Extraer datos de webs con IA sin líos

ENCARGO INGENUO 85/120

BRIEF OPERATIVO 120/120

Extraer datos de webs con IA, sin meterte en líos

Precios, catálogos, ofertas o normativa en una tabla: la IA escribe el código, tú decides cuándo sí y cuándo no

Edición: 05/11/2026. **Para quién:** eres autónomo, analista o llevas un negocio pequeño y quieres recopilar datos públicos de webs (precios, catálogos, ofertas de empleo, normativa, noticias) para tu trabajo. Usas un chat de IA con soltura y no te asusta abrir una terminal y pegar un comando. No hace falta que sepas programar: la IA escribe el código y el libro te enseña a pedírselo, a ejecutarlo y a comprobar que lo que sale es verdad.

Alcance

Las pruebas de este libro se hicieron el 05/11/2026 en Windows 11 con Python 3.12, requests 2.34, Beautiful Soup 4.15 y Playwright 1.62, siempre contra dos sitios creados para practicar scraping: books.toscrape.com y quotes.toscrape.com. Las normas citadas son las vigentes en España y en la Unión Europea en esa fecha, con enlace a su texto oficial. El capítulo legal explica qué dice la norma y qué preguntas debes hacerte; no es asesoramiento jurídico para tu caso. Si tu proyecto toca datos personales a escala, datos de un competidor o un uso comercial dudoso, consulta con un abogado. Es el único aviso de este tipo que vas a leer.

Cómo usar este libro

Haz el capítulo 01 de principio a fin: en una hora tienes una tabla de 40 filas sacada de una web y sabes qué tres preguntas te ahorran un problema. Después ve a lo que necesites:

Lo que quieres	Capítulo
Saber si hace falta raspar o hay una vía más limpia	02
Leer robots.txt y las condiciones, y no saturar a nadie	03
Pedirle a la IA buen código, o que extraiga ella del texto	04
La página sale vacía cuando la descargas	05
Dejar los datos limpios y la extracción programada	06
Entender RGPD, bases de datos, propiedad intelectual y minería de datos	07

Qué necesitas

- Un ordenador con Windows, macOS o Linux. Los comandos del libro son de Windows (PowerShell); en los otros sistemas cambia `py` por `python3`.
- Acceso a un chat de IA cualquiera (ChatGPT, Claude, Gemini, Copilot u otro). No hace falta plan de pago para lo que pedimos aquí.
- Los dos scripts y la checklist del paquete, en una carpeta como `C:\scraping`.

Contenido

Extraer datos de webs con IA, sin meterte en líos . . . 2
Precios, catálogos, ofertas o normativa en una tabla: la IA escribe el código, tú decides cuándo sí y cuándo no . . . 2
Alcance . . . 2
Cómo usar este libro . . . 2
Qué necesitas . . . 2
01 · Tu primera tabla en 60 minutos 5
Paso 0 (5 minutos): las tres preguntas . . . 5
Paso 1 (10 minutos): prepara Python . . . 5
Paso 2 (10 minutos): la vía rápida, copiar y pegar . . . 6
Paso 3 (10 minutos): enséñale el HTML de un elemento . . . 6
Paso 4 (10 minutos): pídele el programa . . . 7
Paso 5 (10 minutos): ejecútalo y compruébalo . . . 7
Paso 6 (5 minutos): compáralo con el del paquete . . . 7
Cómo sabes que salió bien . . . 8
Cuándo no hacer esto . . . 8
02 · Antes de raspar: API, datos abiertos y exportaciones 9
1. ¿Tiene API? . . . 9
2. ¿Están en un portal de datos abiertos? . . . 9
3. ¿Se puede exportar o hay un feed? . . . 9
4. ¿Puedes pedirlo? . . . 10
5. Solo entonces: raspar . . . 10
La tabla de decisión . . . 10
Pídele ayuda a la IA para la búsqueda, no para la conclusión . . . 10
Cómo sabes que salió bien . . . 10
03 · robots.txt, condiciones de uso y buenos modales 11
robots.txt: qué es y qué no es . . . 11
Leer robots.txt desde Python . . . 12
Condiciones de uso: dónde mirar y qué buscar . . . 12
El ritmo: lento para ti, invisible para el servidor . . . 13
Preséntate . . . 13
Lo que no se hace . . . 13
Cómo sabes que salió bien . . . 13
04 · La IA como extractora: código a partir del HTML o tabla a partir del texto 14
Cuál elegir . . . 14
Ruta A: que la IA escriba el código . . . 14
Ruta B: que la IA extraiga del texto . . . 16

Combinar las dos . . . 17
Cómo sabes que salió bien . . . 17

05 · Páginas que se pintan con JavaScript 18

Paso 1: confirma que es JavaScript . . . 18
Paso 2: antes del navegador, busca los datos en la fuente . . . 18
Paso 3: si no hay otra, un navegador sin ventana . . . 19
Lo que cuesta el navegador . . . 19
Lo que no hacer con Playwright . . . 20
Cómo sabes que salió bien . . . 20

06 · Limpiar, validar y automatizar con prudencia 21

Limpiar: decide el formato antes de mirar los datos . . . 21
Validar: cinco comprobaciones que no fallan . . . 21
Guarda solo lo que necesitas . . . 22
Automatizar: un programa que nadie vigila es un riesgo . . . 22
Cómo sabes que salió bien . . . 23

07 · Lo legal, sin sustos 24

1. ¿Hay datos personales? El RGPD . . . 24
2. ¿Es una base de datos? El derecho «sui generis» . . . 25
3. ¿Copias contenidos protegidos? Propiedad intelectual . . . 25
4. ¿Qué aceptaste? Las condiciones de uso . . . 26
5. ¿Vas a analizar en masa? La minería de textos y datos . . . 26
6. ¿Te saltas una barrera? El Código Penal . . . 26
Resumen en una tabla . . . 27
Fuentes . . . 27

01 · Tu primera tabla en 60 minutos

Vas a sacar de una web una tabla con título, precio, valoración y enlace de 40 libros, sin escribir tú una sola línea de código. Lo harás de dos maneras: primero copiando texto y pidiéndole la tabla al chat, y luego pidiéndole al chat un programa que la saque solo. Por el camino vas a ver lo que casi nadie cuenta: la vía rápida, la de copiar y pegar, pierde la mitad de los datos sin avisar. Y antes de empezar vas a hacerte tres preguntas que, en una web real, son las que te evitan el problema.

Trabajamos sobre books.toscrape.com, una librería ficticia hecha para que la gente aprenda scraping. Sus precios y valoraciones son inventados (lo dice un aviso en la propia página) y no tiene robots.txt que prohíba nada. Es el sitio adecuado para equivocarte sin molestar a nadie.

Paso 0 (5 minutos): las tres preguntas

Antes de tocar una web que no es de práctica, contesta esto. Si alguna respuesta es «no lo sé», la respuesta correcta es parar y mirarlo.

1. **¿Hay una vía oficial?** Una API, una descarga en CSV, un portal de datos abiertos, un feed. Si existe, úsala: es más estable y no te pone en duda (capítulo 02).
2. **¿Lo permiten?** Mira `robots.txt` y las condiciones de uso de la web. Si prohíben la extracción automatizada, no la hagas (capítulo 03).
3. **¿Hay personas en los datos?** Nombres, correos, teléfonos, fotos, perfiles. Si los hay, entras en el terreno del RGPD aunque los datos sean públicos (capítulo 07).

En books.toscrape.com las tres respuestas son fáciles: no hay API, es un sitio pensado para esto y no hay datos de personas. Adelante.

Paso 1 (10 minutos): prepara Python

Comprueba si ya tienes Python. Abre PowerShell (tecla Windows, escribe «PowerShell», Intro) y escribe:

```
py --version
```

Si responde algo como `Python 3.12.10`, sigue. Si dice que no reconoce el comando, descarga el instalador desde python.org/downloads, instálalo con las opciones por defecto, cierra PowerShell, ábrelo otra vez y repite la comprobación.

Ahora instala las dos librerías que usa casi todo el scraping sencillo en Python: requests, que descarga páginas, y Beautiful Soup, que las lee:

```
py -m pip install requests beautifulsoup4
```

Crea una carpeta de trabajo y entra en ella:

```
mkdir C:\scraping
cd C:\scraping
```

Paso 2 (10 minutos): la vía rápida, copiar y pegar

Abre books.toscrape.com en el navegador. Selecciona con el ratón la cuadrícula de libros, desde el primero hasta el último de la página, cópiala y pégala en tu chat de IA con esta petición:

```
Te pego el texto que he copiado de una página de una librería. Conviértelo
en una tabla Markdown con estas columnas: título completo, precio en libras
(solo número), valoración en estrellas (1 a 5) y disponibilidad. Si un dato
no aparece en el texto, escribe «no consta» en esa celda; no lo deduzcas ni
lo completes. Al final, di cuántas filas tiene la tabla.
```

Lo probamos el 05/11/2026 con un modelo de chat. La tabla salió con 20 filas y los 20 precios correctos, comparados uno a uno con la página. Pero 10 de los 20 títulos venían cortados, como «A Light in the ...», y la columna de estrellas salió entera como «no consta». No es un fallo de la IA: es que el texto visible no lleva esa información. La página corta los títulos largos para que quepan en la cuadrícula y dibuja las estrellas con iconos, que no se copian.

Fíjate en la frase «no lo deduzcas ni lo completes». Sin ella, un modelo puede rellenar títulos que conoce o poner estrellas plausibles, y la tabla parece perfecta. Con ella, te dice lo que falta. El capítulo 04 vuelve sobre esto.

La vía rápida sirve para una página suelta y para datos que se ven escritos. Para 40 filas con todos los campos, necesitas leer el HTML.

Paso 3 (10 minutos): enséñale el HTML de un elemento

El HTML es el código de la página, y ahí sí están el título completo y las estrellas. No hace falta que lo entiendas entero: basta con copiar el trozo de un libro.

1. En books.toscrape.com, haz clic derecho sobre el título del primer libro y elige **Inspeccionar**.
2. Se abre un panel con el código. Sube con la flecha del teclado (o con el ratón) hasta la línea que empieza por `<article class="product_pod">`; al pasar por encima, el navegador resalta el libro entero.
3. Clic derecho sobre esa línea → **Copiar** → **Copiar elemento** (en Chrome y Edge; en Firefox, **Copiar** → **HTML externo**).

Tendrás algo así (lo hemos acortado):

```
<article class="product_pod">
  <p class="star-rating Three"><i class="icon-star"></i></p>
  <h3><a href="catalogue/a-light-in-the-attic_1000/index.html"
         title="A Light in the Attic">A Light in the ...</a></h3>
  <div class="product_price"><p class="price_color">£51.77</p></div>
</article>
```

Aquí está todo: el título completo en `title`, las estrellas como palabra en la clase (`Three`) y el enlace a la ficha. Busca también, al final de la página, el botón **next**, inspecciónalo y copia su elemento: `<li class="next"><a href="catalogue/page-2.html">next</a></li>`.

Paso 4 (10 minutos): pídele el programa

Pega en el chat esta petición, con los dos trozos de HTML que acabas de copiar donde se indica:

```
Quiero un script de Python para extraer datos de un sitio de práctica de
scraping, https://books.toscrape.com/ (está hecho para practicar).

Datos que quiero, uno por libro: título completo (está en el atributo title
del enlace), precio (número, sin el símbolo £), valoración en estrellas como
número del 1 al 5 y la URL completa de la ficha.

Este es el HTML de UN libro, copiado de la página:
[pega aquí el <article class="product_pod">...</article>]

La página siguiente está en un enlace así:
[pega aquí el <li class="next">...</li>]

Condiciones:
- Usa requests y BeautifulSoup.
- Recorre solo las 2 primeras páginas.
- Espera 1 segundo entre peticiones, pon timeout de 20 segundos y un
  User-Agent que diga "PracticaScraping/1.0 (+mailto:tu-correo@ejemplo.com)".
- Guarda el resultado en libros.csv con cabecera, en UTF-8.
- Al final imprime cuántas filas has guardado.
- Responde SOLO con el código, en un único bloque, con comentarios breves en español.
```

Cada condición tiene su porqué. «Solo 2 páginas» limita el daño si el programa se equivoca. La pausa de 1 segundo y el `timeout` son buenos modales y evitan que el programa se quede colgado; la documentación de requests dice que casi todo el código de producción debería poner `timeout` en casi todas las peticiones. El User-Agent con un correo dice quién eres. Y pedir que imprima el número de filas te da la primera comprobación sin abrir el archivo.

Cuando lo probamos, el modelo devolvió un script que funcionó a la primera.

Paso 5 (10 minutos): ejecútalo y compruébalo

Copia el código, ábrelo en el Bloc de notas, pégalo y guárdalo en `C:\scraping` como `libros_ia.py` (en «Tipo», elige «Todos los archivos» para que no lo guarde como `.txt`). Ejecútalo:

```
py libros_ia.py
```

Debe decir `Filas guardadas: 40`. Ahora comprueba tres cosas, que son las mismas que harás en cualquier extracción:

1. **El número.** La web muestra 20 libros por página; 2 páginas son 40. Si salen 0, el programa no encuentra los elementos (capítulo 04). Si salen 20, no ha pasado de página.
2. **Tres filas al azar.** Abre `libros.csv` con el Bloc de notas o con Excel, elige tres libros y compáralos con la web: título completo, precio y estrellas.
3. **Los raros.** Busca celdas vacías o con símbolos extraños, como `Â£`. Si ves eso, es un problema de codificación: pídele al chat que fije `respuesta.encoding = "utf-8"`.

Si Excel te muestra los acentos mal, abre el CSV desde **Datos** → **Desde texto/CSV** y elige UTF-8.

Paso 6 (5 minutos): compáralo con el del paquete

El paquete trae `libros_toscrape.py`, que hace lo mismo y añade dos cosas que un script de chat rara vez incluye si no se las pides: lee `robots.txt` antes de empezar y se detiene si el servidor contesta con un 429 («demasiadas peticiones»). Lánzalo con el catálogo entero:

```
py C:\scraping\libros_toscrape.py --paginas 50 --salida catalogo.csv
```

En nuestra prueba recorrió las 50 páginas y guardó 1.000 libros en 76,4 segundos, casi todo esperando la pausa de un segundo entre páginas. Ese es el ritmo que tiene que tener una extracción respetuosa: lenta para ti, invisible para el servidor.

Cómo sabes que salió bien

- Tienes `libros.csv` con 40 filas y cuatro columnas, y tres filas al azar coinciden con la web.
- Sabes por qué la tabla del paso 2 tenía títulos cortados y estrellas vacías.
- Puedes contestar las tres preguntas del paso 0 para la próxima web que quieras usar.

Cuándo no hacer esto

Lo que acabas de aprender funciona en casi cualquier web, y por eso conviene decir dónde no lo usarías. No raspes una web que lo prohíbe en sus condiciones o en `robots.txt`, aunque técnicamente puedas. No montes una base de datos de personas (perfiles, contactos, opiniones con nombre) sin haber leído el capítulo 07. No copies el contenido de otro (artículos, fotos, fichas de producto) para publicarlo como tuyo. Y nunca intentes saltarte un bloqueo, un inicio de sesión o un CAPTCHA: si la web te para, la respuesta es parar.