GUÍA NODESTUDIO · REVERSE API ENGINEER

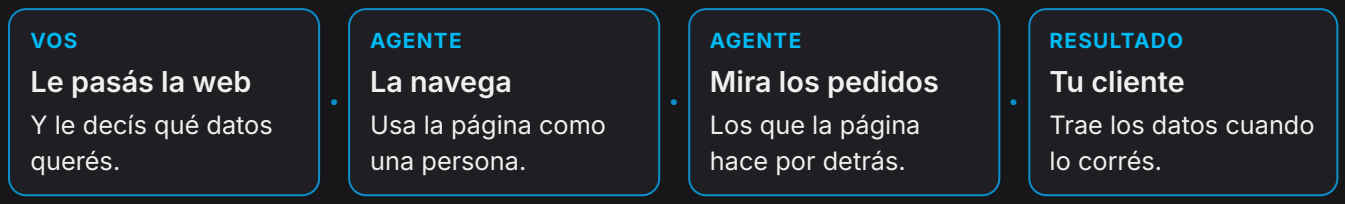
# Conectá tu IA a cualquier web, aunque no tenga API

Un agente mira qué pedidos hace una página por detrás y te arma un cliente para traer esos mismos datos cuando los necesites.

**Herramienta gratis · Claude Code · instalación en 10 minutos**

## Qué hace y qué necesitás

**Reverse API Engineer** es una herramienta gratis y de código abierto (licencia MIT) [1]. Casi todas las webs le piden sus datos a un servidor por detrás: precios, stock, listados. La herramienta mira esos pedidos y te arma un **cliente**: un programa chico que hace los mismos pedidos sin abrir la página.




Sirve para lo que hoy alguien copia a mano: el precio de tu competencia, el stock, los productos nuevos.

### Qué necesitás

- [ ] **Claude Code** OBLIGATORIO
  Con tu sesión iniciada. En nuestra prueba alcanzó con ese login, sin API key.
- [ ] **Python 3.11 y Node 20.19, o más nuevos** OBLIGATORIO
  Claude Code los revisa por vos en la instalación.
- [ ] **Google Chrome** OBLIGATORIO
  Funciona en Linux, Mac con chip Apple y Windows de 64 bits. Mac con Intel ya no.

> **QUÉ ES GRATIS Y QUÉ NO**
> La herramienta es gratis. Lo que consume es el modelo, mientras arma el cliente: en nuestras dos corridas la herramienta informó USD 0,67 y USD 1,27, con páginas simples. Una web más grande puede gastar más. Una vez armado, el cliente es código común: correrlo no usa IA.

## 1 Instalalo

Pegale esto a Claude Code. El link del repo va adentro, por si el prompt viaja solo.

**REPO** github.com/kalil0321/reverse-api-engineer

**PROMPT PARA PEGAR EN CLAUDE CODE**

```
Quiero instalar Reverse API Engineer: https://github.com/kalil0321/reverse-api-engineer

Revisá que tenga Python 3.11 y Node 20.19, o más nuevos. Si falta algo, avisame antes de instalarlo.

Instalá la herramienta con "uv tool install reverse-api-engineer" y la skill con "npx skills add kalil0321/reverse-api-engineer".

Después poné "agent_provider": "auto" en ~/.reverse-api/config.json, para que use un navegador limpio y no mi Chrome con mis sesiones. Confirmá con "reverse-api-engineer --version" que quedó instalado.
```

Instalación y skill probadas el 4/11/2026 con la versión 0.14.0.

**POR QUÉ "AUTO"**

Si no lo cambiás, usa tu Chrome de verdad, con tus sesiones abiertas. Para datos públicos no hace falta, y con "auto" fue como lo probamos. Si la web te pide iniciar sesión, ese otro modo es el que sirve, pero ese no lo probamos.

**ANTES DE INSTALAR UNA SKILL**

Las skills corren con todos los permisos de tu agente. Esta está en el repo: si querés, pedile a Claude Code que te la resuma antes de instalarla.

## 2 Armá tu primer cliente

Pedile los datos que querés, no los clics. Cambiá lo que está entre corchetes.

**ESCRIBILE ESTO**

```
Creá una carpeta nueva y vacía y trabajá ahí. Con Reverse API Engineer en modo
agente y --headless, entrá a [URL DE LA WEB] y mirá qué pedidos hace la página por
detrás para cargar [LOS DATOS QUE QUERÉS].

Armame un cliente que use esos mismos pedidos. Poné la URL también dentro del
texto del pedido, no solo en la opción -u.

Cuando termine, decime qué pedidos usa el cliente y cuánto costó la corrida, leé
el código y corrélo para confirmar que trae datos reales.
```

En un sitio de práctica tardó menos de 2 minutos. El repo avisa que puede llevar varios.

### Qué te deja

- [ ] **api_client.py**
  El cliente. Se corre sin navegador y sin IA.
- [ ] **README.md**
  Qué pedidos usa y cómo correrlo.

Quedan en una carpeta scripts/ con un nombre que describe lo que hace.

**CÓMO SABER SI QUEDÓ BIEN**

Fijate qué pedidos usa. Si ves rutas como /api/... va bien: en nuestra prueba usó /api/quotes?page=1 y trajo las 100 frases del sitio. Si en cambio lee el HTML (vas a ver BeautifulSoup), ese cliente sí se puede romper con un rediseño.

## 3 Tres pedidos listos

Son para usar con el prompt de la hoja anterior, en el lugar de los datos que querés.

**PRECIOS DE TU COMPETENCIA**

```
el nombre y el precio de cada producto de [CATEGORÍA], incluidas todas las páginas del listado
```

**STOCK**

```
si cada producto de [CATEGORÍA] tiene stock y, si la página lo muestra, cuántas unidades quedan
```

**PRODUCTOS NUEVOS**

```
los productos ordenados por más nuevos, con nombre, precio y link
```

### Y después, cuando lo necesites

El cliente ya está armado. Cada vez que quieras datos frescos, le pedís esto:

**PARA USARLO SEGUIDO**

```
Corré api_client.py, agregá el resultado a datos.csv con la fecha de hoy y decime qué cambió desde la última vez.
```

> **QUÉ PROBAMOS Y QUÉ NO**
>
> El flujo completo lo probamos en un sitio de práctica. Estos tres pedidos no los corrimos en tiendas reales: cada web es distinta, así que confirmá siempre que los datos que trae son los correctos.

## 4 Lo que nos pasó al probarlo

- [ ] **Se fue a otra página**
  Le pasamos la web solo con la opción -u y el agente dijo que no tenía URL: entró a otra parte del sitio. Con la URL dentro del pedido anduvo. Por eso el prompt la pide en los dos lados.
- [ ] **En una página sin pedidos, armó otra cosa**
  Esa otra parte del sitio era HTML puro, sin pedidos por detrás. Igual terminó bien y trajo los datos, pero con un lector de HTML, que un rediseño puede romper.
- [ ] **Revisó la carpeta donde estaba**
  Antes de arrancar leyó los archivos que encontró. Trabajá siempre en una carpeta vacía.
- [ ] **No guardó la captura**
  En nuestras corridas no quedó el archivo con los pedidos capturados. Para mejorar un cliente, volvimos a correrlo.

### Límites para tener en cuenta

- [ ] **Algunas webs lo van a frenar**
  Las que detectan bots pueden bloquear la captura.
- [ ] **Leé el código antes de usarlo**
  Lo escribe un modelo, y corre en tu computadora.
- [ ] **No es una API oficial**
  Un cambio de diseño normalmente no lo afecta, pero si la web cambia sus pedidos, deja de andar. Ahí lo volvés a armar.
- [ ] **Respetá los términos de cada web**
  Usalo con datos públicos, revisá si el sitio lo permite y no lo llenes de pedidos.



# NodeStudio

Automatizamos procesos de negocio con IA y sistemas a medida.

INSTAGRAM

@valentinpanizza

WEB

nodestudioai.com