

# Conversación con Gemini

Exported: 12/6/2026 11:50:26

Jordi H.

Has dicho google analytics y tag manager me dan dos funciones fáciles de implementar. Añadiendo unas pocas líneas a mis páginas web obtengo unas estadísticas de uso muy completas y además puedo evaluar el rendimiento del browser con los eventos de carga de la página. con la situación internacional y la amenaza de apagado tecnológico a determinados países en función de las relaciones diplomáticas, me parece interesante evaluar alternativas. Cuáles te parecen interesantes, con esos dos funciones en me

Gemini

Es una excelente reflexión. El riesgo de dependencia tecnológica geopolítica (el llamado vendor lock-in a nivel estatal) es un argumento de peso que se suma a las ya clásicas razones de privacidad (GDPR/AEPD) y rendimiento web para buscar alternativas. Cuando buscas sustituir **Google Analytics (GA)** y **Google Tag Manager (GTM)** **manteniendo la facilidad de añadir unas pocas líneas de código, pero asegurando independencia total (Self-Hosted / Software Libre)** y medición del rendimiento del navegador (Core Web Vitals / Performance)\*\*\*, el ecosistema de código abierto actual ofrece soluciones maduras. Aquí tienes las mejores alternativas divididas según tu enfoque de infraestructura:

## 1. El Enfoque Todo en Uno (Análítica + Rendimiento)

Si quieres resolver ambos problemas con una sola herramienta, un único script y una base de datos unificada, estas son las opciones más potentes

### Swetrix

Es una de las promesas más firmes en el ecosistema open-source actual para cubrir exactamente lo que pides. No solo mide métricas de tráfico sin cookies, sino que incluye un módulo nativo de **Performance Monitoring**

- Métricas de Carga: Captura de forma automática datos de rendimiento del navegador (tiempos de carga, Web Vitals) y errores de JavaScript en tiempo real
- Implementación: Un script ligero en el HTML.
- Autohospedaje: Totalmente preparado para correr en tu propia infraestructura mediante Docker y ClickHouse/PostgreSQL

### Matomo (Anteriormente Piwik)

Es el "gigante" del software libre en analítica y el reemplazo más cercano a las capacidades de un entorno GA+GTM. Es la opción elegida por la Comisión Europea

- Métricas de Carga: Viene por defecto con un reporte detallado de "Velocidad de la página" (tiempos de generación del servidor, tiempo de transferencia de red, tiempo de parseo DOM y tiempo de carga total en el navegador del usuario)
- ¿Y el Tag Manager?: Matomo incluye su propio Tag Manager integrado (Matomo Tag Manager). Puedes gestionar etiquetas, eventos personalizados y activadores desde su propia interfaz sin tocar el código de tus páginas tras la instalación inicial
- Autohospedaje: Excelente. Requiere una pila tradicional (PHP/MySQL), aunque para volúmenes muy altos de tráfico exige optimización de base de datos

## 2. El Enfoque Moderno y Ligero (Micro-scripts)

Si prefieres separar las aguas o priorizas scripts ultra-ligeros (de apenas 1 KB frente a los más de 40 KB de Google Analytics) para que la propia analítica no penalice el rendimiento del navegador

Umami o Plausible (Community Edition)

- Ambas son las opciones favoritas de los desarrolladores actuales por su simplicidad y diseño limpio.
- Analítica: Te dan estadísticas completas (visitas, fuentes, países, dispositivos) y permiten configurar eventos personalizados muy fácilmente (equivalente a los eventos de Tag Manager pero definidos por código o atributos data-\* en el HTML
  - Rendimiento: Al ser proyectos enfocados en la simplicidad extrema, no miden por defecto los tiempos de carga detallados del navegador (Web Vitals). Su gran ventaja en rendimiento es que sus scripts pesan tan poco que aceleran la carga de la página respecto a G
  - Autohospedaje: docker-compose up y listo. Umami corre de forma nativa sobre Node.js y PostgreSQL/MySQL; Plausible utiliza PostgreSQL y ClickHouse (ideal para ingerir millones de eventos)

3. ¿Cómo suplir el análisis de Rendimiento si usas un script ligero?

- Si optas por una alternativa ligera como Umami o Plausible pero necesitas auditar los eventos de carga del navegador de tus usuarios reales (RUM - Real User Monitoring) de manera soberana, puedes complementarlo con herramientas de observabilidad específicas y libre
- Lighthouse CI / Web Vitals CLI: Herramientas open-source de automatización para medir el rendimiento en tus pipelines de despliegue antes de que lleguen al usuario
  - OpenObserve: Una alternativa open-source a Datadog. Puedes enviarle mediante un script mínimo los datos de la API de rendimiento del navegador (performance.timing o la librería web-vitals de JS ) directamente a tu servidor mediante un sencillo fetch(

Resumen de Recomendaciones

| Herramienta | Reemplaza a...   | Mide Rendimiento   | Dificultad de      | Stack Ideal   |
|-------------|------------------|--------------------|--------------------|---------------|
| Matomo      | GA4 + GTM        | Sí (Muy completo)  | Media (Requiere    | PHP + MySQL / |
| Swetrix     | GA4 + Monitor de | Sí (Módulo nativo) | Media-Baja (Docker | Node.js +     |
| Umami       | GA4 (Esencial)   | No (Pero mejora la | Muy Baja           | Node.js +     |

**Mi recomendación:** Si quieres mantener la experiencia de un Tag Manager independiente y reportes nativos de rendimiento del navegador sin depender de terceros, **Matomo** es el camino más robusto y maduro. Si buscas algo más moderno, con una arquitectura de datos más eficiente para telemetría y reportes de rendimiento nativos, dale una mirada a **Swetr**

Jordi H.

Has dicho para el proveedor de grandes aplicaciones web, las recomendaciones que me das, son todas igualmente fáciles de integrar en su generación de páginas web? esencialmente, funcionan igual que las herramientas de googl

Gemini

Para un proveedor de grandes aplicaciones web, el término "fácil" adquiere otra dimensión. Cuando pasas de un sitio web estático o un blog a una arquitectura de gran escala (como SPAs con React/Vue, aplicaciones renderizadas en el servidor con Next.js/Nuxt, o microfrontends), **la integración no es exactamente igual**

ptualmente compartan el mismo punto de partida que Google. Para entender cómo encajan en tu infraestructura, analicemos el mecanismo de integración y la inyección de código.

## El Mecanismo: ¿Funcionan igual que Google?

**Sí, en la superficie.** El principio fundamental es el mismo: todas estas herramientas te proporcionan un **SDK de ipt** (un archivo .js ligero)\*\* que debes inyectar en el <head> o al final del <body> de tus plantillas o componentes globales. Sin embargo, a gran escala hay diferencias cruciales en cómo operan por debajo:

CODE

```
[ Navegador del Usuario ]
|
+---> (Descarga Script de Analítica)
|
+---> (Captura Eventos de Carga / Interacción)
|
+---> [ Balancador de Carga / Proxy Inverso ] ---> [ Tu Infraestructura Self-Hosted ]
(Matomo / Umami / Swetrix)
```

### 1. El ciclo de vida de la página y las SPAs (Single Page Applications)

Google Analytics 4 (GA4) gestiona muy bien de forma nativa los cambios de ruta virtuales (cuando el usuario navega sin que la página se recargue realmente)

- Umami y Swetrix: Lo hacen de forma idéntica o incluso más limpia. Detectan los cambios en la API history del navegador automáticamente. Su integración en frameworks modernos es tan sencilla como importar un paquete npm o añadir un componente global
- Matomo: Si usas su Tag Manager o su script estándar en una SPA, a gran escala suele requerir que dispares manualmente el evento de "Page View" cada vez que tu enrutador (ej. React Router) cambie de vista. No es difícil, pero requiere añadir líneas de código específicas en el ciclo de vida de tu aplicación

### 2. El Tag Manager: La gran diferencia operativa

Google Tag Manager funciona inyectando un contenedor vacío en tu web. Luego, desde la nube de Google, tú decides qué scripts de terceros (píxeles de Meta, herramientas de soporte, etc.) se inyectan dinámicamente

- Matomo Tag Manager: Es el único de la lista que clona este comportamiento exacto. Instalas su contenedor y gestionas el resto desde su panel
- Umami / Plausible / Swetrix: No son Tag Managers. Si necesitas inyectar scripts de terceros dinámicamente en tu gran aplicación web, estas herramientas no te sirven para ese propósito. Tendrías que gestionar las etiquetas directamente en el código de tu aplicación (lo cual suele ser mejor para el rendimiento y el control de versiones) o utilizar un Tag Manager de código abierto dedicado (como Apache Unomi o Stape self-host)

## Evaluación de Integración a Gran Escala

A continuación, vemos cómo se comportan estas alternativas cuando se integran en el pipeline de generación de páginas de un gran proveedor

### Matomo: El equivalente corporativo

- Facilidad de integración: Media. Si usas su Tag Manager, es tan fácil como Google. Si usas su analítica avanzada, el script es más pesado y requiere configuración para no penalizar el First Contentful Paint (FCP) en aplicaciones críticas
- Gestión de eventos: Idéntica a Google (puedes usar un `dataLayer.push()`).
- Reto a gran escala: El script estándar de Matomo rastrea muchas cosas por defecto (mapas de calor,

grabaciones de sesión si se activan). En aplicaciones con millones de visitas, esto genera un volumen de peticiones síncronas que debes canalizar muy bien

## Umami: La simplicidad del desarrollador moderno

- Facilidad de integración: Muy Alta. Está diseñado específicamente pensando en la web moderna (Next.js, Remix, Vue). Se integra mediante propiedades `data-` directamente en los elementos HTML
- Ejemplo de evento: Para medir un clic en un botón, en lugar de configurar disparadores complejos, solo haces:HTML

CODE

```
<button data-umami-event="Formulario de Registro - Clic">Registrarse</button>
```

- Gestión en producción: Al no usar cookies, evita la complejidad técnica (y legal) de los banners de consentimiento intrusivos en tu código

## Swetrix: El híbrido de rendimiento

- Facilidad de integración: Alta. Su SDK está pensado para pasarle configuraciones en el momento de la inicialización de la app
- Captura de rendimiento: A diferencia de Google (donde los eventos de carga a veces requieren configurar variables en GTM), Swetrix empieza a medir las APIs de rendimiento del navegador en cuanto se ejecuta, enviando la telemetría automáticamente sin que configures nada en tu código

## El verdadero cambio para un Gran Proveedor: El Endpoint

Hay un detalle crucial que cambia por completo respecto a Google: **la dirección a donde se envían los datos** a `google-analytics.com`. Con tu propia alternativa, apuntará a tu propio subdominio (ej. `analytics.tuempresa.com`).

- Ventaja técnica enorme: Al ser tráfico del mismo dominio o subdominio de confianza, evitas que los bloqueadores de publicidad (AdBlockers) bloqueen la analítica, algo que con Google te hace perder entre el 15% y el 30% de las métricas reales de tus usuarios
- Esfuerzo de infraestructura: Debes configurar un proxy inverso (Nginx, Traefik) frente a tu aplicación web para enrutar las peticiones de analítica hacia los servidores donde tengas alojada la herramienta, garantizando alta disponibilidad

### Jordi H.

Has dicho en mi empresa afortunadamente podemos asumir el endpoint (tenemos hasta datacenter propio). pero, en general, existen proveedores que asuman el papel de Google y sean inmunes al vendor-lock diplomático del que hablamos

### Gemini

Tener centro de datos propio y control sobre vuestros endpoints os coloca en una posición de ventaja absoluta. Esa soberanía de infraestructura es el pilar más difícil de conseguir. Ahora bien, respondiendo a tu pregunta: sí, hay un gran proveedor de aplicaciones web que prefiere externalizar el servicio (modelo SaaS) pero mitigando ese riesgo geopolítico, **sí existen proveedores que asumen el papel de Google**, pero con enfoques legales, geográficos y de gobernanza radicalmente distintos. Para que un proveedor sea verdaderamente "inmune"

" (o altamente resiliente) a un apagado tecnológico por bloqueos diplomáticos bilaterales, debe cumplir tres condiciones: **soberanía jurisdiccional (no depender de leyes de extraterritorialidad como la US Cloud Act)** y **arquitectura de la aplicación de analítica** que cumplen este perfil:

## 1. Proveedores SaaS Europeos "Privacy-First" (Soberanía Jurisdiccional)

En Europa ha florecido un ecosistema de empresas de analítica que nacieron para competir contra Google, no solo en privacidad (GDPR estricto), sino en garantizar que los datos y el software pertenecen al territorio de la Unión Europea. Al operar bajo las leyes de la UE, están protegidas contra decisiones unilaterales de potencias extranjeras

- **Plausible Analytics (Estonia):** Es una empresa completamente independiente, constituida en la UE. Su infraestructura está alojada en Hetzner (un proveedor de infraestructura e infraestructuras críticas 100 % alemán). No utilizan servidores de AWS, Google Cloud ni Azure. Todo su software es de código abierto
- **Fathom Analytics (Canadá / Operaciones en UE):** Aunque la empresa es canadiense, tienen una infraestructura específica para la UE que procesa y almacena todo de manera local a través de proveedores europeos, garantizando que el tráfico no pase por manos estadounidenses
- **Matomo Cloud (Francia/Alemania):** La versión en la nube de Matomo está gestionada por los propios creadores del proyecto de software libre. Sus servidores físicos están en Europa Central y operan bajo estricta legislación comunitaria

**¿Por qué son más inmunes?** Al no tener matriz en EE. UU., no se ven afectados por órdenes ejecutivas de la Casa Blanca ni por la Cloud Act (que permite al gobierno estadounidense exigir datos a empresas de su país aunque estén en servidores extranjeros)

## 2. El Enfoque del Proveedor de Infraestructura Neutral (IaaS)

A gran escala, otra opción habitual para evitar el "bloqueo de Google" sin desplegar hierro propio es utilizar proveedores de infraestructura en la nube que basen su negocio en la neutralidad o en una distribución geográfica dispersa. Si en lugar de usar Google Analytics montas **Umami** o **Matomo** en uno de estos entornos, el riesgo diplomático se reduce drásticamente

- **Hetzner (Alemania / Finlandia):** Es el bastión del hosting europeo. Excelente rendimiento/precio y completamente ajeno al ecosistema tecnológico estadounidense
- **OVHcloud (Francia):** Es el principal promotor de la iniciativa Gaia-X (el proyecto europeo para crear una nube soberana). Tienen centros de datos hiper-seguros y están muy orientados a empresas gubernamentales y grandes corporaciones que exigen inmunidad geopolítica
- **Scaleway (Francia):** Otro actor clave en la nube soberana europea, con infraestructura moderna muy enfocada en contenedores y arquitecturas nativas de la nube

## El "Seguro de Vida" Definitivo: El Enfoque Multicloud / Híbrido

Dado que vosotros contáis con centro de datos propio, la verdadera inmunidad al vendor-lock no te la da un proveedor concreto, sino **la arquitectura de la aplicación de analítica**. Si contratas una solución como **Plausible** o **Umami** en su modalidad SaaS (para ahorrar costes de mantenimiento iniciales), el riesgo de apagado tecnológico se mitiga mediante un plan de contingencia basado en su naturaleza open-source

CODE

```
[ Producción: Plausible/Umami SaaS ]
|
+---> (Si hay conflicto diplomático o caída)
|
v
[ Plan de Contingencia instantáneo ] ---> Levantar el contenedor Docker en vuestro Datacenter
y redirigir el Proxy Inverso (Endpoint).
```

Al ser herramientas de código abierto, si el proveedor SaaS sufriese un bloqueo, vuestro equipo de sistemas podría levantar la versión Self-Hosted en vuestro propio centro de datos en cuestión de horas, importar e

l último backup de datos y cambiar los apuntamientos DNS de vuestro endpoint. Con Google Analytics, si cierras el grifo, la telemetría simplemente muere y el histórico se pierde

**Jordi H.**

Has dicho si hay una CDN implicada, cual es mas o menos el tiempo de propagación de los cambios?

**Gemini**

Cuando hay una CDN (Content Delivery Network) involucrada, el concepto de "tiempo de propagación" varía significativamente dependiendo de **qué** estés cambiando. No es lo mismo modificar el código del script de analítica que cambiar una etiqueta dentro de un Tag Manager. Para un gran proveedor con infraestructura propia y CDN, el comportamiento se divide en dos escenarios

### 1. El script estático (Caso de Umami, Plausible o el script base de Matomo)

Si estás sirviendo el archivo JavaScript de analítica (por ejemplo, tracker.js) a través de tu CDN para que cargue a máxima velocidad desde el nodo más cercano al usuario, el script se comporta como cualquier otro recurso estático (un archivo CSS o una imagen)

- Tiempo de propagación con invalidación activa: Casi instantáneo (de 2 a 30 segundos). Si actualizas el software de analítica en tu centro de datos y lanzas una orden de Purge o Invalidation en la API de tu CDN (ya sea Cloudflare, Fastly, Akamai o un despliegue propio con Varnish), los nodos perimetrales borrarán la versión vieja y pedirán la nueva en la siguiente visita
- Tiempo de propagación pasivo (Cache del navegador): Depende del encabezado Cache-Control. Si el navegador del usuario final ya descargó el script y tiene un TTL (Time To Live) de 24 horas, ese usuario o específico no verá el cambio hasta que expire su caché local o fuerce un refresco, independientemente de lo rápido que actúe la CDN

### 2. El contenedor dinámico (Caso de Matomo Tag Manager)

Aquí es donde la CDN requiere una configuración fina, porque emula el comportamiento de Google Tag Manager. Cuando un analista añade una etiqueta de seguimiento o un evento en el panel de control y pulsa a "Publicar", el sistema genera un nuevo archivo JavaScript (el "contenedor")

- Tiempo de compilación en el servidor: 1 a 5 segundos. El servidor de Matomo genera el nuevo archivo .js con las nuevas reglas
- Propagación en la CDN: Para que esto sea eficiente en grandes aplicaciones, se suelen usar dos estrategias

#### Estrategia A: TTL bajo con Revalidación (Recomendado)

Se configura la CDN para que guarde el script del Tag Manager con un TTL muy corto (por ejemplo, 2 o 5 minutos) o utilizando el encabezado Cache-Control: no-cache, must-revalidate combinado con ETags

- Tiempo de propagación: Máximo 2-5 minutos. La CDN pregunta constantemente a tu centro de datos si el archivo ha cambiado. Si hay una versión nueva, la actualiza sobre la marcha de forma pasiva

#### Estrategia B: Purga por Webhook (Arquitectura Event-Driven)

Es el enfoque síncrono que usan las grandes plataformas. Configuras un webhook en tu sistema de analítica. En el instante en que el usuario hace clic en "Publicar" en el panel de Matomo, este lanza una petición HTTP a la API de tu CDN ordenando purgar específicamente la URL de ese contenedor

- Tiempo de propagación: Menos de 5 segundos a nivel mundial. Es el comportamiento más cercano a Google Tag Manager. El cambio impacta en el borde de la red de inmediato

## Resumen de Tiempos Esperados

| Elemento a cambiar              | Método de actualización             | Tiempo de propagación global |
|---------------------------------|-------------------------------------|------------------------------|
| Código del SDK / Tracker        | Purga de CDN en despliegue (CI/CD)  | 10 - 30 segundos             |
| Configuración de Etiquetas (Tag | Purga automatizada por Webhook      | < 5 segundos                 |
| Configuración de Etiquetas (Tag | Caché pasiva por tiempo (TTL corto) | 2 - 5 minutos                |

**En la práctica:** Si diseñas la arquitectura acoplando la publicación de tu herramienta self-hosted con la API de purga de tu CDN, tus analistas y desarrolladores experimentarán **la misma inmediatez que con la** rolando además de extremo a extremo las políticas de caché para no penalizar el rendimiento del navegador.

---