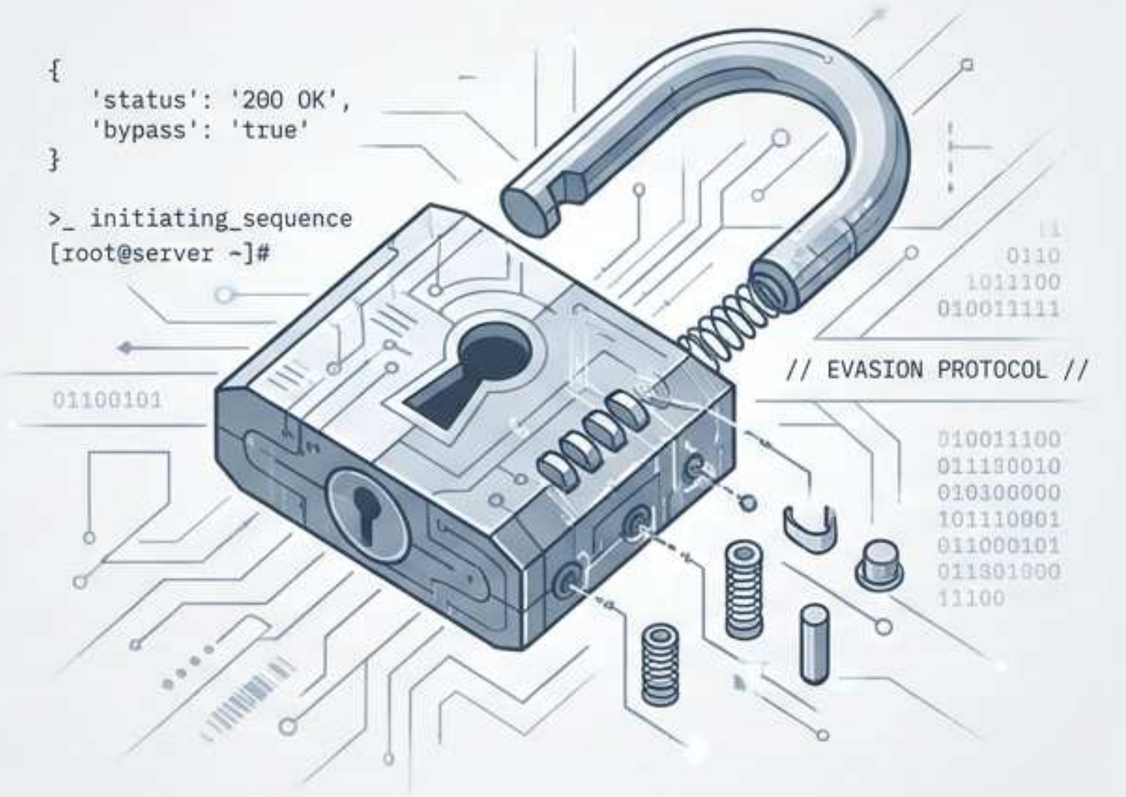
Playwright Anti-Bot: Los 5 Niveles de Defensa

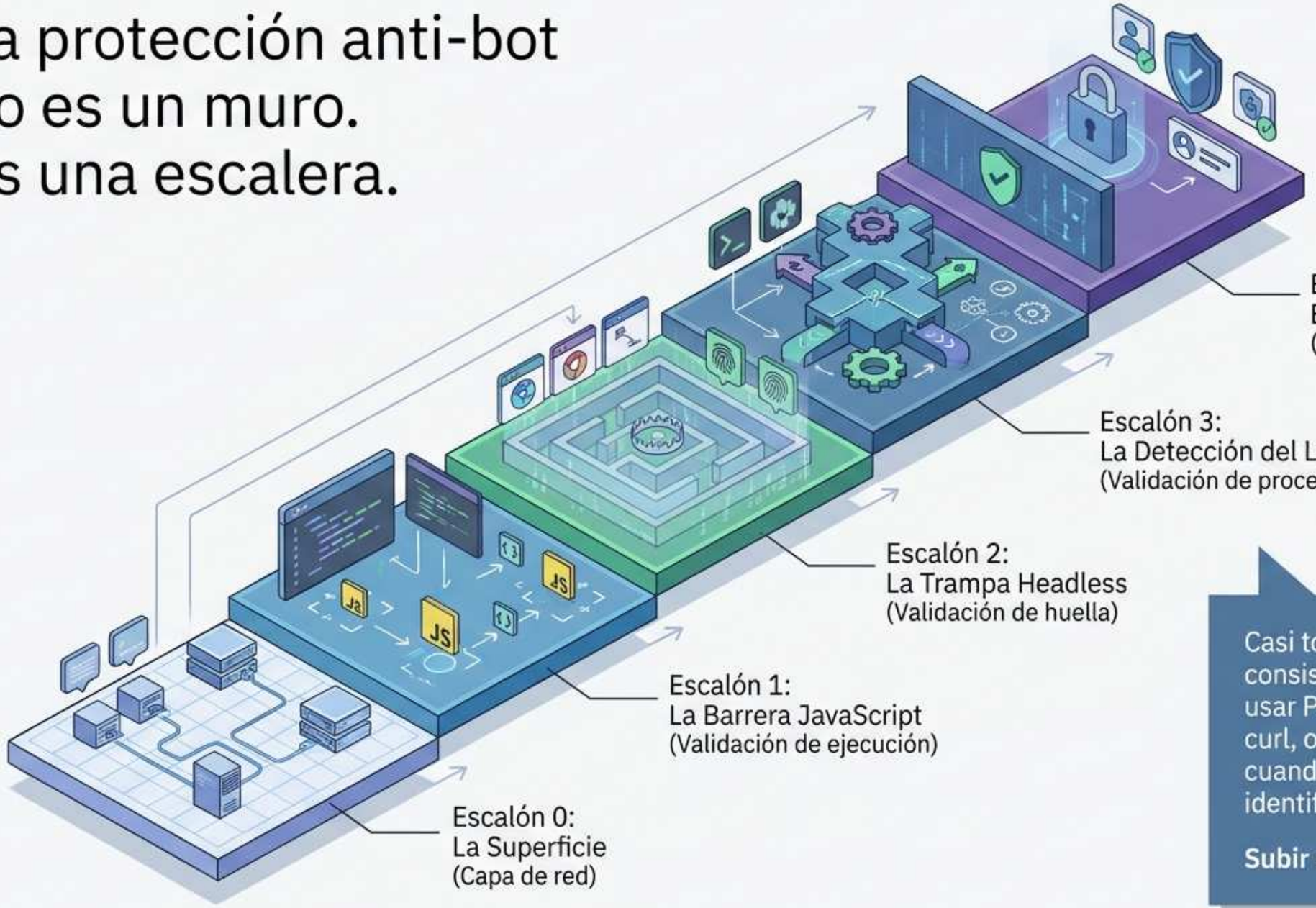
Por qué tu scraper recibe un 403 y cómo escalar tus herramientas sin ensayo y error.



Basado en el análisis táctico de Cristian Palau.

Gemini Notebook

La protección anti-bot no es un muro. Es una escalera.



Escalón 4:
El Muro Infranqueable
(Validación de identidad)

Escalón 3:
La Detección del Lanzador
(Validación de proceso)

Escalón 2:
La Trampa Headless
(Validación de huella)

Escalón 1:
La Barrera JavaScript
(Validación de ejecución)

Escalón 0:
La Superficie
(Capa de red)

Casi todos los intentos fallidos consisten en saltarse un peldaño: usar Playwright donde bastaba curl, o pelear con un CAPTCHA cuando el navegador se identifica como headless.

Subir un nivel cuesta recursos.

Gemini Notebook

Matriz de Diagnóstico Inicial

Síntoma Visual	Nivel de Defensa	Respuesta Táctica
403 instantáneo sin User-Agent o Referer	Nivel 0 (Superficie)	`curl -A -e -b -c`
HTML vacío o mensaje “Activa JavaScript"	Nivel 1 (Guardián SPA/JS)	Navegador Headless (Playwright por defecto)
CAPTCHA que no termina nunca en headless	Nivel 2 (Detección de Huella)	Navegador Headed + Perfil Persistente
403 o CAPTCHA al instante, a pesar de perfil válido	Nivel 3 (Detección CDP)	Arranque manual + Conexión CDP
Tu navegador personal habitual también está bloqueado	Nivel 4 (Bloqueo de Cuenta)	Buscar otra fuente de datos

Gemini Notebook

Nivel 0: La Superficie (Identidad Básica)



Los bloqueos más bajos no tienen protección seria, solo exigen cortesía básica de red. El servidor comprueba cabeceras, no comportamientos.

```
curl -A 'Mozilla/5.0...'
     -e https://origen.com
     -b cookies.txt -c cookies.txt
     https://destino.com
```

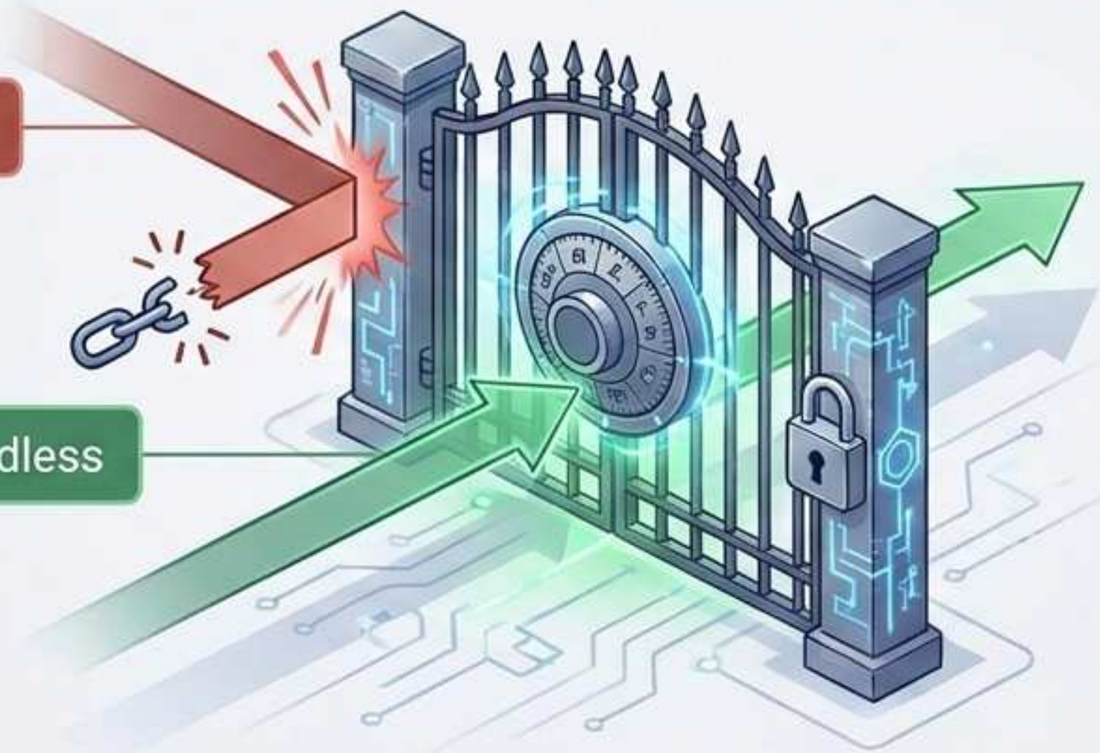
- Falsifica el User-Agent (Identidad)
- Falsifica el Referer (Origen)
- Lee y guarda cookies (Persistencia)

Consejo: Si el sitio rechaza curl con cabeceras correctas, puede estar analizando la huella TLS (JA3/JA4). En ese caso, usa **curl_cffi en Python** para imitar la firma criptográfica exacta.

Gemini Notebook

Nivel 1: Sin JavaScript no hay página

Dos motivos te detienen aquí: la web está renderizada enteramente en JS, o hay un guardián invisible.



Anubis

Exige una "Prueba de Trabajo" resolviendo un hash SHA-256.

Cloudflare JS Detections

Inyecta un script invisible; si no se ejecuta, el HTML real nunca se sirve.

```
browser = pw.chromium.launch() # Headless por defecto
```

El **Chromium headless** es eficiente, pero es un binario distinto al Chrome real. Esto nos lleva directamente a la siguiente trampa.

Gemini Notebook

Nivel 2: La Trampa del Headless

HeadlessChrome	Chrome con ventana
Identidad webdriver: TRUE	**Identidad** webdriver: TRUE (Aún detectado por CDP)
Renderizado GPU: SwiftShader (Emulado)	**Renderizado** GPU: Radeon / NVIDIA (Hardware real)
Objetos Plugins: 0 | window.chrome: vacío	**Objetos** Plugins: 5+ | window.chrome: presente
Geometría outerWidth == innerWidth (Sin marco)	**Geometría** outerWidth > innerWidth (Ventana real)
Comportamiento Movimiento robótico instantáneo	**Comportamiento** Pre-fetch y eventos de ratón

Cloudflare y Turnstile no preguntan si eres headless. Miden cuántas señales anómalas emites simultáneamente. **La solución requiere una GPU real y un entorno gráfico.**

Gemini Notebook

Solución al Nivel 2: El Perfil Persistente

Anatomía de una Credencial



Una cookie de autorización no es solo texto; está entrelazada criptográficamente con el dispositivo que la obtuvo. Un perfil limpio cada vez que arrancas garantiza un bloqueo.

```
context = pw.chromium.launch_persistent_context(
    user_data_dir='.browser-profiles/autorizado',
    channel='chrome',
    headless=False,
    args=['--disable-blink-features=AutomationControlled']
)
```

⚠ **Regla de oro:** Utiliza page.close(), nunca cierres el contexto completo, o destruirás la sesión ganada.

Gemini Notebook

Nivel 3: La Detección del Lanzador



El sitio detecta quién pulsa el interruptor de arranque. Cuando Playwright arranca Chrome, inyecta banderas delatoras de depuración (--enable-automation, --remote-debugging-pipe).

El detalle crucial: El uso de 'pipe' fuerza la variable **navigator.webdriver = true** a nivel profundo del motor Blink, delatando el bot antes de que la página termine de cargar.

Gemini Notebook

Solución al Nivel 3: El Puente CDP

Humano / Terminal



Expone el Puerto **9222**

El Puente CDP

Playwright / Python



Usa **connect_over_cdp** para cruzar y secuestrar la pestaña

Paso 1: Lanzar via Shell

```
google-chrome --remote-debugging-port=9222
--user-data-dir='.perfil'
```

Paso 2: Conectar via Python

```
browser = pw.chromium.connect_over_cdp
('http://127.0.0.1:9222')
```

¡Magia! Al conectar por puerto en lugar de pipe, **navigator.webdriver** baja a **FALSE**. Eres indetectable en el arranque.

Gemini Notebook

Las Trampas Silenciosas del CDP

El Bloqueo de Chrome 136+

Desde la versión 136, Chrome ignora el puerto de depuración si usas el perfil por defecto del sistema. Arrancará sin errores, pero el puerto no existirá.

Solución:
Usar siempre un directorio aislado con `--user-data-dir`.

La Instancia Zombie

Si ya hay un Chrome abierto usando ese perfil, el arranque se delega a esa instancia y el puerto 9222 nunca se abre.

Diagnóstico de Terminal

Antes de culpar a Playwright, verifica que el puente CDP existe:

```
curl -s http://127.0.0.1:9222/json/version | jq -r .Browser
```

(Si no devuelve 'Chrome/XXX', el fallo está en la shell, no en tu código).

Gemini Notebook

Nivel 4: El Muro Infranqueable



Has superado todas las detecciones técnicas.
Estás usando un navegador real, a mano,
y la web sigue bloqueando el contenido.

Falsos Positivos de Anti-Bot

- **Muros de pago (Paywalls):** Si un sitio oculta campos a cuentas gratuitas, físicamente no envía los datos.
- **Límites de cuenta:** Tope diario de consultas por usuario alcanzado.

Decisión Táctica: No hay truco de User-Agent que revele lo que el servidor decide no enviar.
La acción correcta es **buscar otra fuente de datos**.

Gemini Notebook

El Bloqueo Transversal: Límite de Peticiones



Diagnóstico de Bloqueo por IP

Un bloqueo súbito tras una ráfaga de peticiones rápidas, donde incluso `curl` o tu navegador diario fallan simultáneamente.

Bloqueo por IP, no por cliente.

> **“ No optimices el camuflaje tecnológico: optimiza la educación.” ”**

Mejores Prácticas

1. Añadir fluctuación artificial: `time.sleep(random.uniform(1.5, 3.5))`
2. Mantener concurrencia estrictamente limitada (idealmente 1).
3. Respetar siempre las cabeceras HTTP Retry-After.
4. Cachear resultados implacablemente a nivel local.

Gemini Notebook

Nivel Avanzado: La Fuga del Canal CDP (Runtime.enable)

La Detección del Micrófono Abierto

Hay una detección más profunda que el binario: **saber que alguien está escuchando**.

Sistemas inyectan JS que provoca la **serialización** de objetos.

Si Playwright está conectado (usando Runtime.enable), el JS mide el tiempo extra de serialización y deduce la automatización.

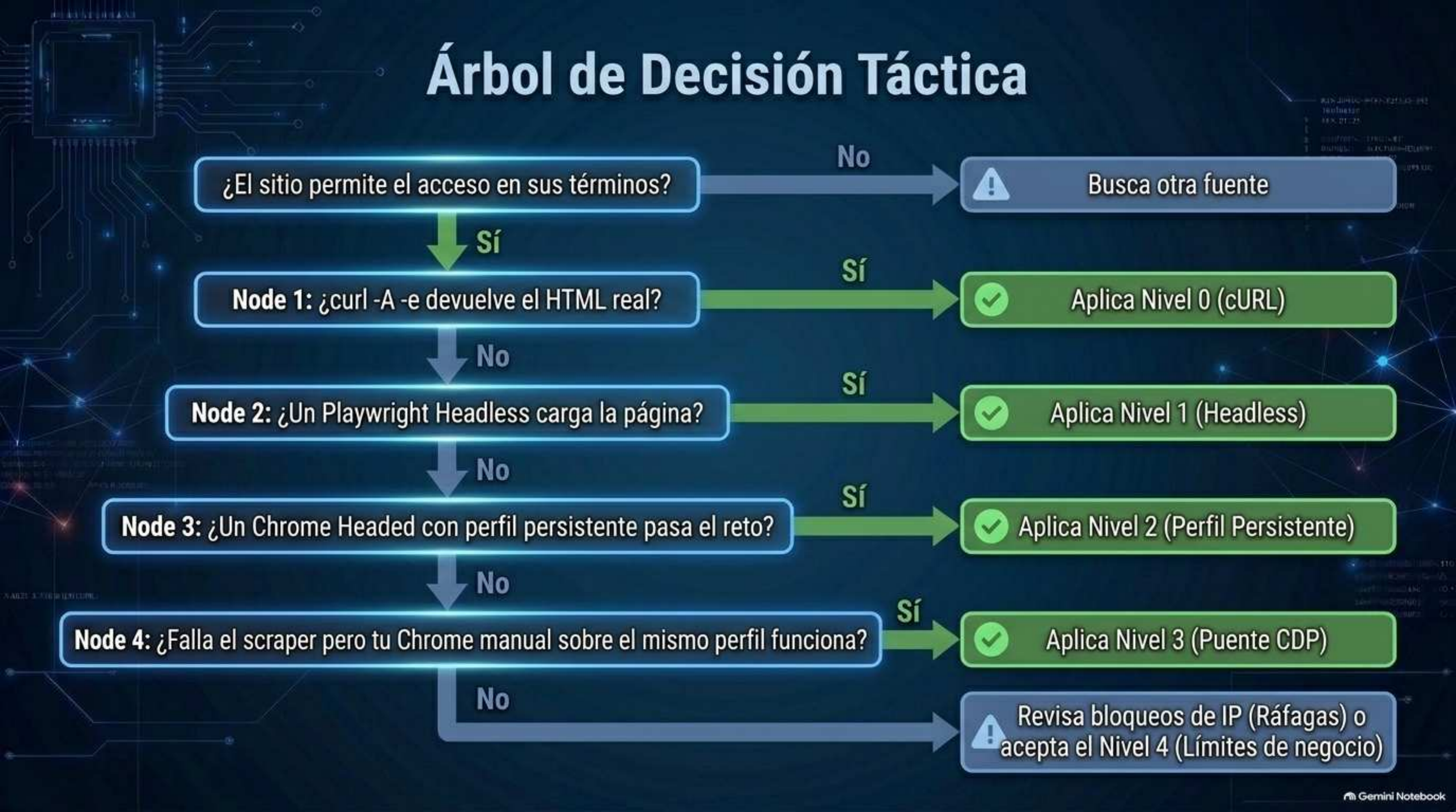


La Carrera Armamentística

Forks como Patchright (Python) interceptan esta fuga desactivando la API de consola y ejecutando JS en contextos contextos aislados para no dejar la puerta abierta.

Gemini Notebook

Árbol de Decisión Táctica



Gemini Notebook

Conclusión: El Precio del Sigilo



- 1. **Escala progresivamente**: El sigilo se compra por capas. No gastes recursos gráficos si un simple curl es suficiente.
- 2. **Identifica el bloqueo**: Usa la matriz de diagnóstico. Conocer qué bandera levantaste es el 90% de la solución.
- 3. **Elige tu escalón**: El objetivo no es ganar siempre, sino saber cuándo el coste técnico supera el valor de los datos.

⚠ Advertencia Legal: Un 403 no es un permiso. Revisa siempre los términos de servicio, la carga del servidor, el RGPD y las Directivas de Minería de Datos (UE 2019/790). Extrae con responsabilidad.

Gemini Notebook