

Estrategias Avanzadas de Extracción Programática de Datos en Marketplaces: Un Marco Técnico para la Inteligencia de Mercado de Bajo Coste

En el actual ecosistema de comercio electrónico, la capacidad de recolectar, procesar y analizar datos de competidores de manera masiva y económica constituye el núcleo de la ventaja competitiva en el sector del retail y la reventa. Las plataformas líderes como Amazon, eBay, Wallapop y Vinted han evolucionado significativamente en sus mecanismos de defensa, pasando de simples bloqueos por dirección IP a sofisticados sistemas de análisis de comportamiento y huella digital del navegador. Para los analistas de mercado que buscan desarrollar soluciones propias basadas en herramientas de código abierto, el desafío no solo consiste en la extracción de la información, sino en la construcción de arquitecturas que garanticen la persistencia, la calidad de los datos y la evasión de contramedidas sin incurrir en los elevados costes de los servicios gestionados comerciales.

La transición de scripts rudimentarios hacia sistemas de extracción industrializados exige un conocimiento profundo de la infraestructura de las plataformas objetivo. Mientras Amazon utiliza AWS WAF para filtrar el tráfico automatizado basándose en señales de red y de cliente, plataformas como Wallapop y Vinted han optado por asegurar sus APIs mediante firmas criptográficas y sistemas de gestión de sesiones vinculados a la actividad del usuario. El presente informe detalla las estrategias técnicas para superar estos obstáculos, priorizando el uso de tecnologías como Python, Playwright, Scrapy, DuckDB y herramientas de ingeniería inversa para la obtención de métricas críticas como el precio histórico, el volumen de ventas estimado mediante deltas de stock y el tiempo de permanencia de los productos en el mercado.

Arquitectura de Datos para el Análisis de Mercado a Gran Escala

El éxito de una operación de scraping para estudios de mercado no se mide únicamente por la cantidad de datos extraídos, sino por la eficiencia con la que estos se transforman en conocimiento accionable. En un entorno donde se monitorizan millones de productos de forma diaria, la elección del motor de base de datos y la biblioteca de procesamiento determina la viabilidad económica y técnica del proyecto.

Optimización del Almacenamiento: El Paradigma de DuckDB y Polars

Tradicionalmente, las soluciones de almacenamiento local se han basado en SQLite debido a

su ligereza. Sin embargo, para cargas de trabajo analíticas (OLAP) típicas de los estudios de mercado, donde se realizan agregaciones complejas sobre series temporales de precios, DuckDB se ha consolidado como la opción superior.¹ DuckDB utiliza un motor de ejecución vectorial y un almacenamiento columnar que permite procesar grandes conjuntos de datos de forma mucho más eficiente que los sistemas basados en filas, optimizando el uso de la CPU y reduciendo la latencia en las consultas de agregación.¹

La integración de DuckDB con Polars, una biblioteca de DataFrames escrita en Rust, permite un procesamiento de datos con "zero-copy" mediante el formato Apache Arrow.³ Esta sinergia es fundamental para analistas que operan en máquinas locales o servidores de bajo coste, ya que permite manejar volúmenes de datos que anteriormente requerirían infraestructuras de nube costosas como BigQuery o Snowflake. Mientras que SQLite puede degradar su rendimiento significativamente cuando los datos superan unos pocos gigabytes, DuckDB está diseñado para escalar hasta terabytes de datos de mercado utilizando técnicas de derrame a disco (spill-to-disk) cuando la memoria RAM es insuficiente.¹

Característica	SQLite (OLTP)	DuckDB (OLAP)
Modelo de Almacenamiento	Basado en filas	Basado en columnas
Ejecución	Fila por fila (Single-threaded)	Vectorizada (Multi-threaded)
Rendimiento Analítico	Lento en grandes agregaciones	3x-50x más rápido en consultas complejas
Formatos Soportados	Archivos.db nativos	CSV, Parquet, JSON, Arrow, Polars
Concurrencia	Un escritor a la vez	MVCC (Multi-version Concurrency Control)
Aplicación Principal	Estado de apps, transacciones pequeñas	Ciencia de datos, BI, series temporales

Para el seguimiento de precios y stock, se recomienda una estructura de datos que priorice el uso de tipos restrictivos. El uso de BIGINT para identificadores y DOUBLE o DECIMAL para precios reduce el tamaño en disco y acelera las operaciones matemáticas.⁶ La partición de los datos por marca de tiempo (TIMESTAMP) es una práctica crítica que permite el "podado"

de archivos, evitando que el motor de búsqueda deba escanear todo el historial cuando solo se requiere analizar el comportamiento del mercado en el último trimestre.⁷

Metodología de Estimación de Ventas mediante el Seguimiento de Stock

Uno de los mayores retos en la investigación de marketplaces es la opacidad de los datos de ventas reales. La estrategia más efectiva y económica para inferir estas cifras es el algoritmo de "seguimiento de deltas de stock".⁸ Esta técnica consiste en registrar el nivel de inventario disponible de un producto en intervalos fijos. La disminución del stock entre dos capturas temporales se interpreta como una venta, asumiendo que el vendedor no ha retirado el artículo manualmente.

Para implementar este algoritmo de forma robusta, el sistema debe ser capaz de identificar las reposiciones. Si el stock en el tiempo t es mayor que en $t - 1$, se ha producido un reabastecimiento, y el cálculo de ventas debe ajustarse para no generar valores negativos.⁸ En Amazon, por ejemplo, los scrapers suelen utilizar la técnica de añadir 999 unidades al carrito para forzar a la plataforma a revelar la cantidad exacta disponible, siempre que esta sea menor al límite permitido.⁸

$$Ventas_{estimadas} = \sum_{t=1}^n \max(0, Stock_{t-1} - Stock_t)$$

Este enfoque permite a los analistas calcular la velocidad de ventas de los competidores y anticipar roturas de stock, lo que a su vez ofrece oportunidades para ajustar los precios propios al alza cuando la oferta del mercado disminuye.⁸ El seguimiento del tiempo en venta (time-on-market) se complementa con esta métrica, permitiendo identificar productos que, aunque tengan un precio bajo, poseen una rotación lenta que podría no justificar la inversión.¹¹

Amazon: Superación de AWS WAF y Automatización Sigilosa

Amazon representa el estándar de oro en protección anti-bot. Su sistema de seguridad, basado en AWS WAF, analiza múltiples capas de la interacción entre el cliente y el servidor para determinar si la petición es legítima. Para los desarrolladores que buscan soluciones de bajo coste, el enfoque debe alejarse de los reintentos masivos de IPs y centrarse en la coherencia de las señales del navegador.¹²

Estrategias de Navegación Sigilosa con Playwright y Puppeteer

El uso de navegadores automatizados es necesario para manejar el contenido dinámico de Amazon, pero el modo "headless" estándar es fácilmente detectable. Las plataformas de seguridad buscan el flag navigator.webdriver, que se establece automáticamente en true cuando el navegador es controlado por software.¹³ La inyección de scripts al inicio de la navegación es la técnica fundamental para enmascarar esta señal:

Python

```
# Ejemplo de inyección de script en Playwright para desactivar el flag webdriver
context.add_init_script("""
    Object.defineProperty(navigator, 'webdriver', {
      get: () => undefined
    })
  """)
```

Sin embargo, el sigilo moderno requiere más que ocultar un flag. La coherencia en la huella digital (TLS fingerprinting) es vital. Amazon puede detectar inconsistencias entre el User-Agent declarado y las capacidades reales del motor de renderizado del navegador, como el soporte de WebGL o la resolución de pantalla.¹² El uso de complementos como puppeteer-extra-plugin-stealth o playwright-stealth automatiza la mayoría de estas correcciones, pero un enfoque de ingeniería superior implica mantener una configuración estable de proxies y zonas horarias para evitar señales contradictorias.¹²

Técnica de Sigilo	Descripción	Impacto en la Evasión
Rotación de User-Agent	Cambiar la cadena de identificación del navegador.	Bajo (fácil de detectar si es inconsistente).
Proxy Rotation	Utilizar diferentes IPs para distribuir la carga.	Alto (previene el bloqueo por tasa de peticiones).
JavaScript Injection	Modificar propiedades del DOM como navigator.webdriver.	Crítico (elimina el marcador de automatización más común).
Human-like Interaction	Simular movimientos de ratón y tiempos de espera	Medio (ayuda frente a análisis de

	aleatorios.	comportamiento).
TLS Fingerprinting	Asegurar que el apretón de manos SSL parezca de un navegador real.	Muy Alto (difícil de evadir sin herramientas específicas).

Resolución Local de CAPTCHAs mediante ddddocr

Cuando las técnicas de sigilo fallan, Amazon presenta desafíos de CAPTCHA visuales. El coste de enviar estos desafíos a servicios humanos como 2Captcha o CapMonster puede escalar rápidamente en proyectos de gran volumen.¹⁶ Como alternativa gratuita, el uso de la biblioteca ddddocr permite la resolución local de CAPTCHAs de texto y deslizamiento mediante modelos de aprendizaje profundo pre-entrenados.¹⁸

ddddocr funciona de forma completamente offline, lo que garantiza la privacidad de los datos y elimina la latencia de red. El sistema procesa la imagen convirtiéndola a escala de grises, normalizando los valores de los píxeles y ejecutando una inferencia a través de un modelo ONNX para predecir los caracteres con una precisión sorprendentemente alta para los CAPTCHAs estándar de Amazon.¹⁹ En casos donde la precisión local sea insuficiente, la integración de modelos de visión por inteligencia artificial como GPT-4o-mini ofrece una tasa de éxito cercana al 100% por un coste marginal, siendo una opción híbrida recomendable.²¹

eBay: Arquitectura de Doble Spider y Normalización de Datos

eBay presenta un reto distinto caracterizado por la heterogeneidad de sus listados y la frecuencia con la que actualiza sus selectores CSS, lo que puede romper los scrapers basados en rutas estáticas.²² Para mitigar esto, se recomienda una arquitectura basada en dos tipos de spiders especializados, optimizando tanto la velocidad como el consumo de recursos.

Especialización de Spiders: Búsqueda vs. Producto

El "Spider de Búsqueda" (Search Spider) está diseñado para la amplitud. Su objetivo es escanear los resultados de búsqueda, identificar los productos y recolectar metadatos básicos como el título, el precio y el identificador único. Una lección crítica en eBay es el manejo de los elementos promocionales inyectados, como las tarjetas de "Shop on eBay", que carecen de IDs reales y pueden contaminar los datos si no se filtran mediante lógica local (por ejemplo, validando que el ID de producto tenga al menos 9 dígitos).²²

El "Spider de Producto" (Product Spider) se encarga de la profundidad. Una vez obtenidos los

IDs, este spider visita cada página individual para extraer especificaciones técnicas, historial del vendedor y descripciones detalladas. Para combatir la fragilidad de los selectores, se implementan sistemas de "cascada" (fallbacks), donde el script intenta recolectar el precio o el stock desde múltiples ubicaciones posibles en el DOM si el selector principal falla.²²

Python

```
# Ejemplo de lógica de selectores en cascada para eBay
price = response.css('.x-price-primary.ux-textspans::text').get()
if not price:
    price = response.css('.x-price-approx__price.ux-textspans::text').get()
if not price:
    price = response.css('#prclsum::text').get()
```

Pipelines de Limpieza y Estandarización Internacional

eBay opera en múltiples regiones con diferentes monedas y formatos de estado del producto. Un pipeline de datos robusto debe incluir una etapa de normalización local. Esto implica el uso de expresiones regulares para extraer valores numéricos de cadenas de texto complejas y la unificación de términos de condición (e.g., "Brand New", "Nuevo", "New with tags") en categorías estandarizadas para el análisis posterior en DuckDB.²² Esta limpieza es esencial para realizar estudios de mercado transfronterizos donde se busca comparar precios reales sin el sesgo de la divisa o la terminología regional.²³

Wallapop: Ingeniería Inversa y el Desafío del X-Signature

Wallapop es una plataforma diseñada principalmente para el entorno móvil, lo que significa que su versión web es una aplicación de una sola página (SPA) que consume una API interna muy estructurada. El acceso a esta API es, en teoría, la forma más eficiente de extraer datos, ya que devuelve objetos JSON limpios sin necesidad de parsear HTML pesado. Sin embargo, Wallapop protege sus endpoints mediante una cabecera de autenticación denominada X-Signature.²⁴

Descifrado del Algoritmo X-Signature en la Era React

Tras su migración a React, Wallapop actualizó su algoritmo de firma. La comunidad de ingeniería inversa ha identificado que la firma se genera mediante un proceso de mangling y hashing que combina el método HTTP, el endpoint de la URL y una marca de tiempo con una

clave secreta.²⁵ El descubrimiento de esta clave fue posible mediante el análisis de las herramientas de desarrollo del navegador y el uso de Frida para interceptar funciones en tiempo de ejecución.²⁴

Componente	Descripción	Función
Clave Secreta	Cadena Base64 específica de la plataforma.	Semilla para el hashing HMAC-SHA256.
Separador	Carácter pipe (`	`).
Payload	[método, url, timestamp].	Datos que garantizan la unicidad de la petición.
X-Signature	Resultado del hash codificado en Base64.	Cabecera de validación requerida por la API.

La implementación programática de esta firma permite realizar consultas directas al endpoint `/api/v3/general/search` con parámetros como `items=keyword`, permitiendo una extracción de datos a una velocidad que los scrapers basados en navegador no pueden igualar.²⁴ Este enfoque de "API-first" es el epítome de la eficiencia de bajo coste, ya que reduce el consumo de ancho de banda y CPU en órdenes de magnitud.

Análisis de Tráfico Móvil y Reverse Engineering

Para descubrir nuevos endpoints o cambios en la estructura de la API móvil, se utilizan técnicas de interceptación de tráfico mediante proxies como Charles Proxy o Burp Suite.²⁶ Dado que las versiones modernas de Android (7.0+) restringen la confianza en certificados de usuario, los analistas suelen recurrir a emuladores con versiones antiguas de Android (como la 6.0) o dispositivos con privilegios de root para instalar certificados TLS a nivel de sistema, permitiendo la inspección del tráfico cifrado.²⁸ Este proceso revela no solo los datos de los productos, sino también identificadores internos de usuarios y reputación, vitales para estudios de confianza en la economía colaborativa.²⁹

Vinted: Gestión de Sesiones y la API Pro como Alternativa

Vinted presenta una arquitectura de seguridad basada en el control de sesiones. A diferencia de Wallapop, donde la firma es estática por petición, Vinted requiere que el cliente obtenga una cookie de sesión válida antes de permitir cualquier consulta a su catálogo. Ignorar este

paso resulta en bloqueos inmediatos o en la recepción de datos vacíos.³⁰

Inicialización de Sesiones y Cookies en Python

Las herramientas de código abierto como vinted-scraper gestionan este proceso de forma automática. La lógica consiste en realizar una petición inicial a la página principal de Vinted para recolectar las cookies de sesión y CSRF necesarias, que luego se adjuntan a todas las peticiones subsiguientes a la API interna.³⁰ Es fundamental que el scraper mantenga estas cookies en un objeto `requests.Session()` para simular la persistencia de un usuario real.

Python

```
# Lógica de inicialización de sesión para Vinted
```

```
import requests
```

```
session = requests.Session()
```

```
# El primer GET establece las cookies necesarias
```

```
session.get("https://www.vinted.es", headers={"User-Agent": "Mozilla/5.0..."})
```

```
# Ahora se puede consultar la API interna
```

```
params = {"search_text": "board games", "catalog": "1904"}
```

```
response = session.get("https://www.vinted.es/api/v2/catalog/items", params=params)
```

Vinted Pro Integrations: ¿Cuándo merece la pena pagar?

Vinted ofrece una API oficial denominada "Vinted Pro Integrations" dirigida a vendedores profesionales. Aunque esta opción puede parecer contraria al enfoque de bajo coste, su estabilidad y el uso de webhooks para recibir notificaciones en tiempo real sobre cambios en los artículos pueden justificar la inversión en proyectos comerciales de alta frecuencia.³³ La API Pro utiliza firmas HMAC-SHA256 para cada petición, lo que elimina la necesidad de gestionar cookies de sesión volátiles y reduce el riesgo de baneos de IP.³³

Aspecto	API Interna (Gratis)	Vinted Pro API (Oficial)
Coste	\$0	Requiere cuenta Pro y validación.
Seguridad	Cookies de sesión, anti-bot.	Firma HMAC-SHA256, API Keys.

Fiabilidad	Media (se rompe con cambios de web).	Alta (documentada y estable).
Velocidad	Limitada por rate-limiting de IP.	120 llamadas/min por endpoint.
Datos	Todos los públicos.	Gestión de inventario y pedidos.

Infraestructura de Red: Proxies, TOR y Evasión de Bloqueos

Incluso con el mejor scraper, una dirección IP única será bloqueada rápidamente si realiza miles de peticiones. La gestión de la red es el componente que suele consumir la mayor parte del presupuesto, por lo que las estrategias de bajo coste deben ser creativas.

El Uso de TOR y Circuitos Stem para Rotación Gratuita

La red TOR ofrece una fuente inagotable de direcciones IP de forma gratuita. Mediante la biblioteca stem en Python, es posible controlar el proceso de TOR para solicitar un cambio de identidad (y por ende, una nueva IP) de forma programática.³⁵ Aunque TOR es más lento que un proxy residencial y muchas plataformas bloquean sus nodos de salida conocidos, sigue siendo una herramienta valiosa para la extracción de datos de baja prioridad o para plataformas con defensas menos agresivas.³⁷

Para optimizar el uso de TOR, se recomienda integrar un proxy intermedio como Privoxy, que convierte el tráfico SOCKS5 de TOR en HTTP, permitiendo que bibliotecas como Requests o Scrapy lo consuman de forma nativa.³⁶

Gestión Inteligente de Proxies de Bajo Coste

Cuando TOR es insuficiente, la alternativa es recurrir a listas de proxies públicos. Estos son inherentemente inestables y requieren una lógica de validación constante. Un sistema de rotación eficaz debe:

1. Mantener una lista de proxies activos.
2. Implementar reintentos con una nueva IP cuando una petición devuelve un código de error (403, 429).¹²
3. Aplicar un "backoff" exponencial, aumentando el tiempo de espera entre peticiones si la tasa de fallos se incrementa.¹²

Para proyectos que requieren un rendimiento superior, el uso de servicios como ScraperAPI o

Scrape.do ofrece una capa de abstracción que maneja la rotación de proxies y la resolución de CAPTCHAs por un coste mínimo (a menudo con capas gratuitas generosas), lo que puede ser más rentable que desarrollar y mantener una infraestructura de red propia compleja.⁴⁰

Análisis Legal y Ético del Scraping en España y la UE

El marco legal del web scraping en España ha sido moldeado por la jurisprudencia del Tribunal Supremo y la normativa europea de protección de datos. Comprender estos límites es esencial para que un estudio de mercado sea sostenible y no resulte en sanciones económicas o demandas judiciales.

El Precedente de Ryanair contra Atrápalo

La sentencia del Tribunal Supremo de 9 de octubre de 2012 (STS 572/2012) es el pilar fundamental en España. El tribunal determinó que el scraping de datos públicos de precios para su posterior comparación no constituye un acto de competencia desleal ni una infracción de los derechos de autor, siempre que la extracción no suponga una carga técnica excesiva para el servidor del sitio web original y no se apropie de la reputación ajena de forma ilícita.⁴² Esta sentencia valida la técnica del scraping para fines de transparencia y comparación de mercado, pilares de la libre competencia.

RGPD, LSSI y la Protección de Datos Personales

A pesar de la legalidad de extraer datos factuales (precios, descripciones de productos), la recolección de datos personales es un área de alto riesgo. El RGPD en Europa y la LSSI en España prohíben el procesamiento de información que identifique a personas físicas sin una base legal clara (como el consentimiento o el interés legítimo debidamente justificado).⁴⁵

Para un estudio de mercado ético y legal, se deben aplicar las siguientes salvaguardas:

- **Anonimización:** Eliminar nombres de vendedores particulares, correos electrónicos y números de teléfono de los conjuntos de datos extraídos.⁴⁵
- **Minimización:** No recolectar más datos de los estrictamente necesarios para el análisis de precios y stock.⁴⁶
- **Respeto al robots.txt:** Aunque no siempre es vinculante, seguir las directrices de este archivo demuestra buena fe y reduce el riesgo de ser acusado de intrusión maliciosa o abuso de recursos.⁴⁵

Riesgo Legal	Descripción	Mitigación
Violación del RGPD	Recolección de nombres, perfiles de redes sociales o IPs.	Filtrado de PII (Personally Identifiable Information).

Incumplimiento Contractual	Violación de los Términos de Servicio (ToS).	Scraping de datos públicos sin login siempre que sea posible.
Infracción de Propiedad Intelectual	Copia masiva de imágenes protegidas o bases de datos creativas.	Uso de datos factuales; no redistribución de contenido visual.
Daño al Servidor (DoS)	Sobrecarga de la infraestructura de la plataforma.	Implementación estricta de rate-limiting y throttling.

Conclusiones Estratégicas y Recomendaciones Técnicas

La construcción de un sistema de scraping programático de bajo coste para marketplaces líderes es un desafío de ingeniería que recompensa la profundidad técnica sobre la fuerza bruta. La capacidad de operar en la sombra de las plataformas de seguridad permite obtener datos de una calidad superior a los agregadores comerciales.

1. **Priorizar la Eficiencia en el Almacenamiento:** El uso de DuckDB no es solo una elección técnica, es una decisión económica que permite realizar análisis de "Big Data" en máquinas locales de bajo coste, eliminando facturas de almacenamiento en la nube.¹
2. **Invertir en Ingeniería Inversa:** Descifrar firmas como el X-Signature de Wallapop o entender la gestión de sesiones de Vinted ofrece una ventaja operativa masiva, permitiendo el acceso a datos limpios y rápidos a través de APIs internas.²⁵
3. **Adopción de IA para el Mantenimiento:** El uso de modelos de lenguaje para ajustar automáticamente los selectores CSS cuando las páginas cambian reduce drásticamente las horas de mantenimiento manual, uno de los costes ocultos más altos del scraping propio.¹⁰
4. **Enfoque en el Seguimiento de Stock:** Más que el precio puntual, el seguimiento de la disponibilidad permite una comprensión profunda de la demanda y las ventas reales, proporcionando una inteligencia de mercado que va más allá de la simple comparación de etiquetas de precio.⁸
5. **Responsabilidad Legal:** El cumplimiento del marco español y europeo no solo evita sanciones, sino que garantiza que los datos obtenidos puedan ser utilizados de forma legítima en informes comerciales y procesos de toma de decisiones estratégicas.⁴²

En última instancia, el scraping programático es un juego de gato y ratón. Las plataformas seguirán elevando sus muros, pero mediante el uso de herramientas de código abierto, la

simulación de comportamiento humano y el análisis inteligente de protocolos, los analistas de mercado pueden mantener una ventana abierta a la realidad del comercio digital de forma sostenible y económica.

Obras citadas

1. DuckDB vs SQLite: Performance, Speed, and Use Cases Compared - Hakuna Matata Tech, fecha de acceso: febrero 3, 2026, <https://www.hakunamatatatech.com/our-resources/blog/sqlite>
2. Ultimate guide to DuckDB library in Python - Deepnote, fecha de acceso: febrero 3, 2026, <https://deepnote.com/blog/ultimate-guide-to-duckdb-library-in-python>
3. Integration with Polars - DuckDB, fecha de acceso: febrero 3, 2026, <https://duckdb.org/docs/stable/guides/python/polars>
4. Integration with Polars - DuckDB, fecha de acceso: febrero 3, 2026, <https://duckdb.org/docs/stable/guides/python/polars.html>
5. DuckDB beats Polars for 1TB of data. - Confessions of a Data Guy, fecha de acceso: febrero 3, 2026, <https://www.confessionsofadataguy.com/duckdb-beats-polars-for-1tb-of-data/>
6. Schema - DuckDB, fecha de acceso: febrero 3, 2026, <https://duckdb.org/docs/stable/guides/performance/schema>
7. Streaming Patterns with DuckDB, fecha de acceso: febrero 3, 2026, <https://duckdb.org/2025/10/13/duckdb-streaming-patterns>
8. Optimize Inventory Levels with Scraper APIs - Traject Data, fecha de acceso: febrero 3, 2026, <https://trajectdata.com/scrape-inventory/>
9. How Web Scraping Enhances Real-Time Inventory Tracking & Pricing? - PromptCloud, fecha de acceso: febrero 3, 2026, <https://www.promptcloud.com/blog/optimizing-inventory-and-pricing-with-web-scraping/>
10. eCommerce Data Scraping for Pricing, Sentiment & Stock - GroupBWT, fecha de acceso: febrero 3, 2026, <https://groupbwt.com/blog/ecommerce-data-scraping/>
11. Why Retailers Need Web Scraping, Product Matching, BI - Datahut Blog, fecha de acceso: febrero 3, 2026, <https://www.blog.datahut.co/post/why-retailers-should-invest-in-web-scraping-product-matching-and-bi>
12. Stealth Scraping with Puppeteer or Playwright at Scale - Browserless, fecha de acceso: febrero 3, 2026, <https://www.browserless.io/blog/stealth-scraping-puppeteer-playwright>
13. Avoid Bot Detection With Playwright Stealth: 9 Solutions for 2025, fecha de acceso: febrero 3, 2026, <https://www.scrapeless.com/en/blog/avoid-bot-detection-with-playwright-stealth>
14. From Puppeteer stealth to Nodriver: How anti-detect frameworks evolved to evade bot detection - Security Boulevard, fecha de acceso: febrero 3, 2026, <https://securityboulevard.com/2025/06/from-puppeteer-stealth-to-nodriver-how-anti-detect-frameworks-evolved-to-evade-bot-detection/>

15. How to Use Puppeteer Stealth in 5 Steps (2025) - Roundproxies, fecha de acceso: febrero 3, 2026, <https://roundproxies.com/blog/puppeteer-stealth/>
16. How To Solve CAPTCHAs with Python - ScrapeOps, fecha de acceso: febrero 3, 2026, <https://scrapeops.io/python-web-scraping-playbook/python-how-to-solve-captcha-has/>
17. How to Solve Amazon CAPTCHA - Medium, fecha de acceso: febrero 3, 2026, https://medium.com/@captcha_solver/how-to-solve-amazon-captcha-4382924842cf
18. mzdsk100/ddddocr-rs: Rust implementation of OCR for captcha recognition - GitHub, fecha de acceso: febrero 3, 2026, <https://github.com/mzdsk100/ddddocr-rs>
19. dddocr - Rust OCR Library for Captcha Recognition - Lib.rs, fecha de acceso: febrero 3, 2026, <https://lib.rs/crates/ddddocr>
20. dddocr CAPTCHA Recognition: Text, Object, Slider Solver - MCP Market, fecha de acceso: febrero 3, 2026, <https://mcpmarket.com/server/ddddocr-captcha-recognition>
21. How to Bypass Amazon CAPTCHA for Web Scraping - Bright Data, fecha de acceso: febrero 3, 2026, <https://brightdata.com/blog/web-data/bypass-amazon-captcha>
22. why I built an Open-Source eBay Scraper instead of buying one | by ..., fecha de acceso: febrero 3, 2026, <https://medium.com/@noorsimar/why-i-built-an-open-source-ebay-scraper-instead-of-buying-one-ae1823ac5406>
23. AI Web Scraper for E-commerce: Monitor Prices, Deals & Stock - TenUp Software Services, fecha de acceso: febrero 3, 2026, <https://www.tenupsoft.com/blog/ai-web-scraper-helps-ecommerce-companies-track-competitors.html>
24. Wallapop API - Reddit, fecha de acceso: febrero 3, 2026, https://www.reddit.com/r/Wallapop/comments/1j0oe3f/api_wallapop/?tl=en
25. rmonvfer/wallapop_secret: Wallapop's API X-Signature ... - GitHub, fecha de acceso: febrero 3, 2026, https://github.com/rmonvfer/wallapop_secret
26. Understanding Mobile App Reverse Engineering: How Attackers Actually Break Your Apps | Iterators, fecha de acceso: febrero 3, 2026, <https://www.iteratorshq.com/blog/understanding-mobile-app-reverse-engineering-how-attackers-actually-break-your-apps/>
27. Resources for reverse engineering “unofficial APIs”. - GitHub, fecha de acceso: febrero 3, 2026, <https://github.com/ropcat/reversing-unofficial-APIs>
28. Reverse engineering the private API of an Android app secured by certificate pinning, fecha de acceso: febrero 3, 2026, <https://data-dive.com/reverse-engineer-android-app-api/>
29. toniprada/wallapop-users-scraper - GitHub, fecha de acceso: febrero 3, 2026, <https://github.com/toniprada/wallapop-users-scraper>
30. Vinted-API/VintedApi.py at main - GitHub, fecha de acceso: febrero 3, 2026, <https://github.com/hipsuc/Vinted-API/blob/main/VintedApi.py>

31. vinted-scraper - PyPI, fecha de acceso: febrero 3, 2026, <https://pypi.org/project/vinted-scraper/>
32. Giglium/vinted_scraper: A very simple Python package that scrapes the Vinted website to retrieve information about its items. - GitHub, fecha de acceso: febrero 3, 2026, https://github.com/Giglium/vinted_scraper
33. Vinted API Guide 2025: How to Extract Data Safely - Lobstr.io, fecha de acceso: febrero 3, 2026, <https://www.lobstr.io/blog/vinted-api>
34. Vinted Pro Integrations - API Documentation, fecha de acceso: febrero 3, 2026, <https://pro-docs.svc.vinted.com/>
35. How to Rotate Proxies in Python - Scrapeless, fecha de acceso: febrero 3, 2026, <https://www.scrapeless.com/en/blog/how-to-rotate-proxies-in-python>
36. tor-ip-rotation-python-example/README.md at master - GitHub, fecha de acceso: febrero 3, 2026, <https://github.com/baatout/tor-ip-rotation-python-example/blob/master/README.md>
37. Requests With Tor | learning-stem - GitHub Pages, fecha de acceso: febrero 3, 2026, https://sigmapie8.github.io/learning-stem/requests_with_tor.html
38. Stem list Tor circuit generated in file - python - Stack Overflow, fecha de acceso: febrero 3, 2026, <https://stackoverflow.com/questions/51162657/stem-list-tor-circuit-generated-in-file>
39. Using Python to Rotate Proxies and Avoid Detection | by swiftproxy - Medium, fecha de acceso: febrero 3, 2026, <https://medium.com/@swiftproxy/using-python-to-rotate-proxies-and-avoid-detection-9689ac3845b5>
40. Complete 2026 Amazon Scraping Guide: Product Data, Prices, Sellers, and More, fecha de acceso: febrero 3, 2026, <https://scrape.do/blog/amazon-scraping/>
41. How to Rotate Proxies in Python Using Requests (Easy Guide) - ScraperAPI, fecha de acceso: febrero 3, 2026, <https://www.scraperapi.com/blog/how-to-use-and-rotate-proxies-in-python/>
42. ¿Es legal el scraping en España? Guía sobre normativa y casos prácticos, fecha de acceso: febrero 3, 2026, <https://datstrats.com/blog/scraping-es-legal-espana/>
43. LA LEGALIDAD DEL SCREEN SCRAPING SEGÚN EL TRIBUNAL SUPREMO, fecha de acceso: febrero 3, 2026, <https://www.crconsultoreslegales.com/la-legalidad-del-screen-scraping-segun-el-tribunal-supremo/>
44. Análisis - Gómez-Acebo & Pombo, fecha de acceso: febrero 3, 2026, <https://ga-p.com/wp-content/uploads/2018/03/screen-scraping-condiciones-generales-de-la-contrataciyn-bases-de-datos-y-competencia-desleal.pdf>
45. Web Scraping Legal Issues: 2025 Enterprise Compliance Guide, fecha de acceso: febrero 3, 2026, <https://groupbwt.com/blog/is-web-scraping-legal/>
46. Is Website Scraping Legal? All You Need to Know - GDPR Local, fecha de acceso: febrero 3, 2026, <https://gdprlocal.com/is-website-scraping-legal-all-you-need-to-know/>
47. Is Web Scraping Legal in 2024? A Global Overview - Multilogin, fecha de acceso:

- febrero 3, 2026, <https://multilogin.com/blog/is-web-scraping-legal/>
48. The state of web scraping in the EU - IAPP, fecha de acceso: febrero 3, 2026, <https://iapp.org/news/a/the-state-of-web-scraping-in-the-eu>
49. Web Scraping for Stock Data: What I Learned Building My Own Analyzer | ScrapeGraphAI, fecha de acceso: febrero 3, 2026, <https://scrapegraphai.com/blog/stock-analysis>
50. How to Use Web Scraping for Inventory Data and Pricing Data on DigiKey, fecha de acceso: febrero 3, 2026, <https://www.actowizsolutions.com/web-scraping-inventory-pricing-data-digikey.php>