`
pdftotext (Xpdf)	Saca el texto de cada PDF, página a página
Python	Ejecuta el script del paquete
`preguntar_pdf.py`	Trocea el texto, busca los trozos que importan, pregunta al modelo y verifica las citas

Trabajarás en una carpeta, `C:\pdf-local`, con tres subcarpetas: `scripts` (el paquete), `pdf` (tus documentos) y `txt` (el texto que sacaremos de ellos).

Paso 1 · Instala Ollama y descarga el modelo (15 min, casi todo esperando)

Descarga el instalador de Windows desde ollama.com/download y ejecuta `OllamaSetup.exe`. Según la documentación oficial (docs.ollama.com/windows), no pide permisos de administrador y se instala en tu carpeta de usuario.

Elige el modelo según la memoria de tu tarjeta gráfica (la ves en el Administrador de tareas, pestaña Rendimiento, GPU, «Memoria de GPU dedicada»):

Tu equipo	Modelo	Ocupa en disco
Gráfica con 12 GB o más	`gemma4:12b`	8,0 GB
Gráfica con 6-8 GB, o sin gráfica	`qwen3.5:4b`	3,3 GB

Abre PowerShell (tecla Windows, escribe «PowerShell», Intro) y descarga el que te toque:

```
ollama pull gemma4:12b
```

En nuestro equipo de pruebas (RTX 5060 Ti de 16 GB), `gemma4:12b` ocupó entre 8,3 y 9,3 GB de memoria de la gráfica y escribió a 111-113 tokens por segundo. Sin gráfica, `qwen3.5:4b` bajó a 11,9 tokens por segundo: cada respuesta tarda bastante más, pero funciona. Si usas el modelo pequeño, añade `--modelo qwen3.5:4b` a cada orden de `preguntar_pdf.py` que aparece en el libro.

Mientras descarga, sigue con los pasos 3 y 4, y vuelve al 2 cuando termine.

Paso 2 · Córtale a Ollama la salida a internet (5 min)

Lo primero que conviene saber de Ollama si tus documentos son confidenciales: **con la configuración por defecto, se conecta a internet nada más arrancar.** Lo medimos: al abrirse, abrió una conexión cifrada con ollama.com y descargó una lista de modelos recomendados, algunos de los cuales se ejecutan en los servidores de Ollama y no en tu PC. Con la variable `OLLAMA_NO_CLOUD=1`, en la misma prueba no hubo ninguna conexión externa, ni al arrancar ni mientras el modelo respondía.

Con el modelo ya descargado, pega:

```
[Environment]::SetEnvironmentVariable('OLLAMA_NO_CLOUD', '1', 'User')
```

Cierra Ollama desde su icono en la bandeja del sistema (junto al reloj: clic derecho y salir) y vuelve a abrirlo desde el menú Inicio. Comprueba que la variable ha entrado:

```
Select-String -Path "$env:LOCALAPPDATA\Ollama\server.log" -Pattern 'cloud disabled' |
  Select-Object -Last 1
```

Debe aparecer `Ollama cloud disabled: true`. Si pone `false`, Ollama no se reinició del todo: ciérralo otra vez y vuelve a abrirlo. Cuando más adelante quieras descargar otro modelo, quita la variable un momento (capítulo 07) o compruébalo primero con ella puesta.

Paso 3 · Instala Python (5 min)

Python se instala ahora con su «Python install manager». Búscalo en Microsoft Store («Python Install Manager») o descárgalo de python.org/downloads, e instálalo. Después, en una ventana nueva de PowerShell:

```
py install default
py --version
```

La segunda orden debe responder con un número de versión (`Python 3.x.y`). El script del paquete solo usa la biblioteca estándar de Python: no tienes que instalar nada con `pip`.

Paso 4 · Instala pdftotext y el paquete (10 min)

pdftotext forma parte de las «Xpdf command line tools», gratuitas. En xpdfreader.com/download.html descarga las de Windows (el 05/11/2026 la versión vigente era la 4.06, archivo `xpdf-tools-win-4.06.zip`) y descomprime el zip en `C:\`. Te queda el programa en `C:\xpdf-tools-win-4.06\bin64\pdftotext.exe`.

Crea la carpeta de trabajo y copia en ella el paquete del libro:

```
New-Item -ItemType Directory -Force C:\pdf-local\pdf, C:\pdf-local\txt, C:\pdf-local\scripts | Out-Null
```

Copia el contenido de `scripts\` y `plantillas\` del paquete a `C:\pdf-local\scripts`, y los tres PDF de `ejemplos\` a `C:\pdf-local\pdf`. Son documentos ficticios (un contrato de asesoría de tres páginas y dos políticas de gastos, una sustituida y otra vigente) con respuestas conocidas: empieza por ellos y luego cámbialos por los tuyos.

Paso 5 · Convierte, indexa y pregunta (15 min)

```
cd C:\pdf-local\scripts
py preguntar_pdf.py convertir C:\pdf-local\pdf --salida C:\pdf-local\txt `
  --pdftotext C:\xpdf-tools-win-4.06\bin64\pdftotext.exe
```

Por cada PDF verás una línea con sus páginas y caracteres. Si alguna página sale «sin texto», es una imagen escaneada y necesita OCR (capítulo 02). Abre uno de los `.txt` con el Bloc de notas y míralo: si el texto se lee bien, sigue.

```
py preguntar_pdf.py indexar C:\pdf-local\txt
py preguntar_pdf.py preguntar C:\pdf-local\txt "¿A qué juzgados se someten las partes del contrato de
asesoría?"
```

La primera pregunta tarda más porque Ollama carga el modelo en memoria. La salida tiene esta forma:

```
RESPUESTA: Las partes se someten a los juzgados de Zaragoza.
  · contrato-asesoria-ejemplo.txt, p. 3: «Las partes se someten a los juzgados de Zaragoza.»  →
VERIFICADA

Fragmentos usados: [contrato-asesoria-ejemplo.txt, p. 3], [contrato-asesoria-ejemplo.txt, p. 1], ...
Modelo gemma4:12b · modo palabras · … s · leyó … tokens de un contexto de 16384
```

Lo importante es lo que aparece tras la flecha. `VERIFICADA` significa que el script buscó esa frase en esa página del texto y la encontró (sin tener en cuenta mayúsculas, tildes ni saltos de línea). Si pusiera `OTRA PÁGINA (p. 2)`, la frase existe pero el modelo se equivocó de página. `NO ENCONTRADA` significa que la cita no está en el documento: el modelo la ha parafraseado o inventado, y esa respuesta no vale hasta que la compruebes a mano.

Paso 6 · Las tres preguntas que te dicen si puedes fiarte (10 min)

Haz estas tres antes de meter un solo documento real:

1. **Una pregunta cuya respuesta no esté en ningún documento:**

```
py preguntar_pdf.py preguntar C:\pdf-local\txt "¿Qué persona firma el contrato en nombre de Comercial
Ejemplo S.A.?"
```

Debe responder `PENDIENTE`. Si da un nombre, se lo ha inventado.

2. **Un dato que cambió entre versiones:**

```
py preguntar_pdf.py preguntar C:\pdf-local\txt "¿Cuál es el máximo por noche de alojamiento según la
política de gastos vigente?"
```

La vigente dice 95 euros; la sustituida, 80. Si responde 80, ha cogido el documento equivocado.

3. **Un dato de la última página.** La de los juzgados del paso 5 ya lo era. Con tus documentos, busca uno que solo aparezca al final del archivo más largo.

Si quieres pasarlas todas de una vez, el paquete trae diez preguntas ya preparadas para los ejemplos:

```
py preguntar_pdf.py comprobar C:\pdf-local\txt .\preguntas-ejemplos.csv
```

Cada fila sale como `BIEN` o `REVISAR`, y el detalle queda en un CSV que se abre con Excel. Cómo leerlo y cómo preparar las tuyas está en el capítulo 06.

Lo que hay debajo: por qué no te hace falta un «RAG» complicado

El script no le pasa al modelo tus documentos enteros: busca por palabras clave los cuatro trozos que más se parecen a tu pregunta y le pasa solo esos, con su etiqueta de archivo y página. Es una búsqueda sencilla (BM25, la misma idea que un buscador clásico), sin modelos de embeddings ni bases de datos vectoriales.

Lo elegimos por defecto porque lo medimos. Con 14 documentos de normas internas y 18 preguntas, la búsqueda por palabras clave encontró el documento correcto en todas las preguntas que tenían respuesta,

exactamente igual que una búsqueda con embeddings, y fue la opción más rápida de las tres que probamos. Los capítulos 03, 04 y 05 te enseñan cuándo te conviene cada una de ellas con tus propios documentos.

La segunda sorpresa de esa prueba: los 14 documentos, cortos, ya no cabían en el contexto que Ollama usa por defecto en una gráfica de 16 GB. Si los hubiéramos pegado enteros en la pregunta, el modelo habría leído solo una parte, sin avisar. El script evita ese problema porque envía poco texto y fija el contexto en 16.384 tokens; el capítulo 03 explica cuándo te interesa hacer lo contrario y meterlo todo.

Cómo compruebas que ha salido bien

- `server.log` dice `Ollama cloud disabled: true`.
- La pregunta de los juzgados devuelve Zaragoza con la cita `VERIFICADA` en la página 3.
- La pregunta del firmante devuelve `PENDIENTE`.
- La de alojamiento devuelve 95 euros, no 80.
- Opcional, para quedarte tranquilo del todo: abre otra ventana de PowerShell, lanza `powershell -ExecutionPolicy Bypass -File C:\pdf-local\scripts\comprobar-red.ps1 -Segundos 60` y, mientras vigila, haz una pregunta en la primera ventana. Debe terminar con `0 conexiones externas`.

Si todo cuadra, cambia los ejemplos por tus PDF: vacía `C:\pdf-local\pdf` y `C:\pdf-local\txt`, copia tus documentos y repite el paso 5. Antes de hacerlo con documentos de clientes, lee el capítulo 07.